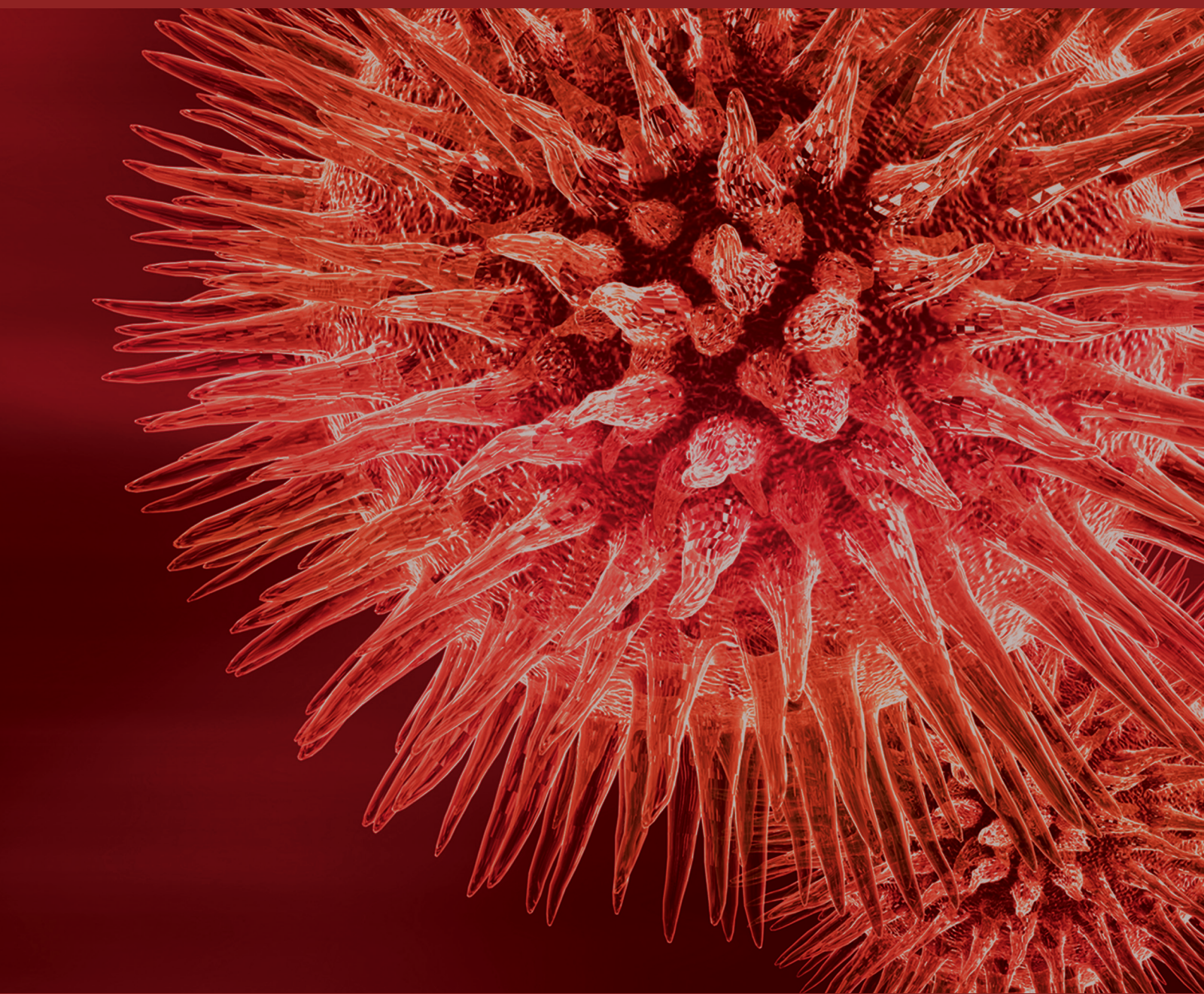
BioMed Research International

Scalable Data Mining Algorithms in Computational Biology and Biomedicine

Guest Editors: Quan Zou, Dariusz Mrozek, Qin Ma, and Yungang Xu



Scalable Data Mining Algorithms in Computational Biology and Biomedicine

BioMed Research International

Scalable Data Mining Algorithms in Computational Biology and Biomedicine

Guest Editors: Quan Zou, Dariusz Mrozek, Qin Ma, and Yungang Xu

Contents

Hindawi Publishing Corporation
BioMed Research International
Volume 2017, Article ID 5652041, 3 pages
http://dx.doi.org/10.1155/2017/5652041

Editorial

Scalable Data Mining Algorithms in Computational Biology and Biomedicine

Quan Zou,[1] Dariusz Mrozek,[2] Qin Ma,[3] and Yungang Xu[4]

[1]*School of Computer Science and Technology, Tianjin University, Tianjin 300354, China*
[2]*Institute of Informatics, Silesian University of Technology, 44-100 Gliwice, Poland*
[3]*Department of Mathematics and Statistics and Department of Agronomy, Horticulture, and Plant Science, BioSNTR, South Dakota State University, Brookings, SD 57007, USA*
[4]*Center for Bioinformatics and Systems Biology, Wake Forest School of Medicine, Winston-Salem, NC 27157, USA*

Correspondence should be addressed to Quan Zou; zouquan@tju.edu.cn

Received 29 December 2016; Accepted 4 January 2017; Published 28 February 2017

Since "Precision Medicine" was initially launched by President Obama, it presents a huge challenge and chance for the computational biology and biomedicine. In recent years, computational methods appeared vastly in the biomedicine and bioinformatics research, including medical image analysis, healthcare informatics, and cancer genomics. Lots of prediction and mining works were required on the medical data, such as tumor images, electronic medical records, microarray, and GWAS (Genome-Wide Association Study) data. Therefore, a growing number of data mining algorithms were employed in the prediction tasks of computational biology and biomedicine.

Advanced data mining techniques have also been developed quickly in recent years. Several impacted new methods were reported in the top journals and conferences. For example, affinity propagation was published in Science as a novel clustering algorithm. Recently, deep learning seems to be suitable for big data and is becoming the next hot topic. Parallel mechanism is also developed by the scholar and industry researchers, such as Mahout. A growing number of computer scientists are devoted to the advanced large scale data mining techniques. However, application in biomedicine has not fully been addressed and fell behind the technique growth.

This special issue targeted the recent large scale data mining techniques together with biomedicine application and provided a platform for researchers to exchange their innovative ideas and real biomedical data. We have received 25 manuscripts from Asia, Europe, and America, of which 21 papers were accepted. We categorize three subtopics for our special issue.

The first part contains 5 papers that are related to biomedicine images. The paper "Convolutional Deep Belief Networks for Single-Cell/Object Tracking in Computational Biology and Computer Vision" proposed a convolutional deep belief network based architecture to dynamically learn the most discriminative features from data for both single-cell and object tracking in computational biology, cell biology, and computer vision. The paper "Objective Ventricle Segmentation in Brain CT with Ischemic Stroke Based on Anatomical Knowledge" proposed detection system of ischemic stroke in CT, which can exclude the stroke regions from segmentation result with a combined segmentation strategy. The paper "Segmentation of MRI Brain Images with an Improved Harmony Searching Algorithm" proposed a modified algorithm to improve the efficiency of the algorithm. First, a rough set algorithm was employed to improve the convergence and accuracy of the HS algorithm. Then, the optimal value was obtained using the improved HS algorithm. The optimal value of convergence was employed as the initial value of the fuzzy clustering algorithm for segmenting magnetic resonance imaging (MRI) brain images. The paper "Functional Region Annotation of Liver CT Image Based on Vascular Tree" proposed a vessel-tree-based liver annotation method for CT images based on the topological graph. A hierarchical vascular tree is constructed to divide the liver into eight

segments according to Couinaud classification theory and thereby annotate the functional regions. The paper "Robust Individual-Cell/Object Tracking via PCANet Deep Network in Biomedicine and Computer Vision" proposed a robust feature learning method for robust individual-cell/object tracking, which constructed a discriminative appearance model via a PCANet deep network without large scale pretraining.

The second part contains 12 papers on bioinformatics. The paper "A Metric on the Space of Partly Reduced Phylogenetic Networks" proposed a polynomial-time computable metric on the space of partly reduced phylogenetic networks based on the equivalent nodes, whose space is much closer to the space of rooted phylogenetic networks than the others. The paper "Statistical Approaches for the Construction and Interpretation of Human Protein-Protein Interaction Network" established a reliable human protein-protein interaction network and developed computational tools to characterize a protein-protein interaction (PPI) network, where confidence measures were assigned to each derived interacting pair and account for the confidence in the network analysis. The paper "*In Silico* Prediction of Gamma-Aminobutyric Acid Type-A Receptors Using Novel Machine-Learning-Based SVM and GBDT Approaches" proposed a machine-learning-based method for GABAARs prediction at high and low identity data, which sufficiently captured features only from the protein (GABAARs and non-GABAARs) sequence information based on 188-dimensional algorithm and made predictions by Gradient Boosting Decision Tree, Random Forest, libSVM, and k-NN classifiers. By integrating gene expression and DNA methylation data, the paper "Uncovering Driver DNA Methylation Events in Nonsmoking Early Stage Lung Adenocarcinoma" proposed a bioinformatics pipeline of differential network analysis to uncover driver methylation genes and responsive DNA methylation-mediated modules contributing to tumorigenesis. The computational pipeline successfully identified driver epigenetic events in nonsmoking early stage of lung adenocarcinoma. The paper "Identification of Bacterial Cell Wall Lyases via Pseudo Amino Acid Composition" proposed a computational method for Bacterial Cell Wall Lyase prediction, which can obtain optimal features from pseudo amino acid composition by using ANOVA-based feature selection technique. The paper "Recombination Hotspot/Coldspot Identification Combining Three Different Pseudocomponents via an Ensemble Learning Approach" proposed a new computational predictor for recombination hotspot identification only based on the DNA sequences, which combined three kinds of features via an ensemble learning technique. It would be a useful tool for DNA sequence analysis. The paper "Analysis of Important Gene Ontology Terms and Biological Pathways Related to Pancreatic Cancer" investigated the pancreatic cancer by extracting important related GO terms and KEGG pathways. The enrichment theory of GO and KEGG pathway was adopted to encode the validated genes and other genes. And the mRMR method was used to analyze the importance of each GO term and KEGG pathway. Furthermore, the obtained GO terms and KEGG pathways were extensively analyzed. The paper "Constructing Phylogenetic Networks Based on the Isomorphism of Datasets" researched the commonness of the methods based on the incompatible graph, the relationship between incompatible graph and the phylogenetic network, and the topologies of incompatible graphs. The paper "A Novel Peptide Binding Prediction Approach for HLA-DR Molecule Based on Sequence and Structural Information" proposed a novel prediction method for MHC II molecules binding peptides, which calculated sequence similarity and structural similarity between different MHC II molecules and produced a combined similarity score to predict binding cores. The paper "ProFold: Protein Fold Classification with Additional Structural Features and a Novel Ensemble Classifier" proposed a machine learning method for protein fold classification, which imports protein tertiary structure in the period of feature extraction and employs a novel ensemble strategy in the period of classifier training. The paper "An Approach for Predicting Essential Genes Using Multiple Homology Mapping and Machine Learning Algorithms" aimed to test whether optimizing the weight coefficients by the machine learning method could improve the accuracy of their previously proposed evolutionary feature based model, which had shown the best prediction among all published algorithms for predicting bacterial essential genes, and finally the adaption achieved a small improvement. The paper "Positive-Unlabeled Learning for Pupylation Sites Prediction" employed PU learning for predicting pupylation sites and got better performance than traditional classifiers.

The third part contains 4 papers and focuses on the computational medicine. The paper "Optimization to the Culture Conditions for Phellinus Production with Regression Analysis and Gene-Set Based Genetic Algorithm" proposed an optimal method for Phellinus production, where regression model was obtained by sampling data and gene-set based genetic algorithm was applied to find optimized factors for Phellinus production. The paper "Depth Attenuation Degree Based Visualization for Cardiac Ischemic Electrophysiological Feature Exploration" implemented a human cardiac ischemic model and revealed the hidden cardiac biophysical behavior under the ischemic condition by the depth attention degree based optic attenuation model, which effectively explored the important features of interest of the heart under the pathological condition with complex electrophysiological context and is fundamental in analyzing and explaining biophysical mechanisms of cardiac functions for the doctors and medical staffs. The paper "Analysis and Classification of Stride Patterns Associated with Children Development Using Gait Signal Dynamics Parameters and Ensemble Learning Algorithms" computed the sample entropy and average stride interval parameters to quantify the gait dynamics of children with different age groups and used the AdaBoost.M2 and Bagging ensemble learning algorithms to effectively perform gait pattern classifications. The paper "A Computational Method for Optimizing Experimental Environments for *Phellinus igniarius* via Genetic Algorithm and BP Neural Network" used training data to build a neural network model, which acts as the fitness function for further optimal condition finding with genetic algorithm.

To conclude, papers in this special issue cover several emerging topics of scalable data mining techniques and

applications for biomedicine or bioinformatics. We highly hope this special issue can attract concentrated attentions in the related fields.

Acknowledgments

We would like to thank the reviewers for their efforts to guarantee the high quality of this special issue. Also, we thank all the authors who have contributed to this special issue. The work was supported by the Natural Science Foundation of China (no. 61370010).

Quan Zou
Dariusz Mrozek
Qin Ma
Yungang Xu

Hindawi Publishing Corporation
BioMed Research International
Volume 2017, Article ID 8690892, 11 pages
http://dx.doi.org/10.1155/2017/8690892

Research Article

Objective Ventricle Segmentation in Brain CT with Ischemic Stroke Based on Anatomical Knowledge

Xiaohua Qian,[1] Yuan Lin,[2] Yue Zhao,[1] Xinyan Yue,[3] Bingheng Lu,[4] and Jing Wang[4]

[1]*College of Electronic Science and Engineering, Jilin University, Changchun 130012, China*
[2]*Division of Research and Innovations, Carestream Health, Inc., Rochester, NY 14615, USA*
[3]*Affiliated Hospital of the Changchun University of Chinese Medicine, Changchun 130021, China*
[4]*Collaborative Innovation Center of High-End Manufacturing Equipment, Xi'an Jiaotong University, Xi'an 710054, China*

Correspondence should be addressed to Bingheng Lu; bhlu@xjtu.edu.cn and Jing Wang; wjwjggg@gmail.com

Received 3 June 2016; Revised 23 August 2016; Accepted 15 December 2016; Published 7 February 2017

Academic Editor: Dariusz Mrozek

Ventricle segmentation is a challenging technique for the development of detection system of ischemic stroke in computed tomography (CT), as ischemic stroke regions are adjacent to the brain ventricle with similar intensity. To address this problem, we developed an objective segmentation system of brain ventricle in CT. The intensity distribution of the ventricle was estimated based on clustering technique, connectivity, and domain knowledge, and the initial ventricle segmentation results were then obtained. To exclude the stroke regions from initial segmentation, a combined segmentation strategy was proposed, which is composed of three different schemes: (1) the largest three-dimensional (3D) connected component was considered as the ventricular region; (2) the big stroke areas were removed by the image difference methods based on searching optimal threshold values; (3) the small stroke regions were excluded by the adaptive template algorithm. The proposed method was evaluated on 50 cases of patients with ischemic stroke. The mean Dice, sensitivity, specificity, and root mean squared error were 0.9447, 0.969, 0.998, and 0.219 mm, respectively. This system can offer a desirable performance. Therefore, the proposed system is expected to bring insights into clinic research and the development of detection system of ischemic stroke in CT.

1. Introduction

Computed tomography (CT) is generally used to assess patients with acute ischemic stroke in America, because of its faster speed, the better contrast of bones and blood, and the lower cost than magnetic resonance images (MRI). The ischemic stroke and cerebrospinal fluid (CSF) regions have a similar appearance in CT images; thus, accurate ventricle segmentation can significantly facilitate ischemic stroke region localization and is an indispensable step for the development of computer-aided detection (CAD) for acute ischemic stroke.

Several state-of-the-art methods have been proposed to segment ventricles in MRI [1], including active contour-based methods [2–4], fuzzy schemes [5, 6], and probability methods [7, 8]. However, these methods may be inappropriate to work on CT images, since there are lower contrast, higher noise level, and larger slice thickness in brain CT images.

Only little literature on the segmentation of brain CT images has been published. For example, Wei et al. proposed a segmentation scheme based on 2D Otsu thresholding approach [9]. Lee et al. applied the k-means and expectation maximization clustering to segment CT images [10]. Another method by Chen et al. was based on a Gaussian mixture model [11]. Gupta et al. integrated the adaptive threshold, connectivity, and domain knowledge to classify the cerebrospinal fluid, white matter, and gray matter on CT images [12]. These methods mentioned above were not designed specifically for ventricle segmentation and were not validated on the images with severe abnormalities. Chen et al. developed a ventricular segmentation system by combining low-level segmentation and high-level template matching [13]. Similarity, Liu et al. proposed a model-guided segmentation for ventricle region [14]. The two methods are both based on the template or model scheme for ventricle extraction in CT. Since these templates were yielded from the MRI brain

image and registration was linear, the templates only provided a rough mask for the ventricle segmentation. Therefore, it is still challenging for these two methods to exclude stroke regions from segmentation results. Qian et al. proposed a level set model to segment CSF, but the result includes the stroke regions [15]. This study will improve the methods and extensively validate our previous work [16].

The significant difficulty of the accurate ventricle segmentation is to deal with CT images of patients with ischemic stroke. Some of the stroke regions and ventricles are connected and have similar intensities. To address this challenge, we developed an objective segmentation strategy of brain ventricles in unenhanced CT with ischemic stroke. We applied the following three schemes to exclude the stroke regions from segmentation results:

(1) We took the largest three-dimensional (3D) connected component in a preliminary segmentation as the ventricular region, removing the lesion or other regions without the 3D connectivity relationship with the ventricle, since the initial segmentation result contained not only the ventricle but also some non-ventricular regions, such as lesion or CSF.

(2) The large stroke regions were removed by the image difference method. The large stroke areas tend to close the brain edge, and their intensities were generally lower than that of the main parts of ventricles. Thus, the stroke region can be extracted by the difference between segmentation results from two optimal threshold values.

(3) The small stroke regions were removed by the adaptive template algorithm. The adaptive template was directly generated from the corresponding image itself based on the big intensity difference between the main part of the ventricle and the brain parenchyma. This template did not contain the whole ventricle but did cover the main part of the ventricular region. Thus, we applied this template to remove the small lesions around the main part of the ventricle, which was not subjected to the registration. Another effect was that the exclusion of these small lesions might break the connectivity relationship between the lesion regions and the ventricular region in 3D space.

2. Materials and Methods

As shown in Figure 1, the automated ventricle segmentation method is comprised of two phases, that is, alignment phase (Section 2.2) and segmentation phase (Section 2.3). In the alignment phase, the light curves/segment of the brain was detected to determine the midsagittal line for each slice. We then aligned the midsagittal line (MSL) with the vertical line of each slice to achieve brain alignment. In the segmentation phase, we first estimated the intensity range of the ventricle region based on clustering technique, connectivity, and domain knowledge. An image difference algorithm was developed to identify and remove the large stroke regions in the initial segmentation. The remaining small stroke region was further excluded by an adaptive template of the ventricle. Finally, the largest 3D connectivity of the segmented ventricle was employed to refine the segmentation result.

2.1. Dataset. We tested the proposed method on 50 CT scans of patients with ischemic stroke in this study. This dataset was collected from Jilin University Medical Center using CT scanners (Light Speed 16, GE Medical System) with an X-ray tube voltage of 120 kVp. Each patient has 14 slices with the thickness of 5 mm in this study. The matrix size of each slice is 512×512 pixels, and the pixel size is 0.426 mm with a 16-bit gray level. The 50 patients were composed of 29 males and 21 females, and their average age is 57 years with the range between 41 years and 76 years. We established a reference standard of ventricle for evaluation of segmentation result. A medical physicist (XQ, eight years of experience) manually delineated the ventricle boundaries for all the slices on an LCD screen as the reference standard to assess the accuracy of segmentation results.

2.2. Alignment of the Brain Image. Prior to the alignment of brain image, the skull was stripped by a threshold method since CT number of bone tissues are consistently higher than brain tissues. Generally, the CT number of soft tissue is less than 60 Hounsfield units (HU) (such as 1–12 HU of ventricle, 25–38 HU of white matter, and 35–60 HU of gray matter), while average CT intensity is 1000 HU for bones. Thus, we extracted the skull using a fixed threshold of 100 HU. The region inside the skull was considered as brain region and the region outside the skull served as background.

After the extraction of the brain, the inclination angle and position were corrected by aligning MSL with the vertical centerline of each slice. The determination of MSL is a key step in this alignment. Since the falx cerebri (i.e., narrow light curve/segment) presents on about 30% images, we applied the falx cerebri as a reference to identify the MSL. Therefore, we utilized two steps to achieve alignment of the brain, including (1) detection of a light curve in the brain and (2) affine transformation based on MSL.

2.2.1. Detection of Light Curves in Brain. Figure 2 shows the schematic diagram of light curve detection. To accelerate the detection, we defined a rectangle region of interest (ROI), whose size was chosen to include the light curves to be detected. We selected a smallest minimum bounding rectangle of the brain area in the whole scan and then defined the half width of this rectangle as the width of the ROI. The height of the ROI was taken the default value of 512. Figure 2(b) shows the rectangle ROI of the brain.

CT brain image has a high level of noise. The common filtering may blur the weak edge, making detection of the light curve difficult. The light curve has a slight angle with the vertical direction; however, it is still regarded as vertical. Thus, we designed a one-dimensional (7×1) Gaussian filter with the variance of 2 to smooth the image along the vertical direction, which can preserve the edge information of the light curve in the horizontal direction as shown in Figure 2(c).

We then design a horizontal Laplacian detection mask, that is, $[0.5, 0, 1, 0, 0.5]$, to detect the light curve, since the vertical strip included more edge points of the light curve

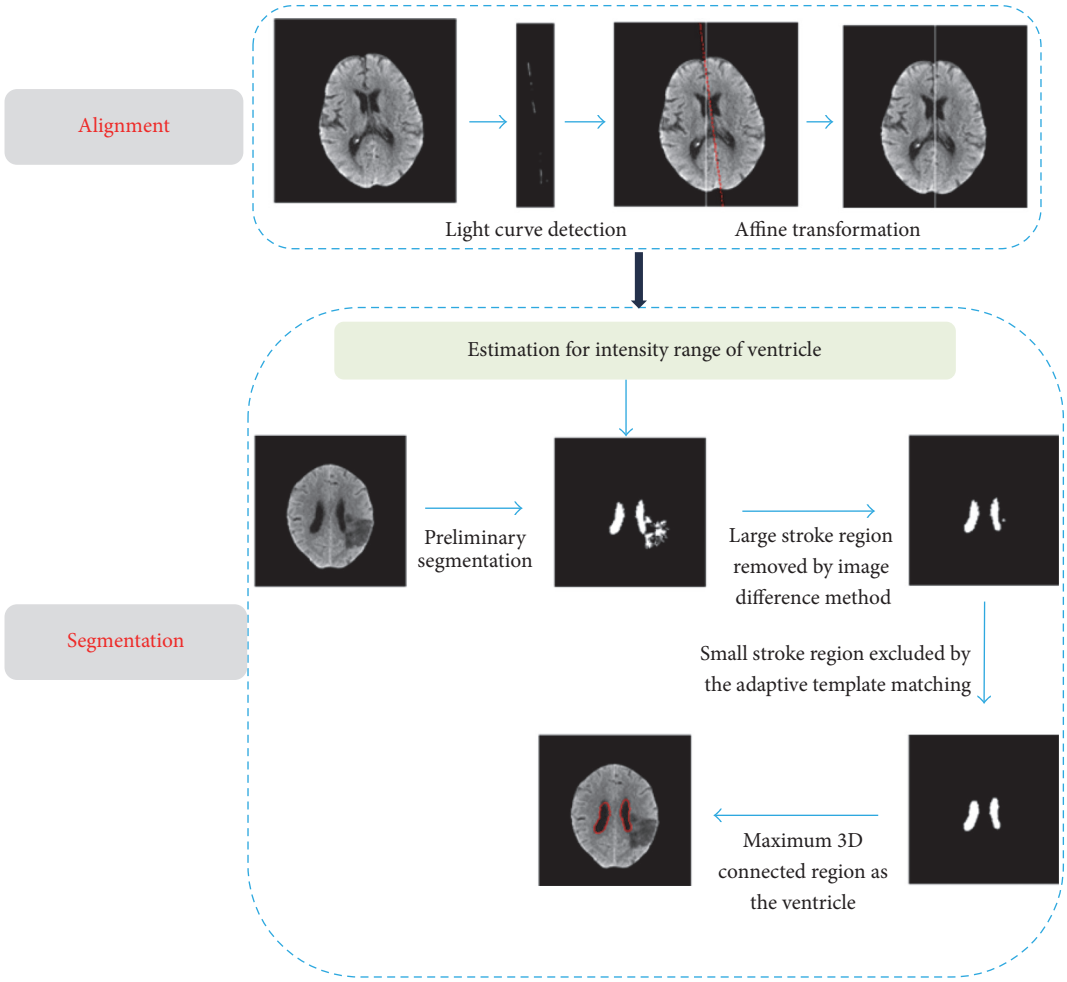


Figure 1: Schematic framework for segmentation of the brain ventricle in CT of patients with ischemic stroke.

than other places. With the Laplacian image (Figure 2(c)), we employed an adaptive threshold to yield an edge map, as shown in Figure 2(d). We empirically set the threshold as the average value with 2.5 multiple of the standard deviation of the Laplacian image.

After that, we erased the small unconnected noise point clouds in the edge map based on 3D connectivity. The noise points in edge map may negatively affect the subsequent 3D fitting of the middle sagittal plane. However, the 3D connected volume of these noise points is small; thus, we can remove them with a threshold in 3D connected volume. In our experiment, we applied thirty pixels as the threshold to obtain the clean edge map of light curve (Figure 2(f)). Figure 2(g) shows the 3D edge map of light curves.

2.2.2. Affine Transformation Based on MSL. To obtain the precise MSL, we first fitted a middle sagittal plane in 3D Euclidean space through a set of edge segments of light curves using least-squares fitting approach. Let (x_i, y_i, z_i) be a point of edge segments, which has totally M points and $i = 1, 2, \ldots, M$. So, the optimum fitting plane can be achieved by the following formulation as

$$(a^*, b^*, c^*) = \arg\min_{(a,b,c)} \sum_{i=1}^{M} (z_i - ax_i - by_i + c)^2. \quad (1)$$

The MSL of each slice was defined as the intersection line between the image and middle sagittal plane. Let z_i denote the ith slice of 3D image, and we can obtain the MSL of this slice as

$$ax - by = z_i - c. \quad (2)$$

The determined MSL was shown in Figure 3(a). Finally, we aligned the MSL of the brain with the vertical center line of a slice using the affine transformation defined by

$$\begin{aligned} x' &= (x - x_0)\cos\theta + (y - y_0)\sin\theta + x_0 \\ y' &= (x - x_0)\sin\theta + (y - y_0)\cos\theta + y_0, \end{aligned} \quad (3)$$

where (x_0, y_0) is the center point of the vertical center line of a slice and θ is the inclination angle between the MSL

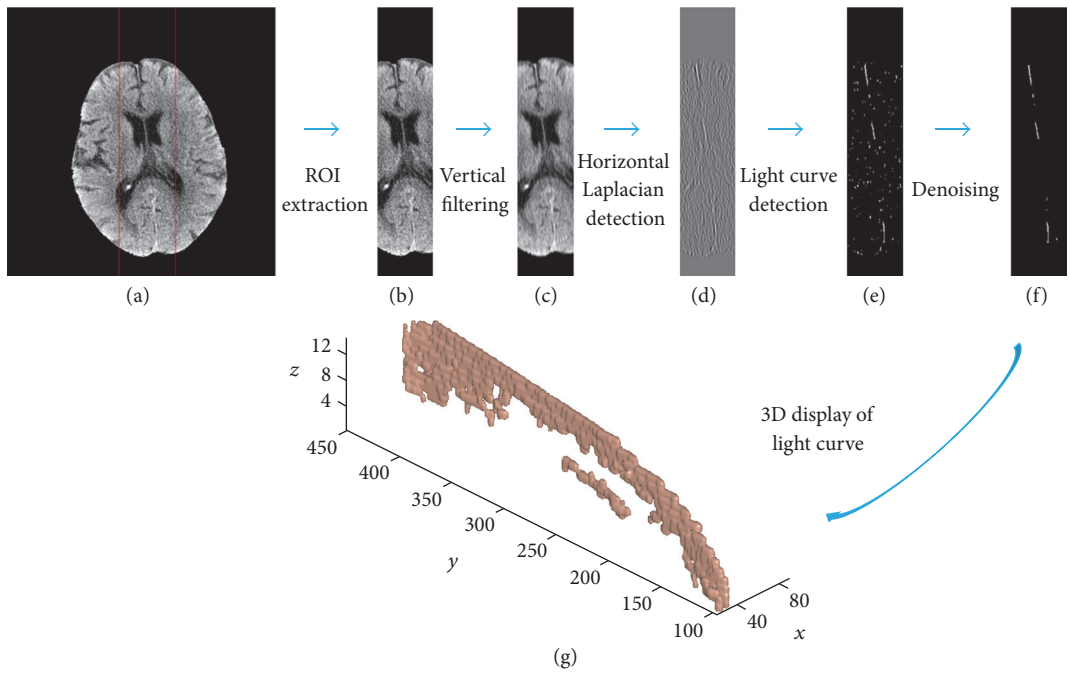


FIGURE 2: Diagram of light curve/segment detection: (a) original image without skull; (b) the ROI of the light curve; (c) the vertical filtered ROI; (d) the Laplacian image; (e) the detected light curve; (f) denoising light curve; (g) 3D display of light curves: z-axis represents the slice number; x- and y-axes denote the pixel number.

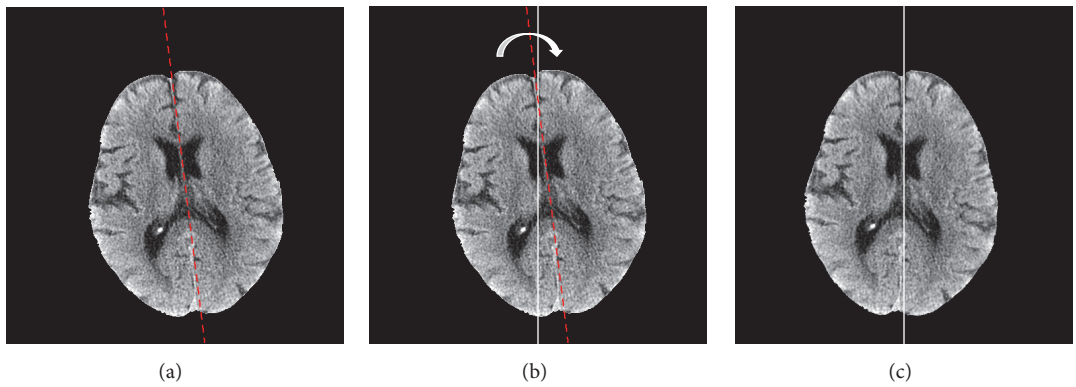


FIGURE 3: Alignment of the brain image: (a) original image with the midsagittal line (MSL, dashed line); (b) the vertical center line of a slice with white color and the MSL; (c) aligned brain image.

and vertical center line. Figures 3(b) and 3(c) show that the inclination angle and position of the brain were corrected.

2.3. Segmentation of the Ventricle. In the phase of ventricle segmentation, we focused on excluding the stroke area in the ventricle segmentation result. The flowchart was shown in Figure 4.

2.3.1. Parameter Estimation for the Ventricle. Prior to the segmentation of ventricle, we estimated parameters of the intensity distribution of the ventricle. We first applied the K-means algorithm ($K = 2$) on the 3D images for stratification of the brain image and took the largest 3D connected component of low-intensity category as the ventricle. Then, an estimation method based on connectivity and domain knowledge from the literature [8] was utilized to compute the intensity distribution of different tissues. Specifically, we tracked the slop of the histogram corresponding to the 3D largest connected component in rough intensity range of ventricle to determine a critical intensity, which serves as an initial classifier of cerebral spinal fluid and white matter. Thresholds of cerebral spinal fluid, white matter, and gray matter are optimally derived to minimize spatial overlap errors in different tissue types. In this study, ventricular intensity range of $[V_{\min} \quad V_{\max}]$ will be adopted to extract the ventricular region.

2.3.2. Preliminary Segmentation for the Ventricle Based on Estimated Parameters. $V_{\max}$, the estimated maximum of ventricular intensity range, was applied as a threshold value for

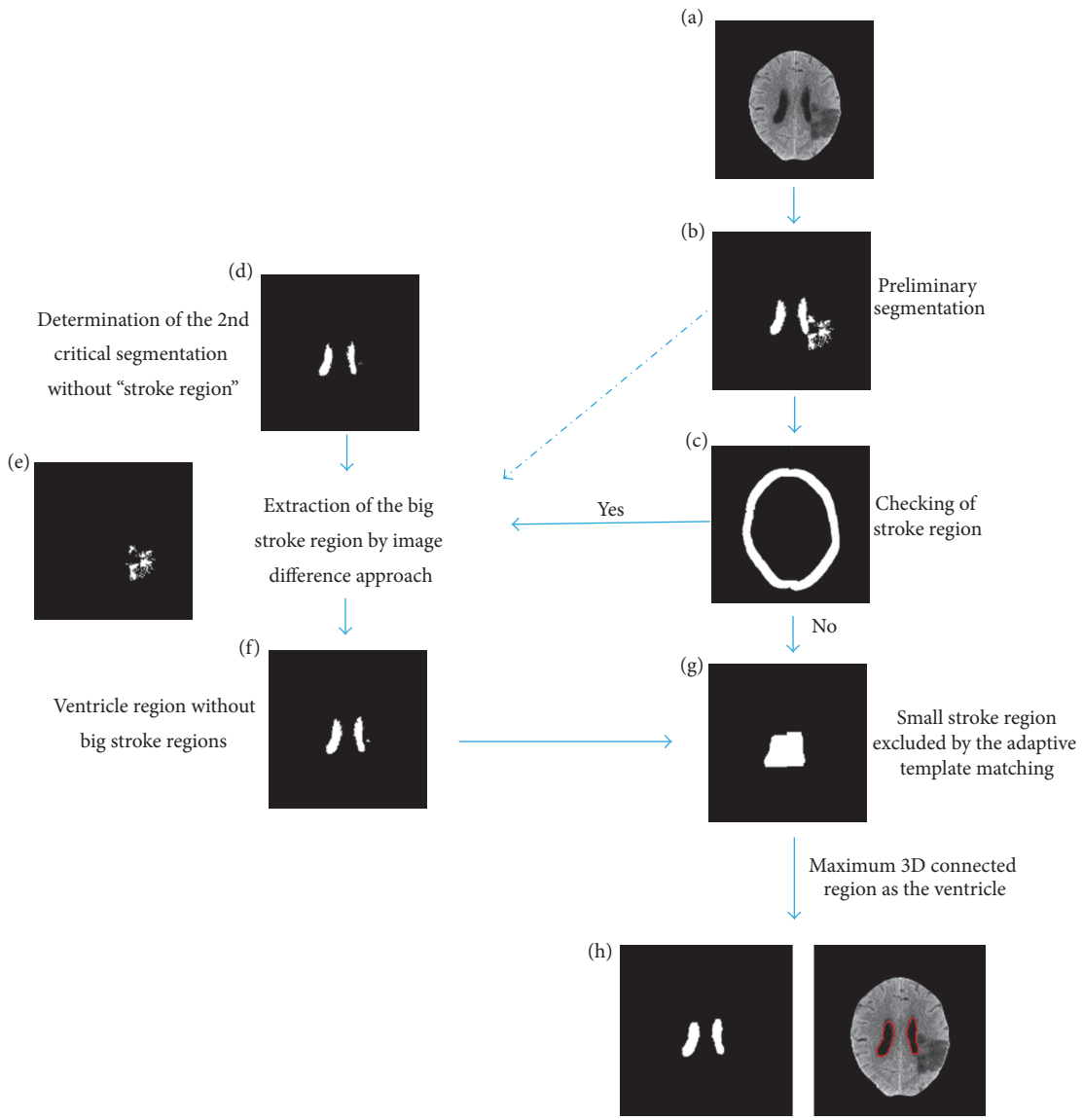


Figure 4: Flowchart of the exclusion of stroke area in the ventricular segmentation result.

preliminary segmentation of the ventricle. If the intensity range of the stroke is greater than V_{max}, the preliminary segmentation is a good result. Whereas, if the intensity range of the stroke is less than V_{max}, the segmentation result may be unacceptable, since it may also contain some stroke regions.

Then, we utilized the 3D connectivity of the preliminary segmentation result to obtain the largest volume as the initial segmentation of the ventricle. The stroke regions or noise areas without the 3D connectivity to the ventricle could be excluded by this step. Figure 4(b) shows that the large stroke regions are connected to the ventricle in the segmentation.

2.3.3. Detection of the Big Stroke Regions. Since big stroke regions are mainly related to the anterior cerebral artery or middle cerebral artery, these stroke regions are mostly closed to the brain edge. Thus, we proposed a brain edge checking algorithm to determine whether the big stroke regions exist in the segmentation result. An annular region of the brain edge was defined to detect the objects. Assumed that the minimum side length of the minimum bounding rectangle of the brain was L_{min}, the width of the annular region could be calculated by $0.15 \times L_{min}$ to avoid some parts of the ventricle falling within the annular region. The mask of the brain edge annular region was shown in Figure 4(c). Thus, if the objective area was greater than the threshold, we labeled it as the stroke region. The threshold was empirically selected as 20 pixels to allow the presence of noise.

2.3.4. Determination of the Big Stroke Regions. We proposed an image difference technique based on the heuristic searching algorithm to extract the big stroke regions, which were successfully detected in the preliminary segmentation by the edge checking method. This image difference technique essentially applied the difference between two segmentation

results by different threshold values for determining the stroke regions. We first defined the critical threshold value (i.e., T_{critical}). If a threshold was greater than T_{critical}, the stroke regions in the segmentation result of this threshold could be detected by the edge analysis method; whereas, if the threshold was smaller or equal to T_{critical}, none stroke region could be detected. We then obtained the stroke regions by

$$\text{PA} \approx f\left(V_{\max}\right) - g\left(f\left(T_{\text{critical}}\right)\right), \tag{4}$$

where $f(*)$ was the threshold method; $g(*)$ represented the subsequent refine algorithms, such as morphology method; and PA represented the stroke regions. So, we obtained the ventricle areas:

$$f\left(V_{\max}\right) - g\left(\text{PA}\right). \tag{5}$$

The vital step in the image difference method is to determine the critical threshold value T_{critical}. We applied the gold searching method and the edge checking method to obtain the T_{critical} in range $[V_{\min} \quad V_{\max}]$.

2.3.5. Exclusion of the Small Stroke Regions. Some small stroke regions may still present in the segmentation result from the image difference approach. To address this problem, we developed an adaptive template matching approach, which applied the mask of the main part of the ventricle to exclude the remaining small stroke regions. The template was generated from each image. It did not contain the whole ventricle but covers the main part of the ventricle.

Figure 5 shows a sectional view of the gray-scale map for a brain image. The intensity difference between the ventricle and brain parenchyma was around 20 intensity values, while the transition area was only 6 to 7 pixels. Thus, we applied $V_{\min}$ as a threshold for ventricle segmentation and took the 3D largest connected region as the ventricle, as shown in Figure 6(b). The ventricle segmentation, merely containing the right and left lateral ventricles and without the 3rd and 4th ventricle, was adaptively selected as the templates. To ensure that the template covers the ventricle, we conducted some morphological analysis, including closed operation and expansion operation. The generated template was shown in Figures 6(c) and 6(d).

After these steps, we linearly registered the template with the corresponding segmentation. The objects within the template served as the ventricle so that the remaining small stroke areas could be excluded from the segmentation results.

2.3.6. Refinement of the Ventricular Segmentation. We employed connected component labeling to the segmented ventricle region. The largest volume served as the ventricular. We then removed the calcification regions in the results and smoothed the ventricular edges using the morphologically closed operation.

2.4. Evaluation of the Segmentation Method. We applied four measures, including Dice metric (Dice), root mean squared error (RMSE), reliability ($\mathscr{R}$)[28], and correlation coefficient (R), to assess the performance of the proposed segmentation method. The four measures are defined as follows.

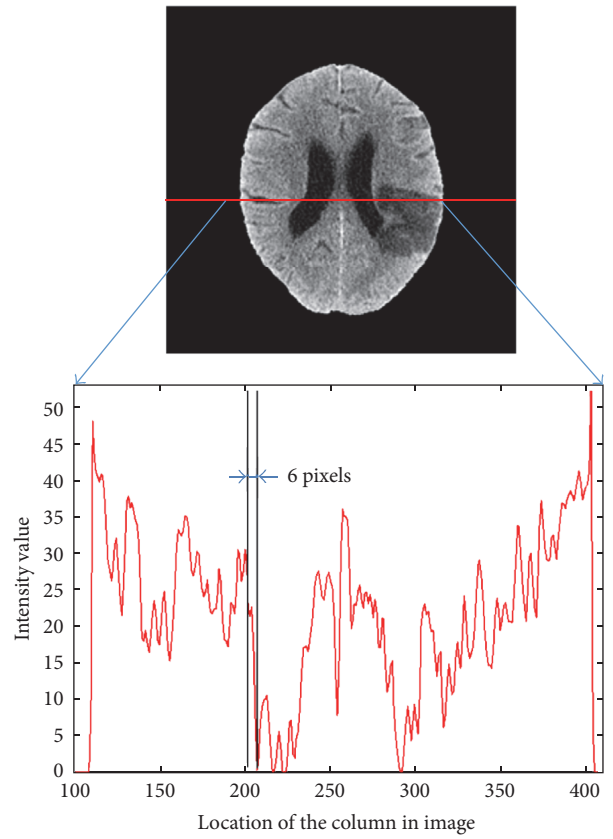


Figure 5: A sectional view of the gray-scale map for brain image.

(1) Dice Metric. Let V_s represent the automatically segmented volume and V_r represent the manual segmentation (i.e., reference standard). The Dice is defined as

$$\text{Dice} = \frac{2V_s \cap V_r}{V_s + V_r}. \tag{6}$$

The value of Dice is between 0 and 1. Higher Dice indicates better overlap between segmented volumes and the reference standard.

(2) Root Mean Squared Error. The RMSE calculates the distance between the corresponding points on the automatically segmented and reference boundaries, defined by

$$\text{RMSE} = \left(\frac{1}{N}\sum_{i=1}^{N}\left(x_{s,i} - x_{r,i}\right)^2 + \left(y_{s,i} - y_{r,i}\right)^2\right)^{1/2}, \tag{7}$$

where $(x_{s,i}, y_{s,i})$ is a point on the segmented boundary and $(x_{r,i}, y_{r,i})$ is the closest point to $(x_{s,i}, y_{s,i})$ on the reference boundary. The lower RMSE, the better performance.

(3) Reliability. The reliability function is used to assess the reliability of segmentation method, defined as

$$\mathscr{R}(d) = \frac{\text{Number of volumes segmented with Dice} > d}{\text{Total number of volumes}}, \tag{8}$$

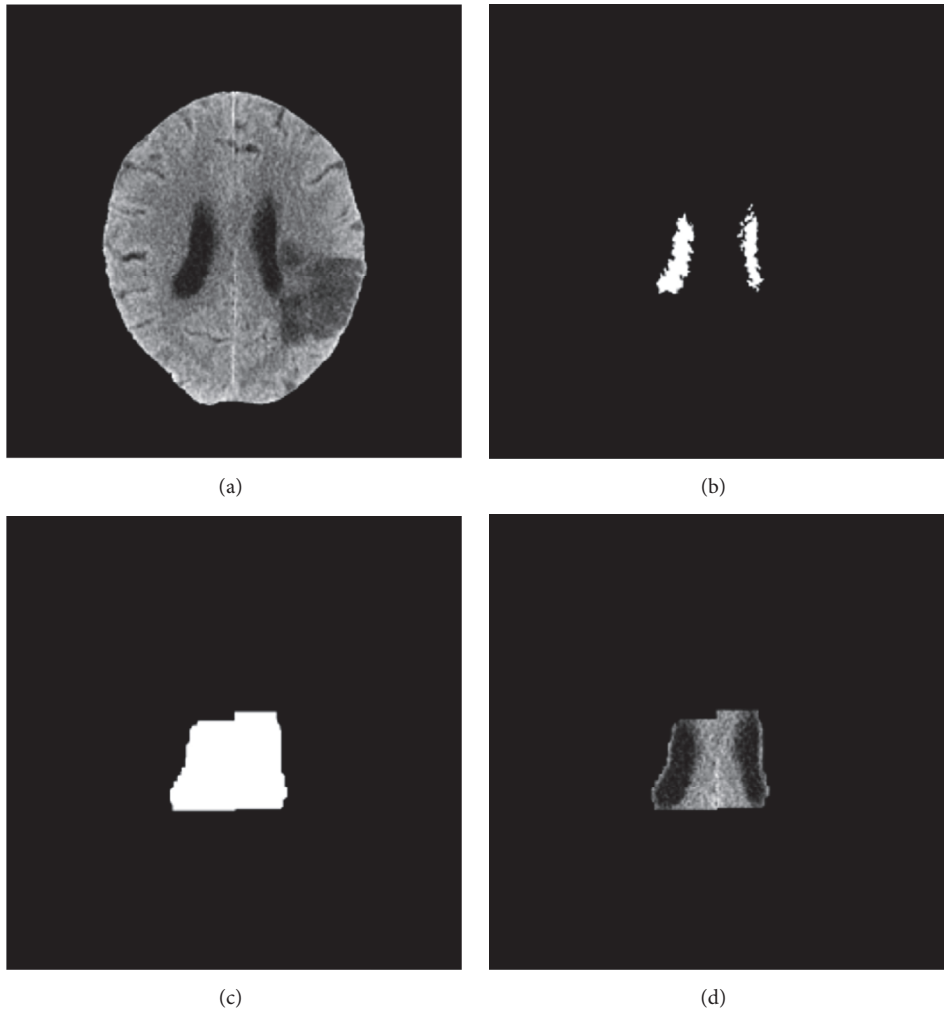


Figure 6: Generation of the template for ventricle: (a) original image; (b) initial segmentation result; (c) the generated template; (d) the corresponding brain area in the template.

where $d \in [0, 1]$. $\mathscr{R}(d)$ represents the reliability in yielding Dice d.

(4) Correlation Coefficient. R between V_s and V_r is used to assess the quality of a least-squares fitting, given by

$$R = \frac{n\sum_{i=1}^{n} V_{s,i}V_{r,i} - \sum_{i=1}^{n} V_{s,i}\sum_{i=1}^{n} V_{r,i}}{\left(n\sum_{i=1}^{n} V_{s,i}^2 - \left(\sum_{i=1}^{n} V_{s,i}\right)^2\right)^{1/2}\left(n\sum_{i=1}^{n} V_{r,i}^2 - \left(\sum_{i=1}^{n} V_{r,i}\right)^2\right)^{1/2}}. \quad (9)$$

The value of R ranges from 0, no match between the two volumes, to 1, a perfect match.

3. Results

3.1. Qualitative Evaluation. Figure 7 displays the alignment of three representative brain images. The original images were shown in (a). (b) to (d) were the segmented light curve/segment, determined midsagittal line, and the final aligned result, respectively. Only a short light curve segment was detected in the brain image of the first row; however, our algorithm still accurately determined the midsagittal line, which was attributable to 3D fitting of the middle sagittal plane based on segmented light curve/segments. We can find that our alignment algorithm yielded good performance.

Figure 8 shows the results of ventricle segmentation. The original brain image, ventricle segmentation result, and reference standard were shown in (a) to (c), respectively. Although some stroke regions were attached to the ventricle in original images, they were all excluded in the segmentation results. This result means that our proposed segmentation method can obtain satisfactory results on images with ischemic stroke.

3.2. Quantitative Evaluation Results. We quantitatively assessed the ventricle segmentation results using Dice, RMSE, the reliability ($\mathscr{R}$) and correlation coefficient (R). The mean Dice, sensitivity, specificity, and RMSE were 0.9447, 0.969, 0.998, and 0.219, respectively, as shown in Table 1. The analysis

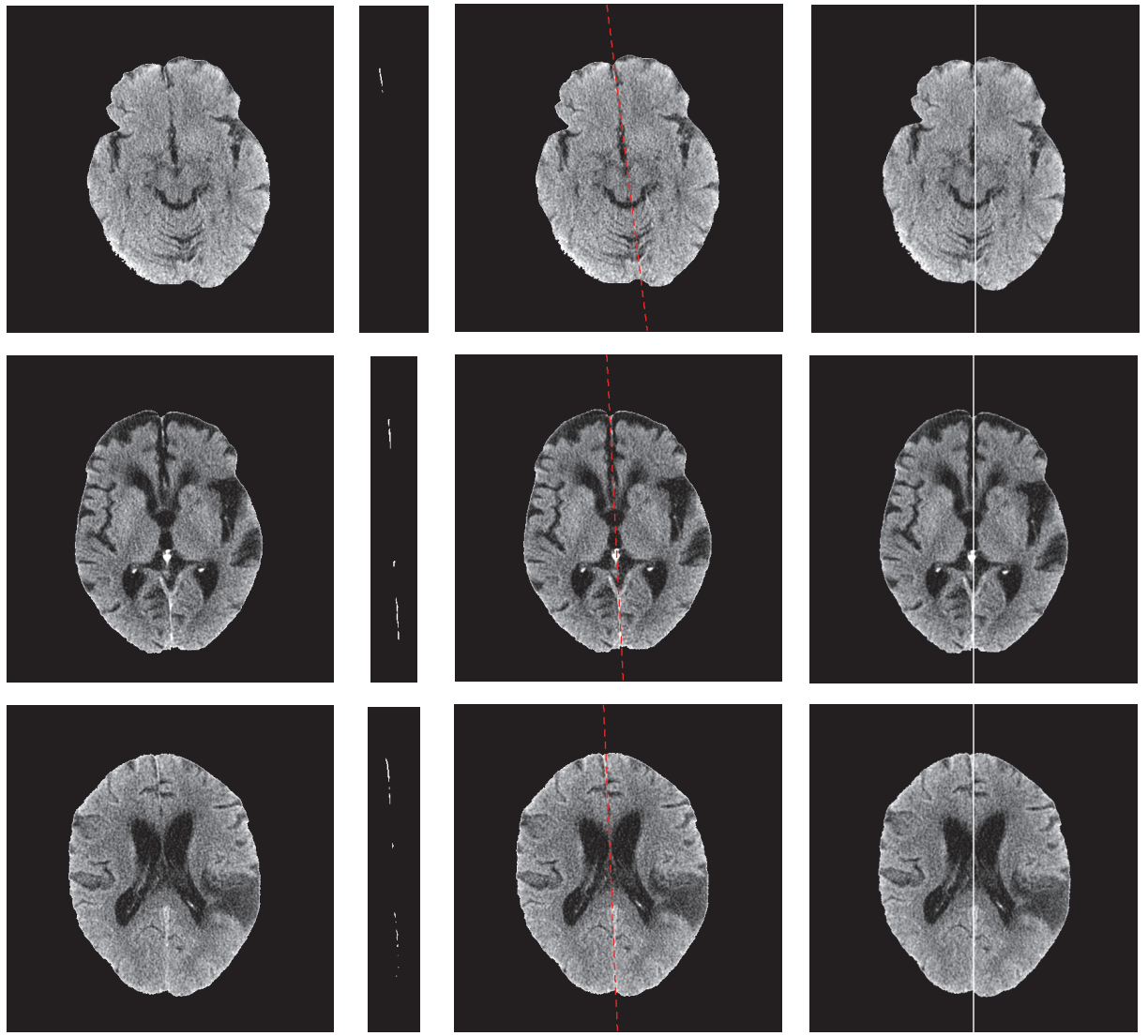


Figure 7: Alignment performance: original image without skull (a); detected light curve/segment (b); determined midsagittal line with red color for each slice based on 3D fitting of light curves (c); aligned brain image, where the white line shows the midline of the image (d).

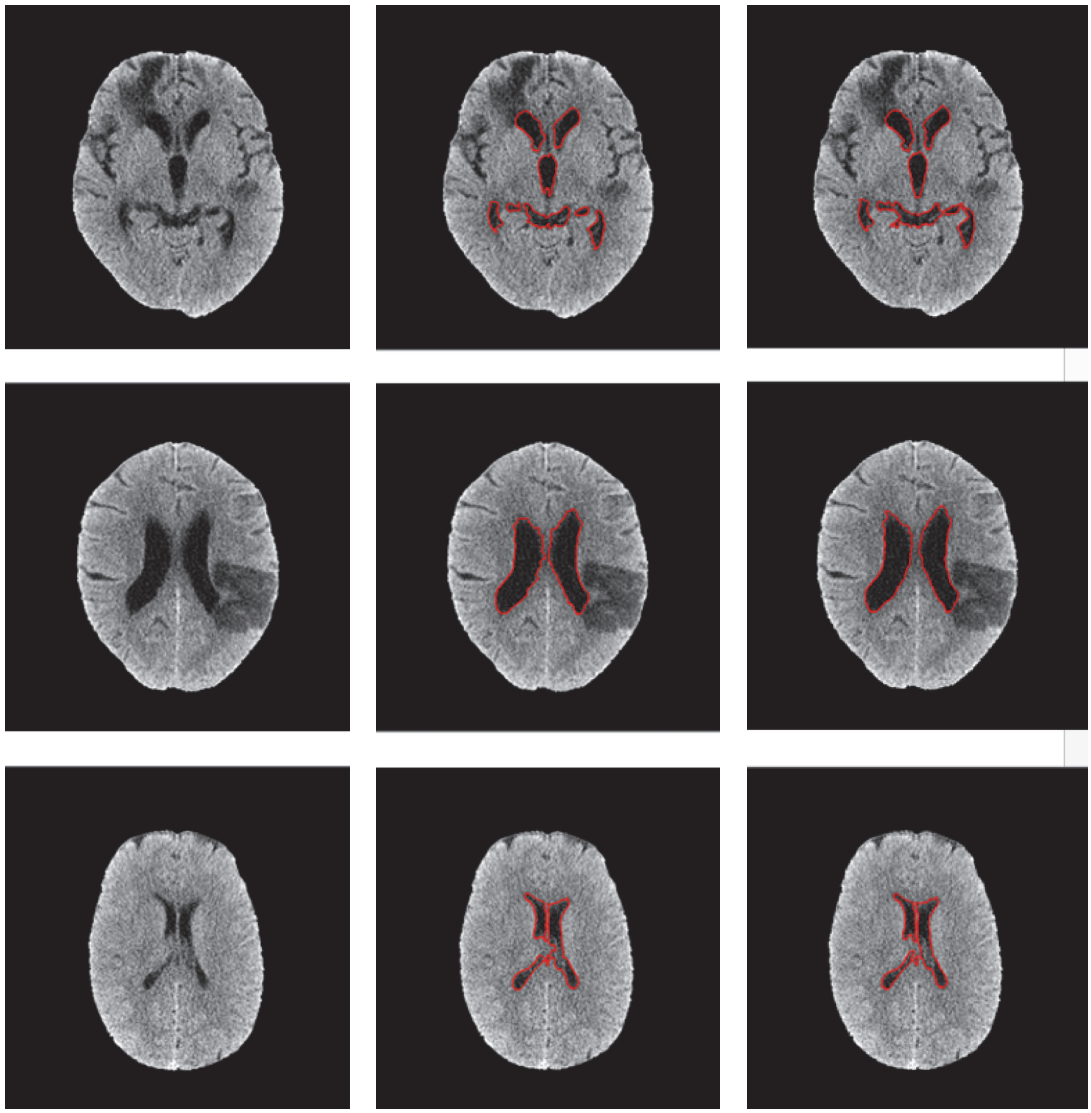
Table 1: Quantitative performance evaluations (Dice, sensitivity, specificity, and RMSE) on 50 cases of patients with ischemic stroke regions.

	Mean	SD	Min	Max
Dice	0.945	0.036	0.801	0.985
Sensitivity	0.970	0.027	0.892	0.997
Specificity	0.998	0.00	0.996	0.999
RMSE (mm)	0.219	0.472	0.007	2.536

results of these metrics confirm the desirable performance of our proposed method.

The proposed method produced a reliability of $\mathscr{R}(0.85) = 0.987$ for ventricle segmentation, which means all these cases have a good agreement (Dice > 0.85). Figure 9(a) plots $\mathscr{R}$ as a function of d ($d \geq 0.78$) for the ventricle segmentation. It further shows the acceptable performance of the proposed method.

The correlation coefficients between automatic segmentation result and reference standard are 0.994. The linear



(a) (b) (c)

Figure 8: Performance of ventricle segmentation: original brain images (a); ventricle segmentation result outlined with red contours (b); contours of the reference ventricle (c).

regression plotted in Figure 9(b), which indicates a close correlation between the results of the proposed method and the reference standard.

4. Discussion

The stroke regions on CT are often adjacent or connected to the ventricle, and their intensities are similar, which makes it highly difficult for accurate segmentation of the ventricle. To achieve this goal, we developed a combined segmentation strategy composed of connectivity, image difference method, and adaptive template method that is developed to exclude stroke regions from the ventricular segmentation result, which constitutes the major strength of our segmentation scheme.

Image difference method was used to extract the large lesion regions. In this approach, the most critical step was to search the critical threshold for obtaining the ventricular segmentation result without stroke regions. This result served as "benchmark ventricular mask," and acted as the subtrahend in the image difference method. However, the edge checking method only worked well for the large stroke regions, so this method was not able to efficiently detect the small stroke areas when they presented in the segmentation result from the critical threshold. If the benchmark ventricular mask contains small stroke areas, these small stroke regions would be left in the final segmentation results. Therefore, the adaptive template method was developed to remove these small stroke regions, which would further break up the connectivity relationship between the lesion regions and the

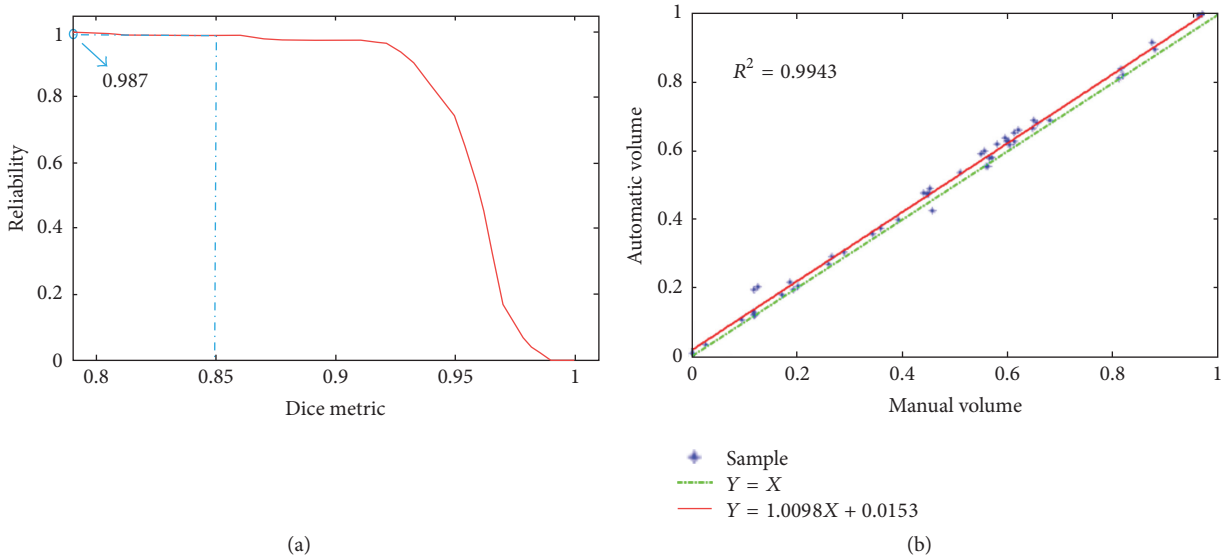


FIGURE 9: (a) The reliability of our method: $\mathscr{R}(0.85) = 0.987$; (b) segmentation volumes of our method versus manual volumes: $R = 0.997$.

ventricular region in 3D space. Finally, we took the largest 3 connected component in the segmentation as the ventricular region to refine the results.

The limitation of this segmentation system is that some small stroke region may still exist in the segmentation result, due to the local property of the adaptive template, which covers the main part of the ventricle. Differentiation of the ventricle and stroke region is a challenging task. In the future, we will combine the prior template of the ventricle and adaptive template to exclude the stroke region in the initial segmentation result. Besides, we will collect more data to validate our proposed segmentation system.

5. Conclusion

The accurate ventricle segmentation is a critical step in the development of CAD for acute ischemic stroke. Since ischemic stroke regions are generally adjacent to the brain ventricle with similar intensity, it is a challenging task to segment ventricle. In this study, we developed an objective segmentation system of brain ventricle in CT. We proposed three different schemes to exclude the stroke regions from initial segmentation, which are the main contributions in this work. The experiments illustrate the proposed segmentation method that can obtain a good performance for segmentation of ventricle in brain CT scans with ischemic stroke, which would significantly facilitate ischemic stroke region localization.

Competing Interests

The authors have no relevant competing interests to disclose.

References

[1] M. A. Balafar, A. R. Ramli, M. I. Saripan, and S. Mashohor, "Review of brain MRI image segmentation methods," *Artificial Intelligence Review*, vol. 33, no. 3, pp. 261–274, 2010.

[2] C. M. Li, R. Huang, Z. H. Ding, J. C. Gatenby, D. N. Metaxas, and J. C. Gore, "A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 2007–2016, 2011.

[3] L. Wang, C. Li, Q. Sun, D. Xia, and C.-Y. Kao, "Active contours driven by local and global intensity fitting energy with application to brain MR image segmentation," *Computerized Medical Imaging and Graphics*, vol. 33, no. 7, pp. 520–531, 2009.

[4] L. Wang, F. Shi, W. Lin, J. H. Gilmore, and D. Shen, "Automatic segmentation of neonatal images using convex optimization and coupled level sets," *NeuroImage*, vol. 58, no. 3, pp. 805–817, 2011.

[5] H. Wang and B. Fei, "A modified fuzzy C-means classification method using a multiscale diffusion filtering scheme," *Medical Image Analysis*, vol. 13, no. 2, pp. 193–202, 2009.

[6] X. Yang and B. Fei, "A multiscale and multiblock fuzzy C-means classification method for brain MR images," *Medical Physics*, vol. 38, no. 6, pp. 2879–2891, 2011.

[7] S. Kumazawa, T. Yoshiura, H. Honda, F. Toyofuku, and Y. Higashida, "Partial volume estimation and segmentation of brain tissue based on diffusion tensor MRI," *Medical Physics*, vol. 37, no. 4, pp. 1482–1490, 2010.

[8] X. Li, L. Li, H. Lu, and Z. Liang, "Partial volume segmentation of brain magnetic resonance images based on maximum a posteriori probability," *Medical Physics*, vol. 32, no. 7, pp. 2337–2345, 2005.

[9] K. Wei, B. He, T. Zhang, and X. Shen, "A novel method for segmentation of CT head images," in *Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering (ICBBE '07)*, pp. 717–720, Wuhan, China, July 2007.

[10] T. H. Lee, M. F. A. Fauzi, and R. Komiya, "Segmentation of CT brain images using K-means and EM clustering," in *Proceedings of the 5th International Conference on Computer Graphics, Imaging and Visualisation, Modern Techniques and Applications (CGIV '08)*, August 2008.

[11] W. Chen and K. Najarian, "Segmentation of ventricles in brain CT images using gaussian mixture model method," in

Proceedings of the ICME International Conference on Complex Medical Engineering (CME '09), April 2009.

[12] V. Gupta, W. Ambrosius, G. Y. Qian et al., "Automatic segmentation of cerebrospinal fluid, white and gray matter in unenhanced computed tomography images," *Academic Radiology*, vol. 17, no. 11, pp. 1350–1358, 2010.

[13] W. A. Chen, R. Smith, S.-Y. Ji, K. R. Ward, and K. Najarian, "Automated ventricular systems segmentation in brain CT images by combining low-level segmentation and high-level template matching," *BMC Medical Informatics and Decision Making*, vol. 9, supplement 1, article S4, 2009.

[14] J. Liu, S. Huang, V. Ihar, A. Wojciech, L. C. Lee, and W. L. Nowinski, "Automatic model-guided segmentation of the human brain ventricular system from CT images," *Academic Radiology*, vol. 17, no. 6, pp. 718–726, 2010.

[15] X. Qian, J. Wang, S. Guo, and Q. Li, "An active contour model for medical image segmentation with application to brain CT image," *Medical Physics*, vol. 40, no. 2, Article ID 021911, 2013.

[16] X. Qian, J. Wang, and Q. Li, "Automated segmentation of brain ventricles in unenhanced CT of patients with ischemic stroke," in *Proceedings of the Medical Imaging 2013: Computer-Aided Diagnosis*, Lake Buena Vista (Orlando Area),Fla, USA, February 2013.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 2979081, 8 pages
http://dx.doi.org/10.1155/2016/2979081

Research Article

Depth Attenuation Degree Based Visualization for Cardiac Ischemic Electrophysiological Feature Exploration

Fei Yang,[1] Lei Zhang,[2] Weigang Lu,[3] Lei Liu,[4] Yue Zhang,[5] Wangmeng Zuo,[5] Kuanquan Wang,[5] and Henggui Zhang[5,6]

[1]*School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264200, China*
[2]*School of Art and Design, Harbin University, Harbin 150086, China*
[3]*Department of Educational Technology, Ocean University of China, Qingdao 266100, China*
[4]*Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China*
[5]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[6]*School of Physics and Astronomy, University of Manchester, Manchester M139PL, UK*

Correspondence should be addressed to Weigang Lu; luweigang@ouc.edu.cn

Received 3 June 2016; Revised 21 September 2016; Accepted 11 October 2016

Academic Editor: Qin Ma

Although heart researches and acquirement of clinical and experimental data are progressively open to public use, cardiac biophysical functions are still not well understood. Due to the complex and fine structures of the heart, cardiac electrophysiological features of interest may be occluded when there is a necessity to demonstrate cardiac electrophysiological behaviors. To investigate cardiac abnormal electrophysiological features under the pathological condition, in this paper, we implement a human cardiac ischemic model and acquire the electrophysiological data of excitation propagation. A visualization framework is then proposed which integrates a novel depth weighted optic attenuation model into the pathological electrophysiological model. The hidden feature of interest in pathological tissue can be revealed from sophisticated overlapping biophysical information. Experiment results verify the effectiveness of the proposed method for intuitively exploring and inspecting cardiac electrophysiological activities, which is fundamental in analyzing and explaining biophysical mechanisms of cardiac functions for doctors and medical staff.

1. Introduction

Cardiac diseases have been the leading cause of death and disability in the world. Evidence has shown that functional abnormity of heart such as the heart failure may lead to the severe cardiac problem with increased mortality [1]. Heart failure manifests insufficient blood flow pumped for delivering oxygen, which generally appears as pulmonary edema and cardiogenic shock [2]. Cardiac researchers and medical staffs have put forward methods to analyze cardiac functional mechanism to understand and treat heart failure. Serpooshan et al. [3] analyzed the structure and function of the failing heart using the biomimetic three-dimensional technology to enhance cardiac healing after injury. Namazi et al. [4] presented an unusual case of amyotrophic lateral sclerosis (ALS) and the cardiac failure was diagnosed at the final stage of the ALS disease. Alickovic and Subasi [5] applied dwt and random forests classifier for analyzing the heart arrhythmia. Keller et al. [6] established a heterogeneous electrophysiological and three-dimensional anatomical model of human atria to explore atrial functional mechanism. Brocklehurst et al. [7] implied the discrete element method (DEM) to investigate the electromechanical mechanism for human atrial tissue. Then, mechanical contractions of cardiac tissues and their corresponding electrical waves' conduction were successfully simulated. Salinet Jr. et al. [8] presented spectral analysis techniques to visualize intracardiac atrial fibrillation (AF) electrograms, helping guide catheter ablation procedures. Aslanidi et al. [9] constructed a 3D virtual human atria model using cell electrophysiological data with detailed DT-MRI anatomy, which provides a valuable way for investigating electrophysiological behavior in the arrhythmic atria during

AF. Zhong et al. [2] discussed the utilization of extracorporeal membrane oxygenation (ECMO) for cardiogenic shock. Sala et al. [10] presented a new transgenic mouse model of to replicate the clinical findings of heart failure.

Ventricle fibrillation (VF) is a serious cardiac functional abnormality that can lead to myocardial infarction. Zhang and Hancox [12] improved Luo-Rudy ventricular action potential models by integrating I-Kr current and inactivation-deficient I-Kr into the previous model and verified that loss of inactivation of the I-Kr led to QT interval shortening. Adeniran et al. [13] further considered stretch-activated channel current (sac) in the single cell models and then incorporated the models into 3D human ventricular tissue models to explore the Short QT Syndrome (SQTS) which is associated with ventricular arrhythmias and sudden cardiac death. The symptom of ischemic greatly increases the probability of occurrence of ventricle fibrillation. It has important meaning to investigate the intricate mechanisms under an ischemic condition in order to better facilitate therapeutic interventions. Although a vast amount of experimental and clinical data of the ionic, cellular, and tissue substrates has been acquired, the precise cardiac mechanisms of ischemia are not well understood. Therefore, any advances in finding and tracking the pathophysiological feature, especially advances that might help analyze and treat the cardiac ischemia more effectively are of great significance. Trejos et al. [14] proposed a mechanism of automatic detecting ischemic events using ECG signals, which allows a better interpretation of cardiac ischemic behavior and results in an increase in the discrimination capability for ischemia detection. Cimponeriu et al. [15] developed a two-dimensional realistic ventricular tissue model. The capacity of the model in simulating pathological conditions was validated on exploring the determinants of electrocardiographic (ECG) morphology and tracking in the ECG pathologic changes of ischemic heart. The cardiac electrophysiological activity has been proven to be important in analyzing functional mechanisms under cardiac physiological and pathological condition. At present, researches have carried out the study on the modeling and simulation of cardiac ischemia based on the ventricular cell model [16–21]. Ten Tusscher and Panfilov [22] created a human ventricular cell model which contains all major ion channel currents and thus simulated the human cardiac electrophysiological properties in a closer way. Chinchapatnam et al. [23] used a fast electrophysiological (EP) model and proposed an adaptive algorithm to estimate cardiac local conduction velocity and apparent electrical conductivity. The method revealed hidden cardiac parameters and can help guide diagnosis and therapy of human left ventricle arrhythmia. A computational cardiac model was applied to simulate the electrophysiological action of two drugs of amiodarone and cisapride in healthy and ischemic ventricle cells for investigating the pharmacological effects, which is helpful to analyze the underlying arrhythmias mechanisms caused by the two drugs [24]. Lü et al. [25] developed a human ventricular cell and tissue ischemic model. Through the model, the functional consequences and mechanisms underlying the arrhythmias in early acute global ischemia are investigated to analyze the influence of acute global ischemia on cardiac electrical activity and subsequently on reentrant arrhythmogenesis. Lu et al. [26] further developed a 3D human ventricular ischemic model combining a detailed biophysical description of the excitation kinetics of human ventricular cells with an integrated geometry of human ventricular tissue. To analyze the spatiotemporal deformation parameters for the myocardial contraction, Han et al. [27] proposed the visualization tools and a strategy for the automatic detection of dysfunctional regions of cardiac ischemic pathologies, which is proved very useful for quantitatively demonstrating the main properties of the left ventricle myocardial contraction. Shenai et al. [28] presented the visualization of normal and ischemic propagation and found intra-QRS changes in and around the ischemic region, which proved that ischemia may cause depolarization changes detectable by both action potentials and unipolar leads. To exhibit the electrophysiological activities under the physiological and pathological condition within the authentic cardiac structure, Wang et al. presented a multivariate visualization method [29] and Zhang et al. proposed an interactive visualization algorithm [30] to visualize both the anatomical data and the electrophysiological data simultaneously. However, these methods cannot explore the hidden electrophysiological feature of pathological tissue in the 3D space.

In this paper, we proposed a visualization framework, which combines the human cardiac ischemic model with a novel depth weighted optic attenuation model, to inspect the occluded cardiac ischemia information with the complicated context of electrophysiological activities under cardiac ischemic condition. First the human ventricle ischemic data is acquired through the cardiac ischemic model. In the proposed depth weighted optic attenuation model, Euclidean Distance Transform (EDT) of each voxel is computed in the electrophysiological data, that is, the Euclidean distance from each voxel to the ventricle boundary, as the coefficient of the attenuation degree of the voxel. This model makes the voxel which is closer to the boundary of the ventricular tissue have the higher attenuation value. Thus, the region that contains the voxels is more transparent. The hidden feature of interest in the ischemic tissue can be revealed from complex overlapping electrophysiological information by the model. The paper is organized as follows. Section 2 presents the human cardiac tissue ischemic model and visualization framework which includes a novel depth weighted optic attenuation model construction. Section 3 provides experimental results and discussions. In Section 3, results of the experiments demonstrate that the method we presented can show the feature of cardiac action potential propagation during ischemia more effectively through surrounding complex information. Finally, our conclusions are given in Section 4.

2. Design Materials and Methods

To explore organs of interest from mass of cardiac tissues, Zhang et al. [31–33] proposed approaches for revealing detailed structures and further presented a cardiac visualization system, which can provide the user different levels of cardiac anatomy rendering [34]. Yang et al. [35] designed

a multidimensional transfer function for visualizing the multiboundary cardiac volume data. Different from the cardiac anatomy characteristic, electrophysiological activities such as excitation propagation in the various human heart tissues are hard to be observed and analyzed in the 3D space. To address this issue, Zhang et al. proposed a GPU-based high performance wave propagation simulation with fine anatomical structure [36]. Based on their work [11, 37], a GPU-based framework for electrophysiological data simulation and visualization is proposed. To fuse cardiac anatomical and electrophysiological model together, Yang et al. [38] designed the fusion transfer function which demonstrated cardiac electrophysiological activity by adjusting the parameter opacity of transfer function.

However, these methods cannot directly explore those cardiac function features at pathological conditions occluded by the complex biophysical information. In this section, we first induce a human cardiac ischemic model to explore cardiac electrophysiological activity and generate the altered ischemic electrophysiology data. Then 3D Euclidean distance transform is implemented on the data, and the depth weighted optic attenuation model is consequently constructed based on the Euclidean distance transform for revealing the hidden cardiac ischemic action potential propagation feature.

2.1. Cardiac Ischemic Electrophysiological Model. To explore the cardiac ischemic feature, in this work, the phase of ischemia is considered in the cardiomyocyte electrophysiological model, which describes the cardiac ischemic action potential (AP) generation through the monodomain reaction-diffusion equation as follows:

$$\begin{aligned}\frac{\partial V_m}{\partial t} &= -\frac{I_{\text{ion}} + I_{\text{stim}}}{C_m} + \nabla \cdot (D \nabla V_m),\\ I_{\text{ion}} &= I_{\text{Na}} + I_{K1} + I_{to} + I_{Kr} + I_{Ks} + I_{\text{CaL}} + I_{\text{NaCa}}\\ &\quad + I_{\text{NaK}} + I_{p\text{Ca}} + I_{pK} + I_{b\text{Ca}} + I_{b\text{Na}} + I_{K(\text{ATP})},\end{aligned} \tag{1}$$

where V_m represents transmembrane potential and t is the time. I_{ion} is the total ionic current depending on the voltage and time and I_{stim} indicates the externally applied stimulate current. C_m is the transmembrane capacitance per unit membrane area. D is the diffusion tensor for describing the tissue conductivity and ∇ is the gradient operator. The ionic current $I_{K(\text{ATP})}$ in I_{ion} is the ATP sensitive K^+ current which is calculated by the following equation [17]:

$$\begin{aligned}I_{K(\text{ATP})} &= (V_m - E_K)\left(\frac{[K^+]_o}{[K^+]_{o,\text{control}}}\right)^n f \rho_0 \frac{g_{\text{ATP}}}{A_m},\\ f &= f_{\text{ATP}} f_T f_M f_N,\end{aligned} \tag{2}$$

where E_K is the potassium ion equilibrium potential which is given by Nerst equation [18]:

$$E_K = \frac{RT}{F} \log\left(\frac{[K^+]_o}{[K^+]_i}\right), \tag{3}$$

where f_{ATP} is the fraction of opened channels and f_T is the temperature dependent factor. f_M and f_N are correction factors caused by intracellular Mg^{2+} ions and intracellular Na^+ ions. ρ_0 is the open probability of a channel in the absence of ATP. g_{ATP} is the gate control variable of adenosine triphosphate (ATP) and A_m represents the ratio of cell membrane surface area and volume.

f_{ATP} is a Hill equation:

$$f_{\text{ATP}} = \frac{1}{1 + \left([\text{ATP}]_i / K_m\right)^H}, \tag{4}$$

where K_m and H are the nonlinear function of $[\text{ADP}]_i$:

$$\begin{aligned}K_m &= 35.8 + 17.9\,[\text{ADP}]_i^{0.256},\\ H &= 1.3 + 0.74 \exp\left(-0.09\,[\text{ADP}]_i\right),\end{aligned} \tag{5}$$

f_T is described by the temperature effect formula:

$$f_T(T) = Q_{10}^{(T - T_0)/10}, \tag{6}$$

where Q_{10}, T, and T_0 represent the temperature coefficient, absolute temperature, and reference temperature, respectively, and $Q_{10} = 1.3$, $T_0 = 36^\circ\text{C}$. f_M is used to explain the inward rectification of intracellular magnesium ions, which is a Hill equation:

$$f_M = \frac{1}{1 + \left[Mg^{2+}\right]_i / K_{h,\text{Mg}}}. \tag{7}$$

Here $K_{h,\text{Mg}}$ is defined as follows:

$$K_{h,\text{Mg}} = K_{h,\text{Mg}}^0\left([K^+]_o\right) \exp\left(-\frac{2\delta_{\text{Mg}} F}{RT} V_m\right), \tag{8}$$

where $\delta_{\text{Mg}} = 0.32$ and $K_{h,\text{Mg}}^0([K^+]_o)$ is defined by

$$K_{h,\text{Mg}}^0\left([K^+]_o\right) = \frac{0.65}{\sqrt{[K^+]_o + 5}}, \tag{9}$$

where f_N is used to explain the inward rectifier ion induced cell Boehner, which is also a Hill equation:

$$f_N = \frac{1}{1 + \left([\text{Na}^+]_i / K_{h,\text{Na}}\right)^2}. \tag{10}$$

Here $K_{h,\text{Na}}$ is defined as follows:

$$K_{h,\text{Na}} = K_{h,\text{Na}}^0 \exp\left(-\frac{2\delta_{\text{Na}} F}{RT} V_m\right), \tag{11}$$

where $\delta_{\text{Na}} = 0.35$ and $K_{h,\text{Na}}^0 = 25.9\,\text{mM}$. The parameter setting in the ischemic model can be found in [18, 19].

The electrophysiological data is acquired by implementing the ischemic model on the Visible Human ventricle data. The value of each voxel in the electrophysiological volume data is the action potential of the cardiac cell under the ischemia condition. Thus, the electrophysiological volume data can represent the ventricle action potential propagation during ischemia.

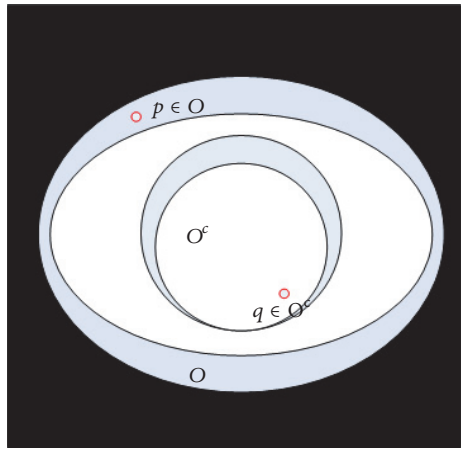


Figure 1: Elements defined in the field of Euclidean distance transform.

2.2. Euclidean Distance Transform. Distance transform (DT) maps each point into its smallest distance to regions of interest [39]. The central problem of EDT (Euclidean Distance Transform) is to compute the Euclidean distance of each point to a given subset of a plane. Let $I : \Omega \subset Z^2 \rightarrow \{0, 1\}$ be a binary image, $\Omega = \{0, \ldots, 1\} \times \{0, \ldots, 1\}$. By convention, 0 is assigned to black and 1 to white. Hence, we have the set O which is represented by all white pixels: $O = \{p \in \Omega \mid I(p) = 1\}$, as shown in Figure 1. The set O is called foreground and can consist of any subset in the image domain, including disjoint sets. The elements of its complement, O^c, the set of black pixels in Ω, are called background. From the DT point of view, the background pixels are called the interest points or feature points.

Definition 1. The distance transform (DT) is the transformation that generates a map D whose value of each pixel p is the smallest distance from this pixel to O^c:

$$\begin{aligned} D(p) &= \min \{d(p, q) \mid q \in O^c\} \\ &= \min \{d(p, q) \mid I(q) = 0\}. \end{aligned} \tag{12}$$

D is called the *distance map* of I. D itself can also be called a distance transform. Moreover, $d(p, q)$ is generally taken as the Euclidean distance:

$$d(p, q) = \sqrt{(p_i - q_i)^2 + (p_j - q_j)^2}. \tag{13}$$

To extend the 2D binary image I to 3D space, we let $I_{3D} : \Omega_{3D} \subset Z^3 \rightarrow \{0, 1\}$ be a set of 2D binary images, where $\Omega_{3D} = \{0, \ldots, 1\} \times \{0, \ldots, 1\} \times \{0, \ldots, 1\}$. 0 and 1 are the same as those in 2D binary image. O_{3D} and ${O_{3D}}^c$ are object set and the set of black pixels in Ω_{3D}, respectively. 3D distance map of each pixel p in I_{3D} is thus defined as

$$D_{3D}(p) \coloneqq \min \{d_{3D}(p, q) \mid q \in {O_{3D}}^c\}, \tag{14}$$

and 3D Euclidean distance $d_{3D}(p, q)$ is given by

$$d_{3D}(p, q) = \sqrt{(p_i - q_i)^2 + (p_j - q_j)^2 + (p_k - q_k)^2}. \tag{15}$$

2.3. Depth Weighted Optic Attenuation Model. The optic radiation function for visualizing the cardiac ischemic data acquired by the reaction-diffusion equation in Section 2.1 is [40]

$$C = \int_0^D C(t)\, \tau(t)\, e^{-\int_0^t \tau(s) ds} dt, \tag{16}$$

where $C(t)$ is the radiance and $\tau(t)$ is the attenuation degree function of a sample t in the cardiac volume data along the view direction.

To inspect the occluded cardiac ischemia information with the complicated context of electrophysiological activities, we consider the calculated 3D Euclidean distance transform of a sample x_i in the cardiac ischemic volume data as the attenuation factor. 3D Euclidean distance transform demonstrates the depth to the boundary of tissues that x_i belongs to. Then the improved depth attenuation degree function can then be acquired as follows:

$$\tau_{\text{depth}}(x_i) = \tau(x_i) \cdot \Theta_{\text{EDT}}(x_i), \tag{17}$$

where $\Theta_{\text{EDT}}(x_i)$ is associated with unit normalized 3D Euclidean distance transform result, which is thought to be the depth of x_i.

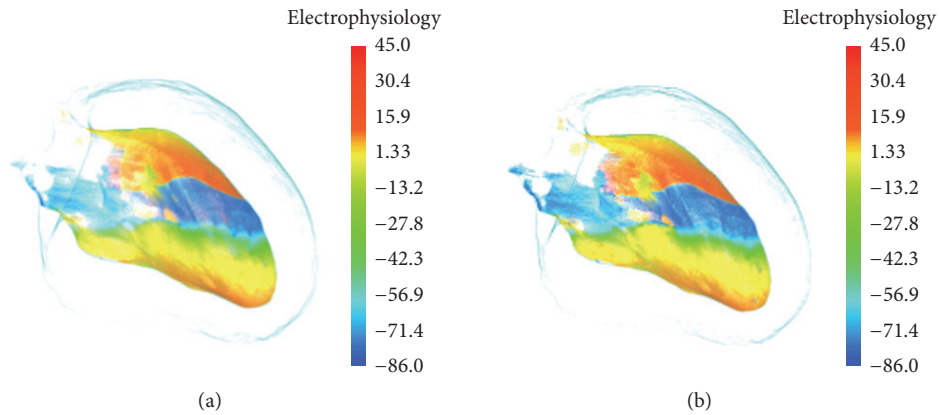
We incorporate the depth attenuation degree function into the optic radiation function and construct the depth weighted optic attenuation model as

$$C = \int_0^D C(t)\, \tau_{\text{depth}}(t)\, e^{-\int_0^t \tau_{\text{depth}}(s) ds} dt, \tag{18}$$

where attenuation $\tau(t)$ is replaced by $\tau_{\text{depth}}(x_i)$. Thus, the opacity of a volume sample will increase when it has a larger depth to the boundary, which means that a sample is more opaque when it is farther from the boundary. The hidden ischemia region of pathological tissue can then be revealed from complex overlapping information generated by the cardiac physiology model.

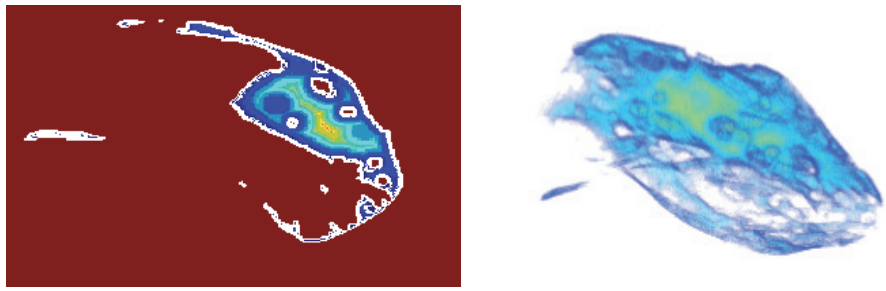
3. Experimental Results

In this section, the proposed depth weighted optic attenuation model was applied on the acquired electrophysiological ischemic data. Then the performance of the visualization method is assessed. The method exploited the visualization toolkit (VTK) libraries and the visualization system was developed under the environment of Visual Studio 2010. In this study, the electrophysiological ischemic model of cardiomyocytes is implemented to describe biophysical properties of the heart under pathological condition. In the simulation, the interior features of acquired cardiac ischemic electrophysiology data are impossible to be explored through traditional optic model. Figure 2 depicts the traditional electrophysiology visualization result of stimulated inner left ventricle muscles under the normal and ischemic condition using the normal optic radiation model. Conventional visualization of excitation propagation under the normal condition is shown in Figure 2(a). Figure 2(b) shows excitation propagation under the ischemic condition. Due to the



(a) (b)

Figure 2: Electrophysiology visualization of inner left ventricle: (a) under the normal condition and (b) under the ischemia condition.



(a) (b)

Figure 3: 3D Euclidean distance transformation in 3D space of inner left ventricle: (a) 2D projection slice and (b) 3D illustration.

mild ischemia and small ischemic region, the influence on excitation propagation is not able to spread to the surface of inner left ventricle muscles. Action potential propagation on the surface under the ischemic condition thus performs the same as the spiral wave shown under the condition of normal propagation. We can therefore hardly differentiate these two excitations from each other based on the patterns of wave propagation on the surface layer.

To implement the 3D exact Euclidean distance transform on the electrophysiology volume data, we associate those voxels on the boundary in volume to "black" pixels in distance transform terminology, and voxels inside the material are associated with the "white" pixels. In this way, the depth of inner voxels to boundary is represented by distance transformation of those voxels. Figure 3(a) shows the effect of 2D projection slice of distance transformation in 3D space of inner left ventricle muscles. The value in each pixel which represents the smallest distance from this pixel to black pixels is mapped onto color which changes from blue to red with increasing of the distance to the boundary. Through the 3D exact Euclidean distance transform, the depth of inner samples in cardiac ischemic data to boundary is represented by distance transformation of those samples. Figure 3(b) shows the effect of exact Euclidean distance transformation in 3D space of inner left ventricle muscles. The value of each pixel represents the smallest distance from this pixel to the "black" pixels and is mapped onto color, which changes from blue to red with increasing of the distance to the boundary.

Figure 4 shows the effect of revealing interior ischemia region at the different time with the proposed depth weighted optic attenuation model. Using traditional optic radiation model, electrophysiology visualization of inner left ventricle muscles at 720 ms under the ischemic condition is demonstrated in Figure 4(a). Since the excitation propagation on the surface layer is the same as the excitation at normal physiological condition, the feature of ischemic electrophysiology activity of left ventricle cannot be distinguished. In Figure 4(b), the electrophysiological information which is mapped onto color gradually fades to transparent with decreasing distance to boundary in the anatomical model. Interior ischemia region is then able to be highlighted from surrounding complex electrophysiological and anatomical context. The region marked in Figure 4(b) in the white ellipse is the myocardial blood clot which is revealed from the occlusion caused by surrounding biophysical activity. As seen in Figure 4(b), since the propagation velocity of the reentrant in the ischemia region slows down, a wavefront gap appears in the result image. At this time, with the propagation of the reentry wave, the conduction block is generated in the region, which will increase the transition probability of ventricular tachycardia to ventricular fibrillation. Figure 4(c) shows ischemic electrophysiological activity at 1040 ms using

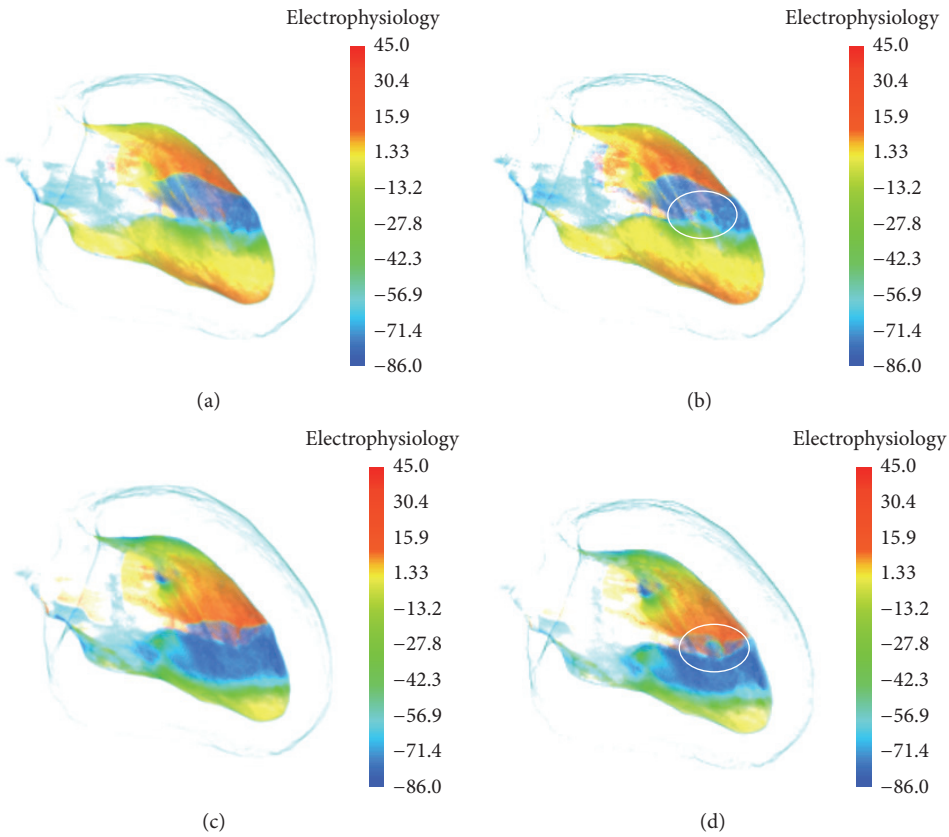


Figure 4: Electrophysiology visualization of inner left ventricle at different time under the ischemia condition: (a) by conventional optic radiation model at 720 ms; (b) by the depth weighted optic attenuation model at 720 ms; (c) by conventional optic radiation model at 1040 ms; (d) by the depth weighted optic attenuation model at 1040 ms.

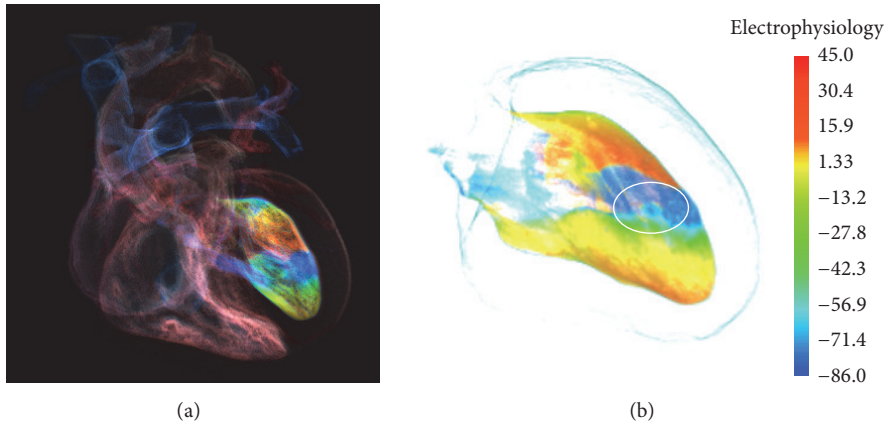


Figure 5: Electrophysiology visualization of inner left ventricle under the ischemia condition: (a) by the GPU-based multimodality simulation [11] and (b) by our proposed method.

traditional optic radiation model. In Figure 4(c), reentrant on the surface propagates forward stably, while the inner feature is occluded due to outer sophisticated overlapping biophysical information. Using the presented depth attenuation degree based model, as marked by the white ellipse in Figure 4(d), the hidden ischemia feature is distinctly explored. From Figure 4(d) we can see that the wave velocity in the ischemic area is slower than that in the surrounding normal tissue and the reentrant attempts to spread through the ischemic area. However, the reentry wave of the surrounding normal tissue continues to spread without conduction block.

Figure 5 shows the GPU-based multimodality simulation in [11] and the effect of exploring cardiac ischemic

electrophysiological activity by the method in this work. In Figure 5(a), inner left ventricle muscles under the ischemia condition are incorporated with cardiac anatomy model, while features in ischemic region are occluded by the action potential propagation in the outer tissue. In addition, the rendering result presented in [11] does not provide quantitative information about the excitation propagations. To improve this situation, as shown in Figure 5(b), through the method proposed in this study, the hidden ischemia region and the feature of excitation propagation in the region are revealed clearly in a quantitative way.

4. Conclusions

In this paper, we implemented a human cardiac ischemic model and revealed the hidden cardiac biophysical behavior under the ischemic condition by the depth attenuation degree based optic attenuation model. To explore the important features of interest of the heart under the pathological condition of ischemia, we first used a human cardiac ischemic model and acquired cardiac ischemic electrophysiology data. Then the depth of a sample to its boundary in the data is computed through the 3D Euclidean distance transform. We integrated the generated depth attenuation degree function based on the 3D Euclidean distance transform into the normal optic radiation model and then constructed the depth weighted optic radiation model. The experimental results showed that hiding features in ischemia region are effectively explored with complex electrophysiological context, which provides those medical staff and cardiac researchers with new information of the underlying cardiac biophysical mechanisms.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (NSFC) no. 61502275 and the Fundamental Research Funds for the Central Universities no. 2015ZQXM004. This work was also supported in part by the National Natural Science Foundation of China (NSFC) Grants nos. 61571165 and 61501450, Youth Foundation of Harbin University (no. HUYF2013-025), and the Higher Education and Teaching Reform Project (no. JG2014011155).

References

[1] K. H. W. J. Ten Tusscher, R. Hren, and A. V. Panfilov, "Organization of ventricular fibrillation in the human heart," *Circulation Research*, vol. 100, no. 12, pp. e87–e101, 2007.

[2] Z. P. Zhong, H. Wang, and X. T. Hou, "Extracorporeal membrane oxygenation as a bridge for heart failure and cardiogenic shock," *BioMed Research International*, vol. 2016, Article ID 7263187, 6 pages, 2016.

[3] V. Serpooshan, M. Zhao, S. A. Metzler et al., "Use of bio-mimetic three-dimensional technology in therapeutics for heart disease," *Bioengineered*, vol. 5, no. 3, pp. 193–197, 2014.

[4] M. H. Namazi, I. Khaheshi, H. Haybar, and S. Esmaeeli, "Cardiac failure as an unusual presentation in a patient with history of amyotrophic lateral sclerosis," *Case Reports in Neurological Medicine*, vol. 2014, Article ID 986139, 3 pages, 2014.

[5] E. Alickovic and A. Subasi, "Medical decision support system for diagnosis of heart arrhythmia using DWT and random forests classifier," *Journal of Medical Systems*, vol. 40, article 108, pp. 1–12, 2016.

[6] D. U. J. Keller, F. M. Weber, G. Seemann, and O. Dössel, "Ranking the influence of tissue conductivities on forward-calculated ecgs," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 7, pp. 1568–1576, 2010.

[7] P. Brocklehurst, I. Adeniran, D. Yang, Y. Sheng, H. Zhang, and J. Ye, "A 2D electromechanical model of human atrial tissue using the discrete element method," *BioMed Research International*, vol. 2015, Article ID 854953, 12 pages, 2015.

[8] J. L. Salinet Jr., G. N. Oliveira, F. J. Vanheusden, J. L. D. Comba, G. A. Ng, and F. S. Schlindwein, "Visualizing intracardiac atrial fibrillation electrograms using spectral analysis," *Computing in Science & Engineering*, vol. 15, no. 2, Article ID 6478759, pp. 79–87, 2013.

[9] O. V. Aslanidi, M. A. Colman, J. Stott et al., "3D virtual human atria: a computational platform for studying clinical atrial fibrillation," *Progress in Biophysics & Molecular Biology*, vol. 107, no. 1, pp. 156–168, 2011.

[10] V. Sala, S. Gatti, S. Gallo et al., "A new transgenic mouse model of heart failure and cardiac cachexia raised by sustained activation of met tyrosine kinase in the heart," *BioMed Research International*, vol. 2016, Article ID 9549036, 13 pages, 2016.

[11] L. Zhang, K. Q. Wang, W. M. Zuo, and C. Q. Gai, "G-Heart: a GPU-based system for electrophysiological simulation and multi-modality cardiac visualization," *Journal of Computers*, vol. 9, no. 2, pp. 360–367, 2014.

[12] H. Zhang and J. C. Hancox, "In silico study of action potential and QT interval shortening due to loss of inactivation of the cardiac rapid delayed rectifier potassium current," *Biochemical and Biophysical Research Communications*, vol. 322, no. 2, pp. 693–699, 2004.

[13] I. Adeniran, J. C. Hancox, and H. G. Zhang, "In silico investigation of the short QT syndrome, using human ventricle models incorporating electromechanical coupling," *Frontiers in Physiology*, vol. 4, article 166, pp. 1–16, 2013.

[14] E. D. Trejos, A. P. Lluna, M. V. Ferrer, P. C. Magrans, and G. C. Domínguez, "Dimensionality reduction oriented toward the feature visualization for ischemia detection," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 590–598, 2009.

[15] A. Cimponeriu, C. F. Starmer, and A. Bezerianos, "A theoretical analysis of acute ischemia and infarction using ECG reconstruction on a 2-D model of myocardium," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 1, pp. 41–54, 2001.

[16] O. V. Aslanidi, R. H. Clayton, J. L. Lambert, and A. V. Holden, "Dynamical and cellular electrophysiological mechanisms of ECG changes during ischaemia," *Journal of Theoretical Biology*, vol. 237, no. 4, pp. 369–381, 2005.

[17] J. N. Weiss, N. Venkatesh, and S. T. Lamp, "ATP-sensitive K+ channels and cellular K+ loss in hypoxic and ischaemic mammalian ventricle," *The Journal of Physiology*, vol. 447, pp. 649–673, 1992.

[18] J. M. Ferrero Jr., J. Sáiz, J. M. Ferrero, and N. V. Thakor, "Simulation of action potentials from metabolically impaired cardiac myocytes: role of ATP-sensitive K+ current," *Circulation Research*, vol. 79, no. 2, pp. 208–221, 1996.

[19] R. M. Shaw and Y. Rudy, "Electrophysiologic effects of acute myocardial ischemia: a theoretical study of altered cell excitability and action potential duration," *Cardiovascular Research*, vol. 35, no. 2, pp. 256–272, 1997.

[20] K. Gima and Y. Rudy, "Ionic current basis of electrocardiographic waveforms: a model study," *Circulation Research*, vol. 90, no. 8, pp. 889–896, 2002.

[21] B. Rodríguez, N. Trayanova, and D. Noble, "Modeling cardiac ischemia," *Annals of the New York Academy of Sciences*, vol. 1080, pp. 395–414, 2006.

[22] K. H. W. J. Ten Tusscher and A. V. Panfilov, "Alternans and spiral breakup in a human ventricular tissue model," *American Journal of Physiology—Heart and Circulatory Physiology*, vol. 291, no. 3, pp. H1088–H1100, 2006.

[23] P. Chinchapatnam, K. S. Rhode, M. Ginks et al., "Model-based imaging of cardiac apparent conductivity and local conduction velocity for diagnosis and planning of therapy," *IEEE Transactions on Medical Imaging*, vol. 27, no. 11, pp. 1631–1642, 2008.

[24] M. Wilhelms, C. Rombach, E. P. Scholz, O. Dössel, and G. Seemann, "Impact of amiodarone and cisapride on simulated human ventricular electrophysiology and electrocardiograms," *Europace*, vol. 14, no. 5, pp. V90–V96, 2012.

[25] W.-G. Lü, J. Li, F. Yang, and K.-Q. Wang, "Simulation study of ventricular arrhythmia at the early stage of global ischemic condition," *Progress in Biochemistry and Biophysics*, vol. 42, no. 2, pp. 189–194, 2015.

[26] W. Lu, J. Li, F. Yang et al., "Effects of acute global ischemia on re-entrant arrhythmogenesis: A Simulation Study," *Journal of Biological Systems*, vol. 23, no. 2, pp. 213–230, 2015.

[27] M. Han, P. Clarysse, P. Croisille, I. E. Magnin, and D. Revel, "Computer aided diagnosis of the myocardial ischemia based on a spatio-temporal deformation features analysis," *Computers in Cardiology*, vol. 25, pp. 749–752, 1998.

[28] M. Shenai, B. Gramatikov, and N. V. Thakor, "Computer modeling of depolarization changes induced by myocardial ischemia," *Computers in Cardiology*, vol. 25, pp. 321–324, 1998.

[29] K. Q. Wang, F. Yang, W. M. Zuo, N. Ding, and H. G. Zhang, "Effective transfer function for interactive visualization and multivariate volume data," in *Proceedings of the 4th International Conference on Biomedical Engineering and Informatics (BMEI '11)*, pp. 272–276, IEEE, Shanghai, China, October 2011.

[30] L. Zhang, K. Wang, W. Zuo, Y. Wu, and D. Han, "GPU-based fusion method for 3D electrophysiological data visualization," in *Proceedings of the International Conference on Computerized Healthcare (ICCH '12)*, pp. 51–56, December 2012.

[31] K. Q. Wang, L. Zhang, C. Q. Gai, and W. M. Zuo, "Illustrative visualization of segmented human cardiac anatomy based on context-preserving model," *Computing in Cardiology*, pp. 485–488, 2011.

[32] L. Zhang, C. Gai, K. Wang, and W. Zuo, "Real-time interactive heart illustration platform via hardware accelerated rendering," in *Proceedings of the 3rd IEEE International Conference on Advanced Computer Control (ICACC '11)*, pp. 497–501, January 2011.

[33] L. Zhang, K. Wang, H. Zhang, W. Zuo, X. Liang, and J. Shi, "Illustrative cardiac visualization via perception-based lighting enhancement," *Journal of Medical Imaging and Health Informatics*, vol. 4, no. 2, pp. 312–316, 2014.

[34] L. Zhang, K. Q. Wang, F. Yang et al., "A visualization system for interactive exploration of the cardiac anatomy," *Journal of Medical Systems*, vol. 40, no. 135, pp. 1–12, 2016.

[35] F. Yang, L. Zhang, W. G. Lu et al., "Multi-boundary cardiac data visualization based on multidimensional transfer function with ray distance," *Bio-Medical Materials and Engineering*, vol. 24, no. 6, pp. 3025–3032, 2014.

[36] L. Zhang, C. Gai, K. Wang, W. Lu, and W. Zuo, "GPU-based high performance wave propagation simulation of ischemia in anatomically detailed ventricle," in *Proceedings of the Computing in Cardiology Conference (CinC '11)*, pp. 469–472, Hangzhou, China, September 2011.

[37] L. Zhang, K. Q. Wang, W. M. Zuo, and M. Z. Yang, "Real-time multi-volume rendering for 3D electrophysiological data visualization based on graphics processing unit," *ICIC Express Letters, Part B: Applications*, vol. 4, no. 6, pp. 1625–1630, 2013.

[38] F. Yang, W. G. Lu, L. Zhang, W. M. Zuo, K. Q. Wang, and H. G. Zhang, "Fusion visualization for cardiac anatomical and ischemic models with depth weighted optic radiation function," in *Proceedings of the Computing in Cardiology Conference (CinC '15)*, pp. 937–940, IEEE, Nice, France, September 2015.

[39] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *Journal of the Association for Computing Machinery*, vol. 13, no. 4, pp. 471–494, 1966.

[40] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 7861274, 10 pages
http://dx.doi.org/10.1155/2016/7861274

Research Article

Analysis of Important Gene Ontology Terms and Biological Pathways Related to Pancreatic Cancer

Hang Yin,[1] ShaoPeng Wang,[2] Yu-Hang Zhang,[3] Yu-Dong Cai,[2] and Hailin Liu[1]

[1]*Department of Gastroenterology, Ninth People's Hospital, School of Medicine, Shanghai Jiao Tong University, Shanghai 200011, China*
[2]*School of Life Sciences, Shanghai University, Shanghai 200444, China*
[3]*Institute of Health Sciences, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China*

Correspondence should be addressed to Hailin Liu; liuhailin@medmail.com.cn

Received 31 May 2016; Revised 18 July 2016; Accepted 7 September 2016

Academic Editor: Yungang Xu

Pancreatic cancer is a serious disease that results in more than thirty thousand deaths around the world per year. To design effective treatments, many investigators have devoted themselves to the study of biological processes and mechanisms underlying this disease. However, it is far from complete. In this study, we tried to extract important gene ontology (GO) terms and KEGG pathways for pancreatic cancer by adopting some existing computational methods. Genes that have been validated to be related to pancreatic cancer and have not been validated were represented by features derived from GO terms and KEGG pathways using the enrichment theory. A popular feature selection method, minimum redundancy maximum relevance, was employed to analyze these features and extract important GO terms and KEGG pathways. An extensive analysis of the obtained GO terms and KEGG pathways was provided to confirm the correlations between them and pancreatic cancer.

1. Introduction

Pancreatic cancer has been widely reported as a malignant tumor subtype involving one of the most significant tissue organs that contribute to both digestive system and endocrine system, the pancreas. Based on clinical symptoms and genetic characteristics, pancreatic cancer can be clustered into various subtypes [1]. Among such subtypes, pancreatic ductal adenocarcinoma (PDAC) accounts for more than 90% of all the cases. With a specific low survival rate (18% for one-year survival rate and 5% for five-year survival rate), pancreatic cancer results in more than thirty thousand deaths around the world and has been regarded as one of the top killers for human beings [1, 2].

Although pancreatic cancer has been included in the list of top killers for human beings, the biological processes and mechanisms that contribute to the initiation and progression of pancreatic cancer have not been fully revealed. Based on recent publications, the underlying mechanisms of pancreatic cancer have been partially uncovered mainly by experimental trials [3, 4]. The traditional experimental trials that contribute to revealing of pancreatic cancer associated genes and pathways can be divided into two levels: the nucleotide level (DNA/RNA) and the protein level. At the nucleotide level, polymerase chain reaction (PCR), high-throughput sequencing, and gene chips (either genomic chips or expression profile chips) contribute to the identification of the genomic and transcriptional background for pancreatic cancer initiation and progression [5]. Taking gene chip as an example, such experimental tool reveals the detailed genetic and expression profile characteristics of tumor cells and has been reported to contribute to the identification of various pancreatic cancer associated biological processes, including DPC4 tumor-suppressor pathway and the famous MAPK signaling pathway which we will analyze below [6–8]. As for the protein level, western blot turns out to be the most commonly used biochemical method to identify the expression and activation status of a known protein in certain in vivo or in vitro environment. Further, relying on in vitro gene expression (RNA) interference technologies, the characteristic alteration of the expression and function of a series of proteins that have been identified on such two levels as we have mentioned above can be validated and such group of proteins can be further concluded into various biological

processes and pathways [9, 10]. Based on the experimental technologies we have mentioned above, various principal regulatory pathways have been identified and confirmed to contribute to the initiation and progression of pancreatic cancer.

Based on existing publications, various principle regulatory pathways and biological processes that contribute to the initiation and progression of pancreatic cancer have been identified. Such signaling pathways and biological processes contribute to three main aspects of the biological processes of pancreatic cancer: transmembrane signal transduction, intracellular metabolic transduction, and the intranuclear proliferative regulation [3, 11, 12]. Different signaling pathways have been identified to contribute to different biological processes of pancreatic cancer during tumorigenesis. According to recent literatures, ErbB signaling pathway and TGF-beta signaling pathway participate in the transmembrane signal transduction of pancreatic cancer [13, 14]. Such transmembrane signal transduction pathways have been further validated to transfer the signals to intracellular pathways (such as p53 signaling pathway, MAPK signaling pathway, PI3K-Akt signaling pathway, and VEGF signaling pathway) [13, 15–17]. Intracellular signaling pathways have been identified to contribute to the abnormal proliferation of pancreatic cells and further initiate the tumorigenesis. Taking MAPK signaling pathway as an example, as the downstream region of Ras signaling pathway, MAPK signaling pathway contributes to the phosphorylation of two crucial families of proteins, ERK and JNK, and further regulates proliferative signaling transportation into the nucleus [8]. Although various functional pathways have been revealed to contribute to the abnormal proliferation during the tumorigenesis of pancreatic cancer, the core trigger for the initiation of pancreatic cancer turns out to be the abnormal intranuclear proliferative regulation [18]. It has been identified that two main biological processes contribute to the abnormal proliferation of pancreatic cells during tumorigenesis: the inhibition of cell apoptosis and the excessive activation of proliferation [19]. All of such regulatory signaling pathways have been reported to be abnormal in pancreatic cancer and further contribute to the tumorigenesis. However, according to such signaling pathways, we still cannot explain all the pathological phenotypes of pancreatic cancer, implying that there are still core regulatory pathways remaining to be uncovered.

The study on the underlying mechanism of pancreatic cancer has lasted for decades [20]. However, based on experimental methods, only limited genes and pathways are proved to contribute to pancreatic cancer. The experimental methods that contribute to the identification and confirmation of pancreatic cancer associated pathways are quite expensive and time-consuming. Recently, with the development of computational biology and bioinformatics, various computational methods have been presented to predict cancer, including pancreatic cancer associated genes [21]. However, up to now, few computational methods have been present to describe the detailed functional pathways and biological processes of pancreatic cancer. In computational biology, KEGG pathways and gene ontology (GO) terms are widely used to describe the detailed and specific biological processes in human cells. KEGG (Kyoto Encyclopedia of Genes and Genomes) has been widely regarded as an integrated database resource for gene and protein annotation [22]. Based on KEGG database, we can obtain the KEGG pathway maps which reflect the functional pathway based network in living cells [22]. On the other hand, GO is a bioinformatics initiative to unify the presentation of gene and gene product attributes across all species [23]. Therefore, KEGG pathways and GO terms can provide a more accurate and clearer panorama for the underlying biological processes of pancreatic cancer.

In this study, we applied a popular feature selection method, minimum redundancy maximum relevance (mRMR) [24], to extract a group of pancreatic cancer associated KEGG pathways and GO terms, filling the gaps of current study in pancreatic cancer. First, genes that have been validated to be related to pancreatic cancer were deemed positive samples, while other genes were deemed negative samples. Second, the enrichment theory of GO term and KEGG pathway was adopted to encode each gene. Third, all GO terms and pathways were analyzed by mRMR method and some of the important ones were extracted. Finally, the extracted GO terms and KEGG pathways were extensively discussed to confirm their relationships to pancreatic cancer.

2. Materials and Methods

2.1. Materials. The validated genes related to pancreatic cancer were retrieved from the KEGG pathway, which is a main database in KEGG database [25]. 65 validated genes were extracted from the pathway hsa05212 (http://www.genome.jp/kegg-bin/show_pathway?map=hsa05212&show_description=show, accessed in December 2014). These genes were termed as positive samples, comprising the gene set S_p, and are listed in Supplementary Material I available online at http://dx.doi.org/10.1155/2016/7861274. To extract the GO terms and KEGG pathways that are specific to pancreatic cancer, it is necessary to employ some genes that are not related to pancreatic cancer. Since we used the enrichment scores of GO terms and KEGG pathways to indicate the associations between genes and GO terms (KEGG pathways), genes without these scores were not considered in this study. Up to now, there are 18,600 genes whose GO and KEGG enrichment scores can be calculated. Beside the 65 genes related to pancreatic cancer, each of the remaining 18,535 genes can be deemed a negative sample because the probability of it being related to pancreatic cancer is not very high. These 18,535 genes comprised the gene set S_n. The whole gene set S was constructed by combining the genes in S_p and S_n; that is, $S = S_p \cup S_n$.

2.2. Feature Construction. To extract important GO terms and KEGG pathways that are related to pancreatic cancer, it is necessary to encode each gene in S based on all GO terms and KEGG pathways. Here, we used the enrichment theory of GO term and KEGG pathway to encode each gene, which can indicate the relationships between genes and GO terms (KEGG pathways). Then, the difference between positive and

negative samples can be distinguished by the key features produced by a feature selection method, which would be described in Section 2.3. The encoding procedure is shown as follows.

GO Enrichment Score. The GO enrichment score was utilized to represent the quantitative correlation between each GO term and involved genes. For a given GO term GO_j and a gene g, let G_1 be a gene set consisting of genes annotated to GO_j and G_2 be another gene set consisting of neighbor genes of g in the protein-protein interaction network reported in STRING (http://string-db.org/) [26], a well-known public database providing known and predicted protein-protein interactions. The GO enrichment score between GO_j and g is defined as the $-\log_{10}$ of the hypergeometric test P value [27–30] of G_1 and G_2, which can be calculated by

$$\mathrm{ES}_{\mathrm{GO}}\left(g, \mathrm{GO}_j\right) = -\log_{10}\left(\sum_{k=m}^{n} \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}\right), \quad (1)$$

where N is the number of genes in human, M is the number of genes in G_1, n is the number of genes in G_2, and m is the number of common genes of G_1 and G_2. A large enrichment score between GO_j and g indicates close relationship between them. In this study, we considered 12,511 GO terms, inducing 12,511 GO enrichment scores for each gene, which can be obtained by an in-house program using R function phyper. The R code is "score ← −log10(phyper(numWdrawn − 1, numW, numB, numDrawn, lower.tail=FALSE))," where numW, numB, and numDrawn are the number of genes annotated to GO_j, the number of genes not annotated to GO_j, and the number of neighbors of gene g and numWdrawn is the number of neighbors of gene g that are also annotated to GO_j.

KEGG Enrichment Score. Similar to that of the GO terms, the relationship between KEGG pathways and genes in S can be represented by the KEGG enrichment scores. For a given KEGG pathway K_j and a gene g, let G_1 be a gene set consisting of genes in K_j and G_2 is same as G_2 in the above paragraph. The KEGG enrichment score between K_j and g is also defined to be the $-\log_{10}$ of the hypergeometric test P value [29–31] of G_1 and G_2, which can be computed by

$$\mathrm{ES}_{\mathrm{KEGG}}\left(g, K_j\right) = -\log_{10}\left(\sum_{k=m}^{n} \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}\right), \quad (2)$$

where the definitions of N, M, n, and m are same as those in (1). Also, a high score between a KEGG pathway K_j and a gene g indicates they have strong associations. Here, we considered 239 KEGG pathways, resulting in 239 KEGG enrichment scores for each gene, which can also be obtained by an in-house program using R function phyper.

As mentioned above, each gene g was represented by 12,511 features derived from GO terms and 239 features derived from KEGG pathways, which can be formulated as a vector

$$f(g) = \left(\mathrm{ES}_{\mathrm{GO}}\left(g, \mathrm{GO}_1\right), \ldots, \mathrm{ES}_{\mathrm{GO}}\left(g, \mathrm{GO}_{12511}\right), \mathrm{ES}_{\mathrm{KEGG}}\left(g, K_1\right), \ldots, \mathrm{ES}_{\mathrm{KEGG}}\left(g, K_{239}\right)\right)^T. \quad (3)$$

2.3. Feature Selection Method. As described in Section 2.2, each gene was represented by 12,750 features derived from the GO terms and KEGG pathways. Considering the unequal roles of these features on pancreatic cancer, that is, some features playing more important roles than others, it is necessary to employ some advanced tools to analyze them, thereby extracting key features that are strongly associated with pancreatic cancer. Here, a reliable and widely used feature selection method, namely, mRMR method [24], was adopted to analyze all investigated 12,750 features. The mRMR method was proposed by Peng et al. [24] and was deemed to be a useful tool to analyze the feature space of complicated problems. Up to now, it has been widely applied to analyze various complicated biological systems or problems [32–45].

The mRMR method has two excellent criteria: Max-Relevance and Min-Redundancy. The criterion of Max-Relevance measures the importance of features based on their correlation to targets, while the criterion of Min-Redundancy gives a guarantee that the selected features have minimum redundancies. It is clear that the former criterion can be used to extract important features for a classification problem, while if one tries to construct an optimal feature subspace, two of them should be used. Because the purpose of this study is to extract key features that are closely related to the pancreatic cancer but not to construct an optimal feature subspace, we only used the Max-Relevance in this study. For each feature, let f be a variable representing values of all samples under the feature and c be the target variable. The mutual information (MI) value of each feature can be computed by

$$I(c, f) = \iint p(c, f) \log \frac{p(c, f)}{p(c)\, p(f)} dc\, df, \quad (4)$$

where $p(c)$ and $p(f)$ are the marginal probabilities of c and f; $p(c, f)$ is the joint probabilistic distribution of c and f. In fact, MI measures the mutual dependence between two variables. Furthermore, it has wide applications because it can deal with random variables that are not real-valued. Thus, mRMR method adopted MI to measure the relevance of each feature. According to the MI values of all features, a feature list, namely, MaxRel feature list, can be built. The MaxRel feature list was formulated as

$$F_M = \left[f_1^M, f_2^M, \ldots, f_N^M\right], \quad (5)$$

where N represented the total number of features. Clearly, features with high ranks in this list are more likely to be related to pancreatic cancer. Extensive investigation of the corresponding GO terms and KEGG pathways may give new insights for the study of pancreatic cancer.

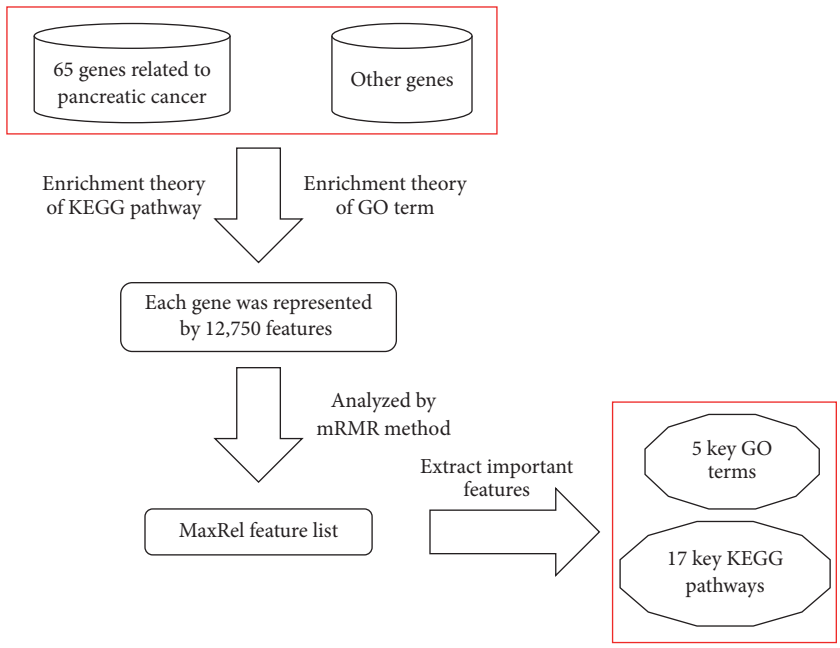


Figure 1: The procedures for extracting important KEGG pathways and GO terms of pancreatic cancer.

3. Results and Discussion

The purpose of this study is to extract important KEGG pathways and GO terms of pancreatic cancer using some computational methods. The detailed procedures are illustrated in Figure 1.

3.1. Results. As described in Section 2.2, each gene in S was represented by 12,750 features derived from the GO terms and KEGG pathways. These features were analyzed by mRMR method described in Section 2.3 by calculating their relevance to targets measured by their MI values. According to the MI value of each feature, the MaxRel feature list was constructed, which is provided in Supplementary Material II.

It is known that not all GO terms and KEGG pathways have strong associations with pancreatic cancer. The rank of a corresponding feature in the MaxRel feature list for a GO term or a KEGG pathway indicates its importance for pancreatic cancer. Thus, we can select the GO terms and KEGG pathways whose features received high ranks in the MaxRel feature list to investigate their importance. Here, we chose the 22 features receiving MI values no less than 0.01 for further analysis, resulting in 22 GO terms or KEGG pathways. Their detailed information is listed in Tables 1 and 2. It can be observed from these two tables that there are 17 important KEGG pathways (listed in Table 1) and five key GO terms (listed in Table 2). In the following section, detailed discussion on these GO terms and KEGG pathways would be given.

3.2. Analysis of Key KEGG Pathways and GO Terms. As shown in Tables 1 and 2, 17 KEGG pathways and five GO terms were extracted, which are deemed to be highly related to pancreatic cancer. According to recent published literature, all of these KEGG pathways and GO terms identified in this study have been confirmed to participate in pancreatic cancer associated biological processes.

3.2.1. KEGG Pathways Associated with Pancreatic Cancer. 17 KEGG pathways were extracted in this study, which are deemed to be associated with the initiation and progression of pancreatic cancer.

(1) Pathways Describe Various Tumor Subtypes. Among the 17 KEGG pathways, 10 KEGG pathways describe the whole metabolic regulatory network of a specific cancer subtype. KEGG pathway *hsa05200* describes the kernel regulatory factors that contribute to the initiation and progression of pan-cancer. Various pathways (e.g., Wnt signaling pathway, cAMP signaling pathway, and VEGF signaling pathway) in such network (hsa05200) and functional genes (e.g., *PKA*, *Rho*, and *VEGF*) have been identified in pancreatic cancer [46–48]. Taking gene *PKA* and its corresponding signaling pathway, the cAMP signaling pathway, as an example, cyclic AMP associated pathway and *PKA* have been identified and confirmed to contribute to the migration and invasion of pancreatic cancer, validating our prediction [48].

Apart from the KEGG pathways describing the pan-cancer, various KEGG pathways have also been predicted to describe the detailed subtypes of cancer. Among them, *hsa05223* which describes the regulatory network and pathways of non-small-cell lung cancer has been predicted to be related to the specific biological processes of pancreatic cancer. Such KEGG pathway contains various tumor associated factors and pathways (such as *KRAS*, *TP53*, and functional pathways that they participate in). It has been proved that *KRAS* and *TP53* as we have mentioned above have both been reported and confirmed to contribute to the initiation and

Table 1: 17 important KEGG pathways for pancreatic cancer.

KEGG pathway ID	KEGG pathway	MI value	Rank in MaxRel feature list
hsa05211	Renal cell carcinoma	0.011	1
hsa04010	MAPK signaling pathway	0.011	3
hsa05212	Pancreatic cancer	0.011	4
hsa05200	Pathways in cancer	0.011	5
hsa05210	Colorectal cancer	0.011	6
hsa05214	Glioma	0.011	7
hsa05220	Chronic myeloid leukemia	0.011	8
hsa05223	Non-small-cell lung cancer	0.01	9
hsa04510	Focal adhesion	0.01	10
hsa05213	Endometrial cancer	0.01	11
hsa05221	Acute myeloid leukemia	0.01	12
hsa05215	Prostate cancer	0.01	13
hsa05160	Hepatitis C	0.01	14
hsa04012	ErbB signaling pathway	0.01	16
hsa04660	T cell receptor signaling pathway	0.01	18
hsa04150	mTOR signaling pathway	0.01	20
hsa04722	Neurotrophin signaling pathway	0.01	22

Table 2: Five important GO terms for pancreatic cancer.

GO term ID	GO term	MI value	Rank in MaxRel feature list
GO: 0007265	Ras protein signal transduction	0.011	2
GO: 0048011	Neurotrophin-TRK receptor signaling pathway	0.01	15
GO: 0016772	Transferase activity, transferring phosphorus-containing groups	0.01	17
GO: 0016303	1-Phosphatidylinositol-3-kinase activity	0.01	19
GO: 0004713	Protein tyrosine kinase activity	0.01	21

progression of pancreatic cancer [49, 50]. Considering factors like *KRAS* and *TP53* which have been identified in both pancreatic cancer associated pathways and non-small-cell lung cancer associated pathways, such two regulatory networks (pancreatic cancer and non-small-cell lung cancer associated pathways) may definitely interact with each other, and our predicted KEGG term hsa05223 may actually participate in pancreatic associated pathways validating the accuracy and efficacy of our prediction. Apart from non-small-cell lung cancer, another four subtypes of cancer (prostate cancer, endometrial cancer, renal cell carcinoma, and colorectal cancer) associated biological processes have also been predicted to be associated with pancreatic cancer. There are various core regulatory factors and pathways in prostate cancer associated pathways (*hsa05215*). Steroid hormone biosynthesis has been reported to contribute to the metastasis of prostate cancer and interacts with the specific oncogene of pancreatic cancer *AR*, implying its core role for the prostate associated pathways that we have predicted [51, 52]. According to recent publications, such core pathway of the steroid hormone biosynthesis may also contribute to pancreatic cancer, revealing the underlying relationships between prostate cancer associated pathways (as we have predicted, hsa05215) and pancreatic cancer [53, 54].

Pathways of endometrial cancer (*hsa05213*), a malignant neoplasm involving the female genital system, have also been predicted to be associated with pancreatic cancer. The core regulatory factors of endometrial cancer and pancreatic cancer have quite a lot of crosstalk and overlap. Take β-catenin as an example, β-catenin has been revealed to participate in the Wnt signaling pathway in various tumor subtypes [55, 56]. The initiation and progression of both pancreatic cancer and endometrial cancer have been confirmed to be associated with Wnt signaling pathway, implying that such two regulatory networks may have crosstalk and our extracted KEGG pathway (hsa05213) may actually contribute to the progression of pancreatic cancer [57–59]. As for the other three important solid tumor associated pathways, two of them, renal cell carcinoma and colorectal cancer associated pathways (*hsa05211* and *hsa05210*), contain functional regulatory genes and pathways that have been reported to contribute to pancreatic cancer at the same time. In renal cell carcinoma, *MET* has been reported to interact with the hepatocyte growth factor (HGF) and turns out be the initial signal for *MAPK* signaling pathway [60]. Coincidentally, in pancreatic cancer, *MET* has also been confirmed to be quite a crucial gene for tumor initiation, progression, and metastasis, implying the crosstalk of pathways associated pancreatic cancer and renal cell carcinoma [61, 62]. As for colorectal cancer associated pathways (hsa05211), during the tumorigenesis of colorectal cancer, chromosomal instability (CIN) has been revealed to be a core driver mechanism and pathogenesis of the initiation [63]. In pancreatic cancer, CIN has also

been regarded as a common phenotype and pathogenic factor, implying the undergoing relationship between such two regulatory networks [64]. According to Table 1, we also obtained the specific KEGG pathway describing pancreatic cancer associated pathway (*hsa05212*) which is definitely associated with pancreatic cancer, validating the accuracy and efficacy of our prediction.

Apart from such solid tumor subtype, two leukemia subtypes and sarcoma associated pathways have also been obtained to contribute to the tumorigenesis of pancreatic cancer. KEGG pathway, *hsa05220*, describes pathogenic biological processes of chronic myeloid leukemia (CML). Various factors have been reported to contribute to the chronic myeloid leukemia. PI3K-AKT pathway has been identified as a core component of chronic myeloid leukemia associated pathways as we have predicted [65]. Based on recent publications, such pathway (PI3K-AKT pathway) has also been confirmed to be quite crucial for pancreatic cancer, validating our prediction [66]. Apart from that, the specific *BCR-ABL* fusion gene has also been identified in some pancreatic cancer patients, implying that *BCR-ABL* fusion gene may also contribute to the tumorigenesis of pancreatic cancer [67]. Apart from CML associated pathways, regulatory networks that contribute to another nonsolid tumor subtype, acute myeloid leukemia (AML) (*hsa05221*), have also been contained in our results. As we all know, *KRAS*, *STAT,* and their respective regulatory pathways have all been associated with our predicted pathway (hsa05221) [68, 69]. As we have mentioned above, *KRAS* has been identified as a core regulatory factor that contributes to pancreatic cancer [49]. According to recent publications, *STAT* as AML associated gene has also been reported to contribute to pancreatic cancer, validating our prediction of pancreatic cancer associated genes [70]. Glioma, rising from glial cells, is a malignant sarcoma involving the brain and central nervous system. Based on our results, glioma associated pathways (*hsa05214*) may also contribute to pancreatic cancer. Genes associated with glioma such as *EGFR* (as oncogene) and *PTEN* (as tumor suppressor) have also been reported to contribute to the initiation and progression of pancreatic cancer [71, 72].

(2) Detailed Pathways That May Participate in Tumorigenesis. Apart from pathways that directly describe the tumorigenesis, four KEGG pathways that describe the detailed pathways were also extracted. KEGG pathway, *hsa04150,* which describes the mTOR signaling pathway has been predicted to contribute to pancreatic cancer. The relationship between mTOR signaling pathway and pancreatic cancer has been revealed by multiple recent publications [73–75]. As a regulatory mechanism for cell proliferation, mTOR signaling pathway has been confirmed to have crosstalk with various core regulatory factors and their respective signaling pathways including *MAPK*, *TP53*, RAS, and *EGFR* [76–79]. Some of such regulatory factors have also been contained in our results. *MAPK* signaling pathway (*hsa04010*) has been confirmed to have crosstalk with mTOR signaling pathway as we have mentioned above and has been reported to be quite crucial for the invasion and metastasis of pancreatic cancer [80, 81]. Apart from such two functional signaling pathways, another pathway, which has been widely reported to contribute to endometrial cancer, *ERBB* signaling pathway (*hsa04012*), is also in Table 1 [82]. During the initiation and progression of pancreatic cancer, *ERBB* signaling pathway has been confirmed to participate in the biological processes, validating our newly presented algorithm [83]. Neurotrophins have firstly been identified as a group of proteins that contribute to the survival, development, and function of neurons [84]. However, recent publications have revealed that neurotrophins may participate in the survival and proliferation of various cell types including the tumor cells [85–87]. Such functional protein, neurotrophin, which is regulated by another functional gene *TRK* has also been reported to contribute to pancreatic cancer, validating the efficacy of our prediction [88]. In Table 1, a specific KEGG pathway, *hsa04722*, which describes the neurotrophin signaling pathway was also listed. Based on our analyses, such biological process may definitely contribute to pancreatic cancer.

(3) Specific Pathways That Contribute to Cell-Cell Interaction. The last three KEGG pathways with MI values no less than 0.01 have been confirmed to contribute to the cell-cell/cell-protein interaction associated pathways. KEGG pathway *hsa04510* describes the focal adhesion associated pathways. The abnormal activation of focal adhesion associated pathway has been widely reported in pancreatic cancer, implying that focal adhesion may be core biological processes during the tumorigenesis of pancreatic cancer [89]. Apart from focal adhesion, another biological process involving cell-cell interaction, T cell receptor signaling pathway (*hsa04660*), was also extracted in this study. It has been widely reported that the T cell receptor signaling pathway has been blocked or abnormally regulated in tumor microenvironment [90]. In pancreatic cancer, the initiation and progression of pancreatic cancer also interfere with the normal function of T cell receptor. Considering the recognition and cytolysis functions of T cells, the tumor cells and the T cells may put the selective pressure on each other and coevolve [91]. During the evolutionary processes, T cells with high recognition and cytolysis ability, which are both induced by T cell receptor signaling pathway, are all screened out, leaving dysfunctional T cells in tumor microenvironment [92, 93]. Such coevolution processes imply the regulatory role of T cell receptor signaling pathway in pancreatic cancer. We also obtained a functional signaling pathway (*hsa05160*) that is related with the infection of hepatitis C virus. Based on meta-analysis and case-control study, the infection of specific virus (hepatitis B and hepatitis C) has been confirmed to increase the risk of pancreatic cancer, validating the prediction based on our new algorithm, though the undergoing mechanism has not been fully revealed [94, 95].

3.2.2. GO Terms Associated with Pancreatic Cancer. Apart from the KEGG pathways mentioned above, five GO terms (listed in Table 2) that describe different biological processes were also extracted in this study, which are also deemed to contribute to the tumorigenesis of pancreatic cancer. The detailed analyses are listed below.

GO: *0048011*, which describes the *neurotrophin-TRK* receptor signaling pathway, has been predicted to contribute to pancreatic cancer. As we have mentioned above, *neurotrophin-TRK* receptor has been reported to contribute to the growth and progression of human pancreatic cancer [96]. Such evidence validates the efficacy and accuracy of our prediction algorithm. Apart from that, another GO term (GO: *0016772*) describes the transferase activity, especially for the activity of transferring phosphorus-containing groups. During the progression of pancreatic cancer, transferases, especially for those that contribute to the transferring phosphorus-containing groups, have been identified to contain various variants and function abnormally. Take a classical transferase *SphK1* as an example, *SphK1* as a tumor associated protein has been reported to be overexpressed in pancreatic cancer [97]. Recent publications have confirmed that *SphK1* regulates the sphingolipid metabolism and further contributes to the resistance against gemcitabine, a widely used anticancer drug for pancreatic cancer, validating the underlying role of phosphorus associated transferase for pancreatic cancer [97]. Another GO term (GO: *0016303*) describes the specific activity of 1-phosphatidylinositol-3-kinase (*PI3K*). As we have mentioned above, *PI3K* associated pathway has been widely reported to contribute to pancreatic cancer [66]. Similarly, the remaining two GO terms (GO: *0004713* and GO: *0007265*) describe the protein tyrosine kinase (*PTK*) activity and the *Ras* protein signal transduction, respectively, which have also been reported to contribute to pancreatic cancer. The inhibitors for protein tyrosine kinase have been widely used in clinical treatment for pancreatic cancer, implying the driving effect of *PTK* for pancreatic cancer [98]. As for Ras protein signal transduction, proteins of *Ras* family have been widely reported to contribute to tumorigenesis [3, 99]. A specific protein of *Ras* family, *K-RAS*, has been confirmed to be a driver gene for pancreatic cancer, validating the accuracy of our prediction.

According to the analyses listed above, all extracted functional KEGG pathways and GO terms are confirmed to definitely contribute to pancreatic cancer. Some new findings may give new insights for the study of pancreatic cancer or other types of cancer.

4. Conclusions

In this study, effective features, derived from the GO terms and KEGG pathways, were utilized to encode the genes related to pancreatic cancer. After being analyzed by the mRMR method, 22 key features were extracted, corresponding to five GO terms and 17 KEGG pathways. These GO terms and KEGG pathways may be the novel materials to investigate pancreatic cancer. Furthermore, they may also be useful to build an effective computational method for identification of novel genes related to pancreatic cancer. In future, we will try our best in this regard.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this article.

Acknowledgments

This study was supported by the Shanghai Science and Technology Commission (10JC1409000 and 15140904100) and the National Natural Science Foundation of China (31371335).

References

[1] M. Hidalgo, S. Cascinu, J. Kleeff et al., "Addressing the challenges of pancreatic cancer: future directions for improving outcomes," *Pancreatology*, vol. 15, no. 1, pp. 8–18, 2015.

[2] C. Verbeke, M. Löhr, J. S. Karlsson, and M. Del Chiaro, "Pathology reporting of pancreatic cancer following neoadjuvant therapy: challenges and uncertainties," *Cancer Treatment Reviews*, vol. 41, no. 1, pp. 17–26, 2015.

[3] C. D. Logsdon and W. Lu, "The significance of ras activity in pancreatic cancer initiation," *International Journal of Biological Sciences*, vol. 12, no. 3, pp. 338–346, 2016.

[4] A. L. Mihaljevic, C. W. Michalski, H. Friess, and J. Kleeff, "Molecular mechanism of pancreatic cancer—understanding proliferation, invasion, and metastasis," *Langenbeck's Archives of Surgery*, vol. 395, no. 4, pp. 295–308, 2010.

[5] M. R. Schweiger, M. Kerick, B. Timmermann, and M. Isau, "The power of NGS technologies to delineate the genome organization in cancer: from mutations to structural variations and epigenetic alterations," *Cancer and Metastasis Reviews*, vol. 30, no. 2, pp. 199–210, 2011.

[6] T. A. Sohn, G. H. Su, B. Ryu, C. J. Yeo, and S. E. Kern, "High-throughput drug screening of the DPC4 tumor-suppressor pathway in human pancreatic cancer cells," *Annals of Surgery*, vol. 233, no. 5, pp. 696–703, 2001.

[7] M. Guo, H. Wei, J. Hu, S. Sun, J. Long, and X. Wang, "U0126 inhibits pancreatic cancer progression via the KRAS signaling pathway in a zebrafish xenotransplantation model," *Oncology Reports*, vol. 34, no. 2, pp. 699–706, 2015.

[8] Y. Q. Tao, X. P. Zhou, C. Z. Liang et al., "TGF-β 3 and IGF-1 synergy ameliorates nucleus pulposus mesenchymal stem cell differentiation towards the nucleus pulposus cell type through MAPK/ERK signaling," *Growth Factors*, vol. 33, no. 5-6, pp. 326–336, 2015.

[9] L. Zheng, G. Jiang, H. Mei et al., "Small RNA interference-mediated gene silencing of heparanase abolishes the invasion, metastasis and angiogenesis of gastric cancer cells," *BMC Cancer*, vol. 10, article 33, 2010.

[10] A. Krühn, A. Wang, J. H. Fruehauf, and H. Lage, "Delivery of short hairpin RNAs by transkingdom RNA interference modulates the classical ABCB1-mediated multidrug-resistant phenotype of cancer cells," *Cell Cycle*, vol. 8, no. 20, pp. 3349–3354, 2009.

[11] S. Lunardi, R. J. Muschel, and T. B. Brunner, "The stromal compartments in pancreatic cancer: are there any therapeutic targets?" *Cancer Letters*, vol. 343, no. 2, pp. 147–155, 2014.

[12] R. M. Perera and N. Bardeesy, "Pancreatic cancer metabolism: breaking it down to build it back up," *Cancer Discovery*, vol. 5, no. 12, pp. 1247–1261, 2015.

[13] A. Grimont, A. V. Pinho, M. J. Cowley et al., "SOX9 regulates ERBB signalling in pancreatic cancer development," *Gut*, vol. 64, no. 11, pp. 1790–1799, 2015.

[14] M. Javle, Y. Li, D. Tan et al., "Biomarkers of TGF-β signaling pathway and prognosis of pancreatic cancer," *PLoS ONE*, vol. 9, no. 1, Article ID e85942, 2014.

[15] S.-C. Tang and Y.-C. Chen, "Novel therapeutic targets for pancreatic cancer," *World Journal of Gastroenterology*, vol. 20, no. 31, pp. 10825–10844, 2014.

[16] G. Vassaux, A. Angelova, P. Baril, P. Midoux, J. Rommelaere, and P. Cordelier, "The promise of gene therapy for pancreatic cancer," *Human Gene Therapy*, vol. 27, no. 2, pp. 127–133, 2016.

[17] S.-X. Liu, Z.-S. Xia, and Y.-Q. Zhong, "Gene therapy in pancreatic cancer," *World Journal of Gastroenterology*, vol. 20, no. 37, pp. 13343–13368, 2014.

[18] R. J. Brais, S. E. Davies, M. O'Donovan et al., "Direct histological processing of EUS biopsies enables rapid molecular biomarker analysis for interventional pancreatic cancer trials," *Pancreatology*, vol. 12, no. 1, pp. 8–15, 2012.

[19] D. Yeo, H. He, G. S. Baldwin, and M. Nikfarjam, "The role of p21-activated kinases in pancreatic cancer," *Pancreas*, vol. 44, no. 3, pp. 363–369, 2015.

[20] E. K. Colvin and C. J. Scarlett, "A historical perspective of pancreatic cancer mouse models," *Seminars in Cell and Developmental Biology*, vol. 27, pp. 96–105, 2014.

[21] F. Yuan, Y.-H. Zhang, S. Wan, S. Wang, and X.-Y. Kong, "Mining for candidate genes related to pancreatic cancer using protein-protein interactions and a shortest path approach," *BioMed Research International*, vol. 2015, Article ID 623121, 12 pages, 2015.

[22] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," *Nucleic Acids Research*, vol. 44, no. 1, pp. D457–D462, 2016.

[23] M. A. Harris, J. I. Deegan, J. Lomax et al., "The gene ontology project in 2008," *Nucleic Acids Research*, vol. 36, pp. D440–D444, 2008.

[24] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[25] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 27, no. 1, pp. 29–34, 1999.

[26] A. Franceschini, D. Szklarczyk, S. Frankild et al., "STRING v9.1: protein-protein interaction networks, with increased coverage and integration," *Nucleic Acids Research*, vol. 41, no. 1, pp. D808–D815, 2013.

[27] P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and A. Pascual-Montano, "GENECODIS: a web-based tool for finding significant concurrent annotations in gene lists," *Genome Biology*, vol. 8, no. 1, article R3, 2007.

[28] T. Huang, L. Chen, Y.-D. Cai, and K.-C. Chou, "Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property," *PLoS ONE*, vol. 6, no. 9, Article ID e25297, 2011.

[29] T. Huang, J. Zhang, Z.-P. Xu et al., "Deciphering the effects of gene deletion on yeast longevity using network and machine learning approaches," *Biochimie*, vol. 94, no. 4, pp. 1017–1025, 2012.

[30] J. Yang, L. Chen, X. Kong, T. Huang, and Y.-D. Cai, "Analysis of tumor suppressor genes based on gene ontology and the KEGG pathway," *PLoS ONE*, vol. 9, no. 9, Article ID e107202, 2014.

[31] L. Chen, Y. Zhang, M. Zheng, T. Huang, and Y. Cai, "Identification of compound–protein interactions through the analysis of gene ontology, KEGG enrichment for proteins and molecular fragments of compounds," *Molecular Genetics and Genomics*, vol. 291, no. 6, pp. 2065–2079, 2016.

[32] L. Chen, C. Chu, and K. Feng, "Predicting the types of metabolic pathway of compounds using molecular fragments and sequential minimal optimization," *Combinatorial Chemistry & High Throughput Screening*, vol. 19, no. 2, pp. 136–143, 2016.

[33] Y. Zhang, C. Ding, and T. Li, "Gene selection algorithm by combining reliefF and mRMR," *BMC Genomics*, vol. 9, no. 2, article S27, 2008.

[34] L. Liu, L. Chen, Y.-H. Zhang et al., "Analysis and prediction of drug-drug interaction by minimum redundancy maximum relevance and incremental feature selection," *Journal of Biomolecular Structure and Dynamics*, 2016.

[35] Z. Li, X. Zhou, Z. Dai, and X. Zou, "Classification of G-protein coupled receptors based on support vector machine with maximum relevance minimum redundancy and genetic algorithm," *BMC Bioinformatics*, vol. 11, article 325, 2010.

[36] H. Mohabatkar, M. Mohammad Beigi, and A. Esmaeili, "Prediction of GABAA receptor proteins using the concept of Chou's pseudo-amino acid composition and support vector machine," *Journal of Theoretical Biology*, vol. 281, no. 1, pp. 18–23, 2011.

[37] L. Chen, Y. Zhang, T. Huang, and Y. Cai, "Gene expression profiling gut microbiota in different races of humans," *Scientific Reports*, vol. 6, article 23075, 2016.

[38] Z. Liu, J. Han, H. Lv, J. Liu, and R. Liu, "Computational identification of circular RNAs based on conformational and thermodynamic properties in the flanking introns," *Computational Biology and Chemistry*, vol. 61, pp. 221–225, 2016.

[39] L. Chen, C. Chu, J. Lu, X. Kong, T. Huang, and Y.-D. Cai, "Gene ontology and KEGG pathway enrichment analysis of a drug target-based classification system," *PLoS ONE*, vol. 10, no. 5, Article ID e0126492, 2015.

[40] S. A. Korkmaz, M. F. Korkmaz, M. Poyraz, and F. Yakuphanoglu, "Diagnosis of breast cancer nano-biomechanics images taken from atomic force microscope," *Journal of Nanoelectronics and Optoelectronics*, vol. 11, no. 4, pp. 551–559, 2016.

[41] L. Chen, C. Chu, T. Huang, X. Kong, and Y.-D. Cai, "Prediction and analysis of cell-penetrating peptides using pseudo-amino acid composition and random forest models," *Amino Acids*, vol. 47, no. 7, pp. 1485–1493, 2015.

[42] T. Huang, M. Wang, and Y.-D. Cai, "Analysis of the preferences for splice codes across tissues," *Protein and Cell*, vol. 6, no. 12, pp. 904–907, 2015.

[43] S. Wang, Y. Zhang, J. Lu, W. Cui, J. Hu, and Y. Cai, "Analysis and identification of aptamer-compound interactions with a maximum relevance minimum redundancy and nearest neighbor algorithm," *BioMed Research International*, vol. 2016, Article ID 8351204, 9 pages, 2016.

[44] T. Huang, Y. Shu, and Y.-D. Cai, "Genetic differences among ethnic groups," *BMC Genomics*, vol. 16, article 1093, 2015.

[45] L. Chen, C. Chu, Y.-H. Zhang et al., "Analysis of gene expression profiles in the human brain stem, cerebellum and cerebral cortex," *PLoS ONE*, vol. 11, no. 7, Article ID e0159395, 2016.

[46] J. Han, F. Wang, S.-Q. Yuan et al., "Reduced expression of p21-activated protein kinase 1 correlates with poor histological differentiation in pancreatic cancer," *BMC Cancer*, vol. 14, no. 1, article 650, 2014.

[47] X. Bai, X. Zhi, Q. Zhang et al., "Inhibition of protein phosphatase 2A sensitizes pancreatic cancer to chemotherapy by increasing drug perfusion via HIF-1α-VEGF mediated angiogenesis," *Cancer Letters*, vol. 355, no. 2, pp. 281–287, 2014.

[48] N. P. Zimmerman, I. Roy, A. D. Hauser, J. M. Wilson, C. L. Williams, and M. B. Dwinell, "Cyclic AMP regulates the migration and invasion potential of human pancreatic cancer cells," *Molecular Carcinogenesis*, vol. 54, no. 3, pp. 203–215, 2015.

[49] R. Kang, W. Hou, Q. Zhang et al., "RAGE is essential for oncogenic KRAS-mediated hypoxic signaling in pancreatic cancer," *Cell Death and Disease*, vol. 5, no. 10, Article ID e1480, 2014.

[50] Q. Wang, Q. Ni, X. Wang, H. Zhu, Z. Wang, and J. Huang, "High expression of RAB27A and TP53 in pancreatic cancer predicts poor survival," *Medical Oncology*, vol. 32, no. 1, p. 372, 2015.

[51] A. Black, P. F. Pinsky, R. L. Grubb et al., "Sex steroid hormone metabolism in relation to risk of aggressive prostate cancer," *Cancer Epidemiology Biomarkers & Prevention*, vol. 23, no. 11, pp. 2374–2382, 2014.

[52] D.-S. Ming, S. Pham, S. Deb et al., "Pomegranate extracts impact the androgen biosynthesis pathways in prostate cancer models in vitro and in vivo," *The Journal of Steroid Biochemistry and Molecular Biology*, vol. 143, pp. 19–28, 2014.

[53] E. J. Duell, N. Travier, L. Lujan-Barroso et al., "Menstrual and reproductive factors in women, genetic variation in CYP17A1, and pancreatic cancer risk in the European prospective investigation into cancer and nutrition (EPIC) cohort," *International Journal of Cancer*, vol. 132, no. 9, pp. 2164–2175, 2013.

[54] J. Zhou and Y. Du, "Acquisition of resistance of pancreatic cancer cells to 2-methoxyestradiol is associated with the upregulation of manganese superoxide dismutase," *Molecular Cancer Research*, vol. 10, no. 6, pp. 768–777, 2012.

[55] J. A. Pierzynski, M. A. Hildebrandt, A. M. Kamat et al., "Genetic variants in the Wnt/β-catenin signaling pathway as indicators of bladder cancer risk," *The Journal of Urology*, vol. 194, no. 6, pp. 1771–1776, 2015.

[56] N. Yoshida, T. Kinugasa, K. Ohshima et al., "Analysis of Wnt and β-catenin expression in advanced colorectal cancer," *Anticancer Research*, vol. 35, no. 8, pp. 4403–4410, 2015.

[57] M. Ilmer, A. R. Boiles, I. Regel et al., "RSPO2 enhances canonical wnt signaling to confer stemness-associated traits to susceptible pancreatic cancer cells," *Cancer Research*, vol. 75, no. 9, pp. 1883–1896, 2015.

[58] I. Wall and I. G. H. Schmidt-Wolf, "Effect of Wnt inhibitors in pancreatic cancer," *Anticancer Research*, vol. 34, no. 10, pp. 5375–5380, 2014.

[59] Y. Zhao, Y. Yang, J. Trovik et al., "A novel Wnt regulatory axis in endometrioid endometrial cancer," *Cancer Research*, vol. 74, no. 18, pp. 5103–5117, 2014.

[60] Y. Han, Y. Luo, Y. Wang, Y. Chen, M. Li, and Y. Jiang, "Hepatocyte growth factor increases the invasive potential of PC-3 human prostate cancer cells via an ERK/MAPK and ZEB-1 signaling pathway," *Oncology Letters*, vol. 11, no. 1, pp. 753–759, 2016.

[61] M. Beuran, I. Negoi, S. Paun et al., "The epithelial to mesenchymal transition in pancreatic cancer: a systematic review," *Pancreatology*, vol. 15, no. 3, pp. 217–225, 2015.

[62] W. Zhou, A. M. Jubb, K. Lyle et al., "PAK1 mediates pancreatic cancer cell migration and resistance to MET inhibition," *The Journal of Pathology*, vol. 234, no. 4, pp. 502–513, 2014.

[63] M. Quimbaya, E. Raspé, G. Denecker et al., "Deregulation of the replisome factor MCMBP prompts oncogenesis in colorectal carcinomas through chromosomal instability," *Neoplasia*, vol. 16, no. 9, pp. 694–709, 2014.

[64] Y. Matsuda, T. Ishiwata, N. Izumiyama-Shimomura et al., "Gradual telomere shortening and increasing chromosomal instability among PanIN grades and normal ductal epithelia with and without cancer in the pancreas," *PLoS ONE*, vol. 10, no. 2, Article ID e0117575, 2015.

[65] W.-Z. Wang, Q.-H. Pu, X.-H. Lin et al., "Silencing of miR-21 sensitizes CML CD34+ stem/progenitor cells to imatinib-induced apoptosis by blocking PI3K/AKT pathway," *Leukemia Research*, vol. 39, no. 10, pp. 1117–1124, 2015.

[66] Y.-T. Zheng, H.-Y. Yang, T. Li et al., "Amiloride sensitizes human pancreatic cancer cells to erlotinib in vitro through inhibition of the PI3K/AKT signaling pathway," *Acta Pharmacologica Sinica*, vol. 36, no. 5, pp. 614–626, 2015.

[67] N. Walsh, A. Larkin, N. Swan et al., "RNAi knockdown of Hop (Hsp70/Hsp90 organising protein) decreases invasion via MMP-2 down regulation," *Cancer Letters*, vol. 306, no. 2, pp. 180–189, 2011.

[68] Y. I. Chang, X. You, G. Kong et al., "Loss of Dnmt3a and endogenous Kras(G12D+) cooperate to regulate hematopoietic stem and progenitor cell functions in leukemogenesis," *Leukemia*, vol. 29, pp. 1847–1856, 2015.

[69] C. Oancea, B. Rüster, B. Brill et al., "STAT activation status differentiates leukemogenic from nonleukemogenic stem cells in AML and is suppressed by arsenic in t(6;9)-positive AML," *Genes and Cancer*, vol. 5, no. 11-12, pp. 378–392, 2014.

[70] M. A. Macha, S. Rachagani, S. Gupta et al., "Guggulsterone decreases proliferation and metastatic behavior of pancreatic cancer cells by modulating JAK/STAT and Src/FAK signaling," *Cancer Letters*, vol. 341, no. 2, pp. 166–177, 2013.

[71] Y. Bian, Y. Yu, S. Wang, and L. Li, "Up-regulation of fatty acid synthase induced by EGFR/ERK activation promotes tumor growth in pancreatic cancer," *Biochemical and Biophysical Research Communications*, vol. 463, no. 4, pp. 612–617, 2015.

[72] J. Liu, D. Xu, Q. Wang, D. Zheng, X. Jiang, and L. Xu, "LPS induced miR-181a promotes pancreatic cancer cell migration via targeting PTEN and MAP2K4," *Digestive Diseases and Sciences*, vol. 59, no. 7, pp. 1452–1460, 2014.

[73] D. C. Morran, J. M. Wu, N. B. Jamieson et al., "Targeting mTOR dependency in pancreatic cancer," *Gut*, vol. 63, no. 9, pp. 1481–1489, 2014.

[74] V. Nair, S. Sreevalsan, R. Basha et al., "Mechanism of metformin-dependent inhibition of mammalian target of rapamycin (mTOR) and Ras activity in pancreatic cancer: role of specificity protein (Sp) transcription factors," *The Journal of Biological Chemistry*, vol. 289, no. 40, pp. 27692–27701, 2014.

[75] F. Wei, Y. Zhang, L. Geng, P. Zhang, G. Wang, and Y. Liu, "mTOR inhibition induces EGFR feedback activation in association with its resistance to human pancreatic cancer," *International Journal of Molecular Sciences*, vol. 16, no. 2, pp. 3267–3282, 2015.

[76] F. Wang, H. Li, X.-G. Yan et al., "Alisertib induces cell cycle arrest and autophagy and suppresses epithelial-to-mesenchymal transition involving PI3K/Akt/mTOR and sirtuin 1-mediated signaling pathways in human pancreatic cancer cells," *Drug Design, Development and Therapy*, vol. 9, pp. 575–601, 2015.

[77] R. Guo, Y. Wang, W.-Y. Shi, B. Liu, S.-Q. Hou, and L. Liu, "MicroRNA miR-491-5p targeting both TP53 and Bcl-XL induces cell apoptosis in SW1990 pancreatic cancer cells through mitochondria mediated pathway," *Molecules*, vol. 17, no. 12, pp. 14733–14747, 2012.

[78] C. Fiorini, M. Menegazzi, C. Padroni et al., "Autophagy induced by p53-reactivating molecules protects pancreatic cancer cells from apoptosis," *Apoptosis*, vol. 18, no. 3, pp. 337–346, 2013.

[79] F. Wei, Y. D. Zhang, L. Geng, P. Zhang, G. Y. Wang, and Y. Liu, "mTOR inhibition induces EGFR feedback activation in association with its resistance to human pancreatic cancer," *International Journal of Molecular Sciences*, vol. 16, no. 2, pp. 3267–3282, 2015.

[80] K. Taniuchi, M. Furihata, K. Hanazaki et al., "Peroxiredoxin 1 promotes pancreatic cancer cell invasion by modulating p38 MAPK activity," *Pancreas*, vol. 44, no. 2, pp. 331–340, 2015.

[81] R. Subramani, R. Lopez-Valdez, A. Arumugam, S. Nandy, T. Boopalan, and R. Lakshmanaswamy, "Targeting insulin-like growth factor 1 receptor inhibits pancreatic cancer growth and metastasis," *PLoS ONE*, vol. 9, no. 5, Article ID e97016, 2014.

[82] G. Androutsopoulos, G. Adonakis, A. Liava, P. Ravazoula, and G. Decavalas, "Expression and potential role of ErbB receptors in type II endometrial cancer," *European Journal of Obstetrics Gynecology and Reproductive Biology*, vol. 168, no. 2, pp. 204–208, 2013.

[83] A. Grimont, A. V. Pinho, M. J. Cowley et al., "SOX9 regulates ERBB signalling in pancreatic cancer development," *Gut*, vol. 64, no. 11, pp. 1790–1799, 2015.

[84] F. C. Bronfman, O. M. Lazo, C. Flores, and C. A. Escudero, "Spatiotemporal intracellular dynamics of neurotrophin and its receptors. Implications for neurotrophin signaling and neuronal function," in *Neurotrophic Factors*, vol. 220 of *Handbook of Experimental Pharmacology*, pp. 33–65, 2014.

[85] P. A. Forsyth, N. Krishna, S. Lawn et al., "p75 neurotrophin receptor cleavage by α- and γ-secretases is required for neurotrophin-mediated proliferation of brain tumor-initiating cells," *Journal of Biological Chemistry*, vol. 289, no. 12, pp. 8067–8085, 2014.

[86] S. Lawn, N. Krishna, A. Pisklakova et al., "Neurotrophin signaling via TrkB and TrkC receptors promotes the growth of brain tumor-initiating cells," *The Journal of Biological Chemistry*, vol. 290, no. 6, pp. 3814–3824, 2015.

[87] G. Esposito, E. Capoccia, F. Turco et al., "Palmitoylethanolamide improves colon inflammation through an enteric glia/toll like receptor 4-dependent PPAR-α activation," *Gut*, vol. 63, no. 8, pp. 1300–1312, 2014.

[88] S. J. Miknyoczki, A. J. P. Klein-Szanto, and B. A. Ruggeri, "Neurotrophin-Trk receptor interactions in neoplasia: a possible role in interstitial and perineural invasion in ductal pancreatic cancer," *Critical Reviews in Oncogenesis*, vol. 7, no. 1-2, pp. 89–100, 1996.

[89] P. L. Che, Y. F. Yang, X. S. Han et al., "S100A4 promotes pancreatic cancer progression through a dual signaling pathway mediated by Src and focal adhesion kinase," *Scientific Reports*, vol. 5, article 8453, 2015.

[90] A. B. Frey, "Suppression of T cell responses in the tumor microenvironment," *Vaccine*, vol. 33, no. 51, pp. 7393–7400, 2015.

[91] D. S. Shin and A. Ribas, "The evolution of checkpoint blockade as a cancer therapy: what's here, what's next?" *Current Opinion in Immunology*, vol. 33, pp. 23–35, 2015.

[92] J. A. Wallace, F. Li, G. Leone, and M. C. Ostrowski, "Pten in the breast tumor microenvironment: modeling tumor-stroma coevolution," *Cancer Research*, vol. 71, no. 4, pp. 1203–1207, 2011.

[93] R. A. Weinberg, "Coevolution in the tumor microenvironment," *Nature Genetics*, vol. 40, no. 5, pp. 494–495, 2008.

[94] J.-H. Xu, J.-J. Fu, X.-L. Wang, J.-Y. Zhu, X.-H. Ye, and S.-D. Chen, "Hepatitis B or C viral infection and risk of pancreatic cancer: a meta-analysis of observational studies," *World Journal of Gastroenterology*, vol. 19, no. 26, pp. 4234–4241, 2013.

[95] A. Kabir, "Comment on: risk of pancreatic cancer in relation to ABO blood group and hepatitis C virus infection in korea: A Case-Control Study," *Journal of Korean Medical Science*, vol. 28, no. 7, pp. 1114–1115, 2013.

[96] S. J. Miknyoczki, W. Wan, H. Chang et al., "The neurotrophin-trk receptor axes are critical for the growth and progression of human prostatic carcinoma and pancreatic ductal adenocarcinoma xenografts in nude mice," *Clinical Cancer Research*, vol. 8, no. 6, pp. 1924–1931, 2002.

[97] J. Guillermet-Guibert, L. Davenne, D. Pchejetski et al., "Targeting the sphingolipid metabolism to defeat pancreatic cancer cell resistance to the chemotherapeutic gemcitabine drug," *Molecular Cancer Therapeutics*, vol. 8, no. 4, pp. 809–820, 2009.

[98] J. Gillespie, J. F. Dye, M. Schachter, and P. J. Guillou, "Inhibition of pancreatic cancer cell growth in vitro by the tyrphostin group of tyrosine kinase inhibitors," *British Journal of Cancer*, vol. 68, no. 6, pp. 1122–1126, 1993.

[99] S. Goel, J. Huang, and L. Klampfer, "K-Ras, Intestinal homeostasis and colon cancer," *Current Clinical Pharmacology*, vol. 10, no. 1, pp. 73–81, 2015.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 5428737, 13 pages
http://dx.doi.org/10.1155/2016/5428737

Research Article

Functional Region Annotation of Liver CT Image Based on Vascular Tree

Yufei Chen,[1] Xiaodong Yue,[1,2] Caiming Zhong,[3] and Gang Wang[1,4]

[1]*Research Center of CAD, Tongji University, Shanghai 200092, China*
[2]*School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*
[3]*College of Science and Technology, Ningbo University, Ningbo 315211, China*
[4]*School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433, China*

Correspondence should be addressed to Xiaodong Yue; yswantfly@shu.edu.cn

Received 2 June 2016; Revised 24 July 2016; Accepted 4 August 2016

Academic Editor: Quan Zou

Anatomical analysis of liver region is critical in diagnosis and treatment of liver diseases. The reports of liver region annotation are helpful for doctors to precisely evaluate liver system. One of the challenging issues is to annotate the functional regions of liver through analyzing Computed Tomography (CT) images. In this paper, we propose a vessel-tree-based liver annotation method for CT images. The first step of the proposed annotation method is to extract the liver region including vessels and tumors from the CT scans. And then a 3-dimensional thinning algorithm is applied to obtain the spatial skeleton and geometric structure of liver vessels. With the vessel skeleton, the topology of portal veins is further formulated by a directed acyclic graph with geometrical attributes. Finally, based on the topological graph, a hierarchical vascular tree is constructed to divide the liver into eight segments according to Couinaud classification theory and thereby annotate the functional regions. Abundant experimental results demonstrate that the proposed method is effective for precise liver annotation and helpful to support liver disease diagnosis.

1. Introduction

As a noninvasive and painless medical test, Computed Tomography (CT) imaging can provide volumetric image data for liver disease diagnosis, which has been widely used in hospitals [1, 2]. The Computer Aided Diagnosis (CAD) on liver is a complex task that depends on a good understanding of the whole liver system, including the features of liver, vessels and lesions, as well as the anatomical features on specific patients [3–5]. Automatically annotating the functional segments of liver is an effective way to support doctors to study and precisely evaluate the liver system. Although there have been limited research works on liver annotation [6–8], its implementation for real CAD applications is still an arduous task for the following reasons.

The first and fundamental step in liver annotation is organ region segmentation [9]. Some methods segment organ through modeling the shape of liver region [10–13]. However, because of the large variance of liver shapes among different patients, it is difficult for shape-based methods to achieve precise liver segmentation. As another popular segmentation method, the level set methods are very sensitive to liver contour initialization and suffer from iterative computation burden [14–16]. In particular, for the images with tumors and vessels located near liver surface, the level-set-based segmentation tends to be trapped into local optima and eliminates tumors and vessels from main target. Recent research works reveal that the graph cut models have a great potential with the advantages of global optimization and practical efficiency for image segmentation [17]. But, for CT images of livers, the graph cut models cannot handle well seriously blurred boundaries and are incapable of distinguishing the liver regions of similar intensities from neighboring organs [18, 19].

Besides organ region segmentation, vessel segmentation has been another challenging task for liver annotation due to the limitation of vascular imaging equipment and the complexity of liver vascular topology. Florin et al. [20] treated the vessel segmentation as a tracking problem, where vessels were iteratively tracked using information on centerlines

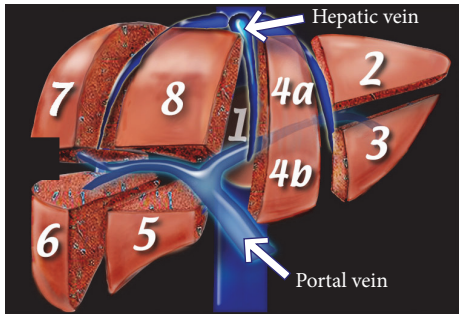


Figure 1: Couinaud classification of liver anatomy.

and local features. The method needs user interaction and requires special routines to handle branch points. Selle et al. [21] presented an intensity-threshold-based method, which defined an optimal intensity threshold through measuring region growing. This kind of methods mainly suits segmenting large vessels. Manniesing et al. proposed a vessel segmentation method based on Hessian matrix, which enhances liver vessels and thereby extracts tubular structures from the organ region [22]. The vesselness of the structure is determined by its geometrical features obtained from the eigensystem of Hessian matrix [23].

The topology of liver vasculature is of importance for recognizing functional segments. Vessel skeleton is widely used to present the topology of vasculature. Generally speaking, there are three types of vessel skeletonization methods. The algorithms based on distance transformation describe well the local structure but cannot guarantee the connectivity and completeness of skeleton [24]. The algorithms based on Voronoi diagram are capable of capturing the topology of the entire vasculature, but the computation of Voronoi skeleton is time-consuming, especially for the medical image containing large-size objects [25]. Compared with the skeletonization methods above, the thinning algorithms can extract the vessel skeleton efficiently and, in the meantime, preserve the connectivity and completeness of the skeleton. Moreover, the thinning algorithm can also record the medial position of the skeleton, which is very helpful to annotate the functional regions [2, 26]. However, only vessel skeletons without quantitative measurement are not sufficient to present the real topology of vasculature [27]. For example, it is difficult for vessel skeletons to keep the important organ features such as length, blood flow direction, and radius of subbranch. Without considering anatomical measurements, the imperfect vessel segmentation results may lead to cyclic skeleton.

Obtaining the topology of liver vasculature, we can partition liver region into functional segments and make the annotation. Referring to the Couinaud classification of liver anatomy, hepatic and portal veins divide liver into eight functionally independent segments as shown in Figure 1 [28]. Each segment has its own vascular inflow, outflow, and biliary drainage. In the center of each segment, there is a branch of the portal vein, hepatic artery, and bile duct. In the periphery of each segment, there is vascular outflow through the hepatic veins. Middle hepatic vein divides the liver into right and left lobes. Right hepatic vein divides the right lobe into anterior and posterior segments. Left hepatic vein divides the left lobe into a medial and lateral part. Portal vein divides the liver into upper and lower segments.

Formulating the Couinaud classification of liver anatomy, Oliveira et al. estimated the planes that best fit each of the three branches of the hepatic veins and the plane that best fits the portal vein. These four planes define the subdivision of the liver in the Couinaud segments [29]. However, the plane-based method does not consider the influence of vascular variation to plane estimation, and it ignores the fact that the separation between liver segments should be a surface. Selle et al. segmented liver through computing the nearest neighbor of different vessel branches [21]. But the result depends on the user defined parameter, which should be manually adjusted under different cases. Schenk et al. used a Laplace model to assign each liver cell to one of the vascular branches to form liver segments [30]. This model suffers from a great computation burden and is not robust enough due to the dependence on vascular branch calibration. Huang et al. designed a fast liver segment method based on the hepatic vessel tree [31]. The method projects the liver and vessel tree to a plane and the classification of liver is achieved by classifying points in the projection plane. Although having high efficiency, the method based on hepatic vessel tree is not a complete functional anatomy, which leads to the inconsistency between the annotated liver segments and the actual blood-supply branch.

As mentioned above, the topology of vasculature can guide the annotation of functional segments. However, because of the high complexity of liver vasculature, it is hard to generate a precise representation of vasculature topology and the computation of geometric structures of all the vessels is always a time-consuming task. In fact, from the view of anatomy, the left and right portal vein branches superiorly and inferiorly project into the center of each segment to supply blood. This means that the functional segments of liver can be recognized by only portal vein branches in it. In the light of this finding, we proposed a hierarchical vascular tree to present the topology of portal veins. The branches of hierarchical vascular tree of portal veins correspond to the functional segments of liver. Based on the obtained functional segments, we annotate the liver region and measure the organ attributes using a standard terminology [6]. The visualization of annotation and the measurements can form a report of liver system for CAD. The contributions of this paper are summarized as follows.

(1) Design a Vessel Tree to Present the Topology of Portal Veins. Connect the topological voxels in vessel skeleton to form a graph. Prune redundant and irrelevant branches of graph to generate a formal vessel tree, which indicates the geometric structures of portal veins and blood flow direction.

(2) Extend the Vessel Tree to a Hierarchical One for Liver Annotation. The vessel tree is hierarchically divided into two levels according to the radius of portal vein branches. The Second Subtree branches are preserved to form functional segments.

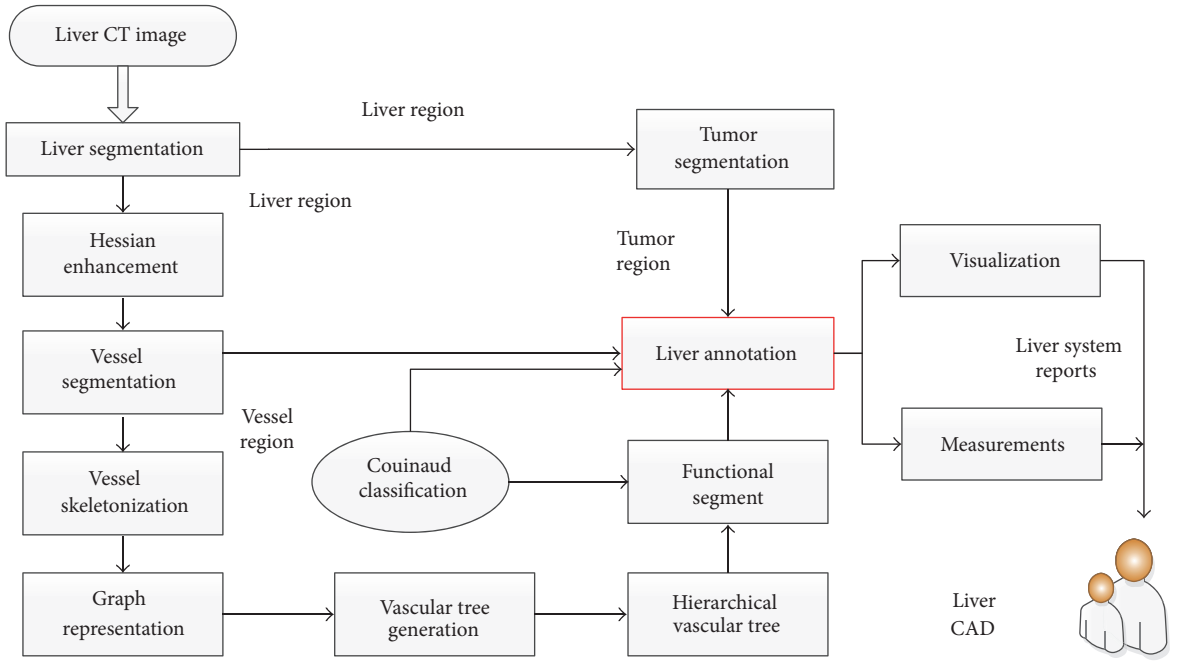


Figure 2: Workflow of the proposed annotation method.

This paper is organized as follows. The workflow of the proposed vessel-tree-based liver annotation method for CT images is described in Section 2. Section 3 gives a detailed introduction of the proposed method, which includes vessel tree generation, liver segment, and annotation. In Section 4, abundant experimental results validate the effectiveness of the proposed annotation method. The paper work is summarized in Section 5.

2. Methodology

In this section, we describe the entire workflow of the proposed functional region annotation method for liver CT images. The workflow consists of four stages. At the first stage, the liver region containing vessels and tumors is segmented from the CT image using an improved graph cut model. More details of liver region segmentation can be found in our previous work [32]. In the liver region, the segmentation for tumors and vessels is further performed. Second, a 3-dimensional thinning algorithm is applied to extract the skeleton of vessels from the segmented vessel region. Based on the skeleton, the topological structure of the vessel system is represented by a vascular tree. Specifically, the vascular tree is formulated by a directed acyclic graph and it can be further extended to a hierarchical version. The hierarchical vascular trees present the connectivity of vessels among the functional segments of liver. According to the connectivity of vascular trees, the liver region can be divided into eight functional segments referring to Couinaud classification theory. At the third stage, integrating the segmentation results of tumors, vessels, and functional segments, we can annotate the liver region and measure the attributes of organ. Finally, a report of liver system which includes the visualization of region annotation and the measurements of functional segments is generated to support doctors to precisely evaluate the liver system. The core steps of the workflow are illustrated in Figure 2; the details will be further introduced in the following section.

3. Vessel-Tree-Based Liver Annotation

As introduced above, the key step of liver annotation is to partition the liver region into multiple functional segments and the topology of vasculature can provide prior information to guide the partition. According to the Couinaud classification theory, the liver system can be divided into eight functional anatomies. It is not necessary to analyze the geometric structure of the entire vasculature; the partition can be performed through constructing a hierarchical vascular tree of portal veins. The methodologies of constructing trees of portal veins and annotating liver segments with hierarchical vessel trees will be elaborated in this section.

3.1. Vessel Tree Generation

3.1.1. Vessel Segmentation. Vessel segmentation is a preliminary step for liver annotation. In this step, first, the liver segmentation is performed on CT image and then the vessels and tumors are further extracted from the segmented liver region. Precise segmentation of liver region is crucial to the subsequent annotation and measurement. In the proposed method, we adopt a semisupervised approach for liver segmentation of CT scans. The segmentation method is based on a graph cut model integrated with domain knowledge, which combines both boundary and regional cues in a global optimization framework. Specifically, the pixels in each CT scan are represented by a graph and the problem of region segmentation is casted to searching for the optimal cut on

graph. The energy function of graph cut is constructed via knowledge-based similarity measure and hard constraints are defined to speed up the graph computation. More details of liver region segmentation can be found in our previous work [32]. We use the same segmentation method to obtain the tumor region.

Extracting vessels from liver region is a prerequisite for the geometrical and structural analysis of vasculature, which is very important for liver disease diagnosis. To segment the regions of vessels, we use Hessian-based filter to enhance the contrast of liver region I_{liver}. The filter is good at searching for tubular-like structures. For discriminating tubular-like structures from blob-like and plate-like structures, the eigenvalues of Hessian matrix for filtering should satisfy condition $\lambda_1 \approx 0$, $\lambda_2 \ll 0$, $\lambda_3 \ll 0$ [22]. The vesselness measure of structures is defined as follows:

$$\begin{aligned}
&\text{Vessel}(\lambda) \\
&= \begin{cases} 0 & \text{if } \lambda_2 \geq 0 \text{ or } \lambda_3 \geq 0 \\ \left(1 - e^{-R_a^2/2\alpha^2}\right) \cdot e^{-R_b^2/2\beta^2} \cdot \left(1 - e^{-R_c^2/2c^2}\right) & \text{otherwise,} \end{cases} \\
&R_a = \frac{|\lambda_2|}{|\lambda_3|}, \\
&R_b = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \cdot \lambda_3|}}, \\
&R_c = \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2},
\end{aligned} \tag{1}$$

where α, β, c are the parameters to control the sensitivity of measure. In the experiments, we set $\alpha = 0.3$, $\beta = 0.7$, $c = I_{\max}/2$. Upper bound $I_{\max}$ corresponds to the brightest intensity value of vessels that can be empirically defined. After the Hessian filtering, a 3D region growing algorithm is utilized on the filtered liver region to segment the liver vasculature. Some morphological operations are adopted to fill small cavities, so as to make the vessel region continuous and smooth. The segmented vessel region consists of portal veins and hepatic veins. As introduced in Section 1, the portal veins are sufficient for distinguishing the functional segments of liver; thus we preserve the connected component of portal vein as binary image I_{vessel}, in which 1 stands for pixels of vasculature and 0 represents the background.

3.1.2. Vessel Skeletonization. To capture the topology of vasculature, first, we should extract the vessel skeleton from the segmented vessel region. Vessel skeletonization aims to reduce the foreground region of binary image I_{vessel} to a skeletal remnant. The skeletonization process should preserve the extent and connectivity of the original vessel region. To satisfy these requirements, we design a 3D thinning algorithm to extract the skeleton of vessels. The skeleton obtained through spatial thinning can preserve the geometric structure of the original vessel region, situate in the middle of I_{vessel}, and be single-voxel wide. Moreover, the thinning-based skeletonization is robust to noisy voxels. The thinning algorithm is implemented through categorizing the voxels.

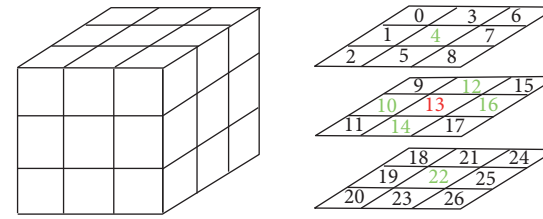


Figure 3: Neighborhood connectivity of a voxel.

In 3D space, a $3 \times 3 \times 3$ lattice is built to examine the local connectivity of a voxel. The 26-neighborhood and 6-neighborhood (marked as green) connectivity is shown in Figure 3. Given voxel v, $N_6(v)$ and $N_{26}(v)$, respectively, denote the 6 neighbors and 26 neighbors of v. For skeletonization, the voxels can be categorized into four types: Border Voxel V_B, Line Voxel V_L, Euler Invariant Voxel V_E, and Simple Voxel V_S. Next, we expatiate the definitions of the voxels of different types.

Definition 1 (Border Voxel). Given vessel voxel $v \in V_{\text{vessel}}$, if at least one of its 6 neighbors has the value of 0, that is belonging to background, the voxel is considered as Border Voxel:

$$\begin{aligned}
V_B &= \left\{v \mid \text{Number}\left(N_6(v) = 0\right) \geq 1,\ v \in V_{\text{vessel}}\right\}, \\
V_{\text{vessel}} &= \left\{v \mid I_{\text{vessel}}(v) = 1\right\}.
\end{aligned} \tag{2}$$

Definition 2 (Line Voxel). Given vessel voxel $v \in V_{\text{vessel}}$, if more than one of its 26 neighbors have the value of 1, that is belonging to vessels, the voxel is considered as Line Voxel:

$$\begin{aligned}
V_L &= \left\{v \mid \text{Number}\left(N_{26}(v) = 1\right) > 1,\ v \in V_{\text{vessel}}\right\}, \\
V_{\text{vessel}} &= \left\{v \mid I_{\text{vessel}}(v) = 1\right\}.
\end{aligned} \tag{3}$$

Definition 3 (Euler Invariant Voxel). Given vessel voxel $v \in V_{\text{vessel}}$, if Euler characteristic χ will not change when removing v from V_{vessel}, the voxel is considered Euler Invariant:

$$\begin{aligned}
V_E &= \{v \mid \chi\left(V_{\text{vessel}} \cap N_{26}(v)\right) \\
&\quad - \chi\left(V_{\text{vessel}} \cap \left\{N_{26}(v) \cup v\right\}\right) = 0,\ v \in V_{\text{vessel}}\}, \\
\chi &= O - H + C, \\
V_{\text{vessel}} &= \left\{v \mid I_{\text{vessel}}(v) = 1\right\},
\end{aligned} \tag{4}$$

where O, H, and C are, respectively, the numbers of connected objects, holes, and cavities in the image.

Definition 4 (Simple Voxel). Given vessel voxel $v \in V_{\text{vessel}}$, if the connectivity in its 26 neighborhoods keeps being invariant when removing v from V_{vessel}, the voxel is considered as Simple Voxel:

$$\begin{aligned}
V_S &= \{v \mid O\left(V_{\text{vessel}} \cap N_{26}(v)\right) \\
&\quad - O\left(V_{\text{vessel}} \cap \left\{N_{26}(v) \cup v\right\}\right) = 0,\ v \in V_{\text{vessel}}\}, \\
V_{\text{vessel}} &= \left\{v \mid I_{\text{vessel}}(v) = 1\right\}.
\end{aligned} \tag{5}$$

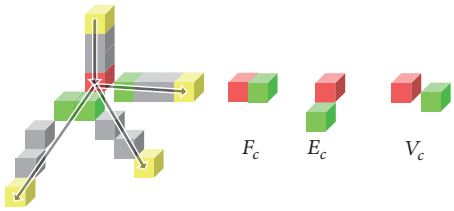


Figure 4: Topological voxels in vessel skeleton.

From the definitions above, we can find that the voxels of border and lines and Euler invariant and simple voxels are redundant for preserving the topology of vasculature. Thus, the skeletonization can be performed through iteratively deleting all those four kinds of voxels from vessel region, until no more change occurs. The output of skeletonization is binary image I_{skeleton} that contains a single-voxel wide skeleton marked as 1, noted as V_{skeleton}.

3.1.3. Graph Representation. To better understand the topology of vasculature, the structure of liver vessels represented by skeleton is further formulated by a graph. The graph consists of a set of vertexes (topological voxels) and connecting edges.

Definition 5 (topological voxels). Topological voxels consist of end-voxels and branch-voxels: end-voxel is the voxel in V_{skeleton} with only one skeleton neighbor and branch-voxel is the skeleton voxel having more than two skeleton neighbors.

As shown in Figure 4, end-voxels can be easily found by counting the skeleton number in its 26 neighborhoods, which are marked as yellow. However, for branch-voxels, there are four candidates that have more than two neighbors (marked in red and green). Among all the possible branch-voxel candidates, the real branch-voxel should have the highest connectivity with all its neighboring branches, as the red voxels shown in Figure 4. The connectivity of neighboring branches can be quantified by the following cost function:

$$\begin{aligned} &v_{\text{candidate}} \\ &\quad = \{v \mid \text{Number}\left(N_{26}(v) = 1\right) \geq 2,\ v \in V_{\text{skeleton}}\}, \\ &\text{cost}(v) \\ &\quad = w_4 \sum \text{NeighborVoxel} + w_3 \sum F_c + w_2 \sum E_c \\ &\qquad + w_1 \sum V_c \left(w_4 > w_3 > w_2 > w_1\right), \\ &\text{branch-voxel} \\ &\quad = \{v_{\text{branch}} \mid \text{cost}\left(v_{\text{branch}}\right) = \max\{\text{cost}\left(v_{\text{candidate}}\right)\}\}, \end{aligned} \tag{6}$$

where $\sum \text{NeighborVoxel}$ means the number of candidates neighbor, $\sum F_c$, $\sum E_c$, and $\sum V_c$ are, respectively, the number of face-connected, edge-connected, and vertex-connected candidates. The voxels of three connected types are also marked in Figure 4. In our implementation, the weighting factors are set as $w_4 = 4$, $w_3 = 3$, $w_2 = 2$, $w_1 = 1$.

Connecting the topological voxels with the corresponding edges, we can construct an undirected graph to present the geometric structure of vessel system. Based on the graph of topological voxels, it is convenient for us to measure the geometric attributes of vessel system. The measurements are summarized in Table 1.

3.1.4. Tree Generation. To simulate the structure of vasculature and indicate the blood flow, we transform the topological graph of vessel system into vessel trees. First, we convert the undirected graph to a directed one through breadth-first-searching from the vessel root. The root of graph V_{root} is the main portal vein, which can be specified as the end-voxel with largest radius summation of it and its branches. It is defined as follows:

$$\begin{aligned} &R(v) = \text{Radius}(v) + \text{MeanRadius}\left(\text{edge}(v)\right), \\ &V_{\text{root}} = \{v_0 \mid R(v_0) = \max\{R(v)\},\ v \in \text{end-voxel}\}. \end{aligned} \tag{7}$$

Since the segmented vessel region contains internal cavities, holes, and bays, the generated graph is always cyclic. There are basically two kinds of loops in the graph: redundant branches (with self-loops) and irrelevant branches (with cycles). The redundant branches are easily removed by deleting branches in which all the skeleton voxels share the same nearest topological voxel. Removing irrelevant branches will be a more complex task. According to anatomy theory, at each ramification point, the blood inflow should be equal to outflow. Based on this, we can match the vessels on blood routine and remove the irrelevant ones. Specifically, the outflow of branches should match the inflow of root vein and the blood flow can be approximated with cross-sectional area of veins, which is square of radius. Figure 5 illustrates a vessel system including one root vein and a branch of five vessels. Among all the connected cyclic edges marked in light blue, we should find a combination set of them that makes outflow most closely match inflow. The vessels out of the combination set are considered as irrelevant branches and should be removed.

The process of determining irrelevant cyclic branches can be formally defined by the following equations:

$$\begin{aligned} &\text{Diff}\left(e_{\text{comb}}\right) = \Big|\text{Radius}\left(e_{\text{in}}\right)^2 \\ &\qquad - \sum \text{Radius}\left(e_{\text{out\_noncyclic}}\right)^2 - \sum \text{Radius}\left(e_{\text{comb}}\right)^2\Big|, \\ &e_{\text{branch}} = \arg\min_{e}\{\text{Diff}\left(e_{\text{comb}}\right)\}, \quad e_{\text{comb}} \subseteq e_{\text{out}_{\text{cyclic}}}, \\ &e_{\text{irrelevant}} = \left\{e \mid e \in e_{\text{out\_cyclic}} \wedge e \notin e_{\text{branch}}\right\}, \end{aligned} \tag{8}$$

where $e_{\text{out\_cyclic}} = \{e^1_{\text{out\_cyclic}}, \ldots, e^m_{\text{out\_cyclic}}\}$ consists of all m connected cyclic branches and e_{comb} denotes a possible combination set of cyclic branches. $\text{Diff}(e_{\text{comb}})$ measure the blood difference between inflow of root vein and outflow of branches. e_{branch} is the combination set whose blood flow matches the inflow of root vein. The branches not contained in e_{branch} are considered as irrelevant cyclic vessel branches.

Table 1: Graph attributes description.

	Attributes	Description
Vertex	Coordinate(v)	3D coordinate values of each vertex v
	Radius(v)	The distance from vertex v to its nearest surface voxel of I_{vessel}
Edge	Length(e)	The actual length of the branch
	Distance(e)	The Euclidean distance between the two vertexes
	MeanRadius(e)	The mean radius of the branch: $\text{MeanRadius}(e) = \sqrt{\text{Volume}(e)/(\pi \cdot \text{Length}(e))}$ Volume(e) is the voxel numbers in e
	Angle(e)	The angle from the parent edge to e
	No.(e)	The edge belonging to which part of vascular system (portal/hepatic vein)

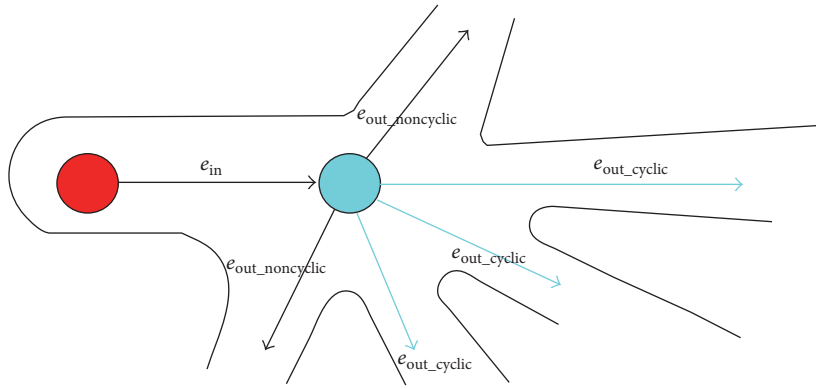


Figure 5: Determining irrelevant vessel branches.

Through removing the redundant branches and irrelevant branches, vascular tree T_{vessel} of portal vessels is generated, which consists of a set of vertexes V_{vessel} and directed edges E_{vessel} to indicate the blood flow.

3.2. Liver Segment and Annotation

3.2.1. Hierarchical Vascular Tree. Considering the blood flow in vasculature, we can further extend the vessel tree to hierarchical vascular tree. As introduced above, vascular tree is formulated by a directed acyclic graph $T_{\text{vessel}} = (V_{\text{vessel}}, E_{\text{vessel}})$; the edge direction represents the blood flow. According to blood-supply amount of branches, the vascular tree can be hierarchically divided into two levels. The First Subtree has the branches of large mean radius and generally denotes the main vessel of liver portal vein. Compared with First Subtree, the Second Subtree denotes the branch of smaller mean radius which is widely distributed in liver segments.

According to the physiological characteristics of vasculature [33], First Subtree generally consists of limited number of main vessels and Second Subtree involves abundant minor vessels of smaller radius. Based on this, we can categorize vessel trees through modeling the distribution of vessel radius. For implementation, we utilize a mixture of Gaussian distributions (GMM) to approximate the vessel radius distribution. Figure 6 illustrates the radius statistics of all the vessel tree branches. Min and Max denote the minimum and maximum radius, respectively. Obviously, there are two

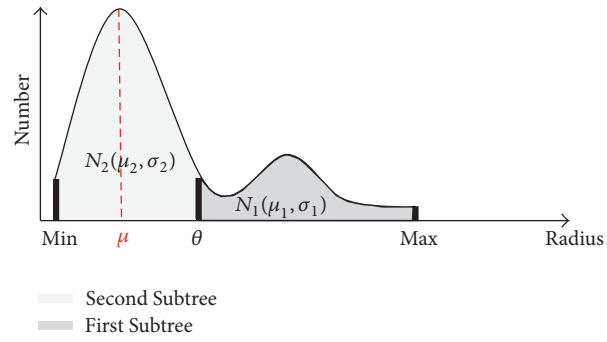


Figure 6: Radius statistics of vessel tree branches.

clusters in the histogram: one centers at small radius and another one locates in the interval of big radius. Having many small branches of similar radius, Second Subtree corresponds to the cluster with higher peak centered at small radius. On the other side, First Subtree is represented by the smaller cluster centered at large radius. Suppose that the radius distributions of two clusters are Gaussian and have forms $N_2(\mu_2, \sigma_2)$ and $N_1(\mu_1, \sigma_1)$, let $\mu = \mu_2$, the radius range of Second Subtree is $[\text{Min}, 2\mu - \text{Min}]$, and the threshold θ that separates First and Second Subtree can be computed as $\theta = 2\mu - \text{Min}$. For easy implementation, the threshold can be set default as $\theta \approx 0.5 \times \text{Max}$. In real applications, the threshold can also be online tuned referring to 3D visualization.

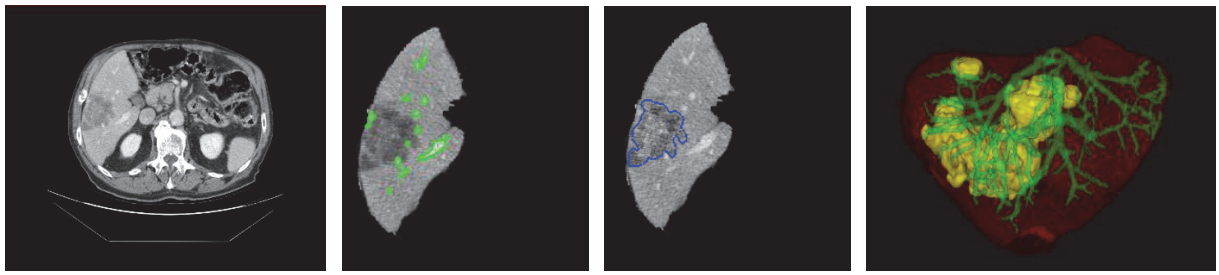


Figure 7: Liver region extraction and visualization.

Based on the distribution of branch radius, we can determine the subtree of vessels at different levels.

Definition 6 (First and Second Subtree). Given vascular tree $T_{\text{vessel}} = (V_{\text{vessel}}, E_{\text{vessel}})$ and threshold θ, for each edge $e \in E_{\text{vessel}}$, if $\theta < \text{Radius}(e) \leq \text{Max}$, edge e belongs to the First Subtree. Otherwise, if $\text{Min} \leq \text{Radius}(e) \leq \theta$, e belongs to the Second Subtree.

As introduced in Section 1, only the connecting branches in Second Subtree will be preserved for the subsequent liver annotation. Among those branches, the Micro Subtree, which represents the trivial structure of vessel system, will be further removed.

Definition 7 (Micro Subtree). Given a tree in Second Subtree, if the number of vertexes in the tree is no more than five, that is $|V_{\text{vessel}}| \leq 5$, the tree is considered as Micro Subtree.

The blood flow of Second Subtree actually presents the circulation of vessel system and thereby indicates the structure of functional segments. We use K-means++ to cluster the root vertexes of the preserved Second Subtrees and the root clustering will induce a partition of liver region. Each cluster of vessel trees corresponds to a functional segment of liver. Since all the vertexes in the same tree belong to the same blood-supply branch, they are definitely in the same segment. Anatomically, the liver is divided into eight segments according to Couinaud classification. Therefore, the number of the vessel tree clusters is set as $K = 8$. After clustering, we complete the branch division of vascular tree.

3.2.2. Liver Annotation. Based on the branch division of hierarchical vascular tree, the liver voxels are iteratively classified into eight parts using a minimum distance classifier [34]. Let B stand for the divided branches in vascular tree, B_i; $i = 1, 2, \ldots, 8$ is ith subtree with vertexes V_{B_i}. For each voxel v in liver region I_{liver}, the classifier computes the minimum distance between v and B_i to determine which branch supplies blood to v, see Definition 8. Through classifying the voxels to different vessel branches, the functional segments of liver are partitioned.

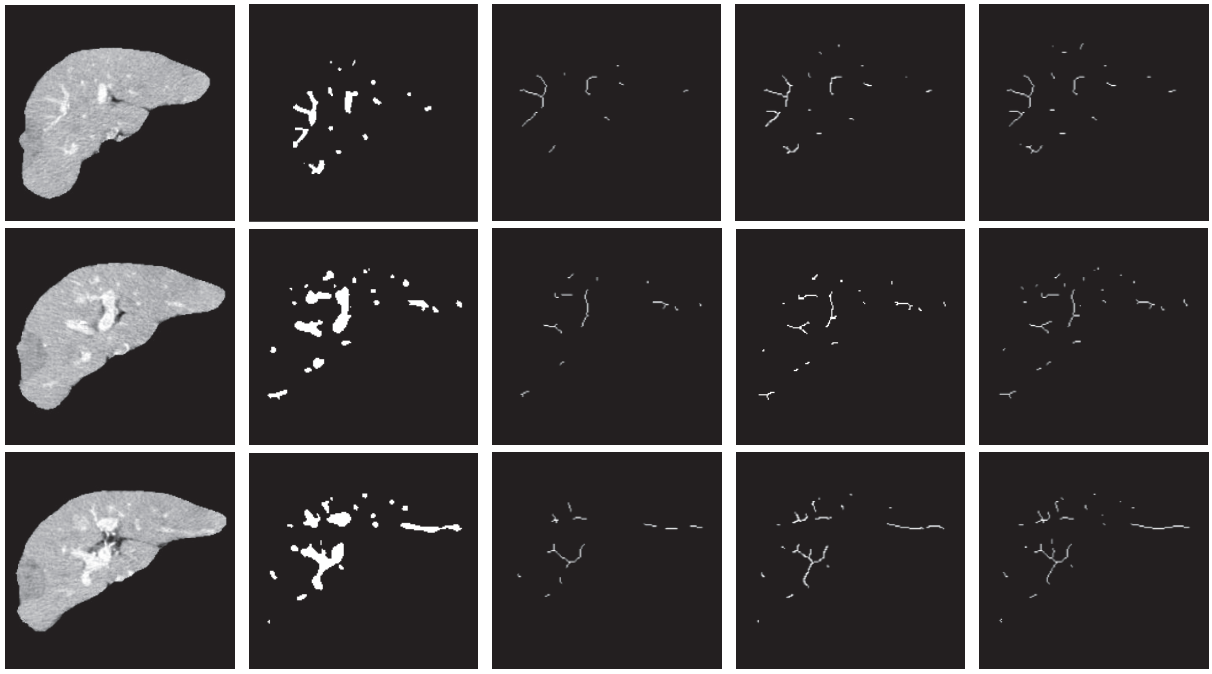
Definition 8 (Branch Distance). For any pair of voxels $v_{\text{liver}} \in I_{\text{liver}}$ and $v_{\text{branch}} \in B$, $\text{Dist}(v_{\text{liver}}, v_{\text{branch}})$ is the Euclidean distance between the two voxels. Based on the voxel distance, we can define the distance between v_{liver} and branch B_i as $\text{MinDist}(v_{\text{liver}}, B_i) = \text{MinDist}(v_{\text{liver}}, v_{\text{branch}})$. The liver voxel v_{liver} will be classified into kth branch if $k = \arg\min_i(\text{MinDist}(v_{\text{liver}}, B_i))$.

Integrating the functional segments of liver and the organ features obtained from the topological graph of vessels, we can generate the report of liver annotation. The annotation report consists of the functional region visualization and the clinical features to describe the characteristics of liver system. Moreover, the clinical features can be categorized into two groups. Global features mainly include the size of liver, vessels, and lesions, as well as the ratio of each segment to liver. Individual features usually consist of anatomical locations, such as the spatial relationship among vasculature, lesions, and liver, and also the segment in which the lesion resides. The annotation results are helpful for doctors to achieve precise evaluation of liver system and reduce the risk of operation.

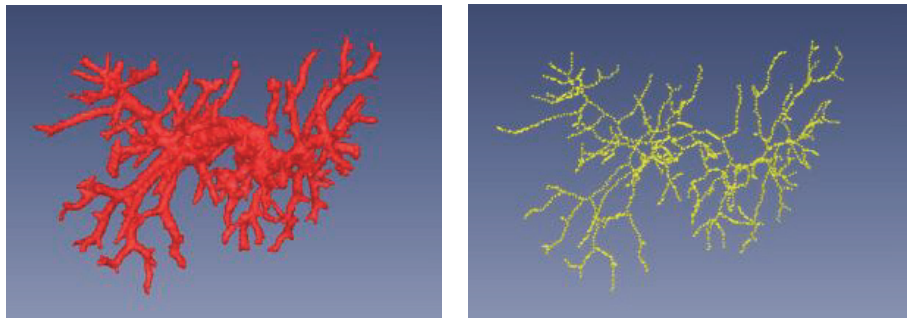
4. Experimental Results

In the experiments, we expect to validate the effectiveness of the proposed vessel-tree-based liver annotation method. The experiments consist of the tests of vessel tree generation and liver annotation. In the test of vessel tree generation, we verify the vessel skeletonization algorithm and the construction of directed acyclic graph to present the topology of vasculature. In the test of liver annotation, we focus on validating the strategy of partitioning the liver region into functional segments based on hierarchical vascular trees. The experiments are performed on CT dataset stored in format of DICOM images. Each volume has an in-plane resolution of 512×512 pixels. The model was implemented based on the toolkits ITK (https://itk.org/) and VTK (http://www.vtk.org/) and was integrated into the MITK framework (http://www.mitk.org/) as a plugin. The computer for program development has an Intel(R) Core(TM)2 Quad CPU (2.66 GHz) and 3.25 GB RAM.

4.1. Test of Vessel Tree Generation. Applying the improved graph cut model to the treated abdominal CT volumes, we can efficiently produce the reliable segmentation results of liver region. An example is given in Figure 7. The first column is one of the original CT slices. The following two columns show the result of liver region segmentation. The



(a)



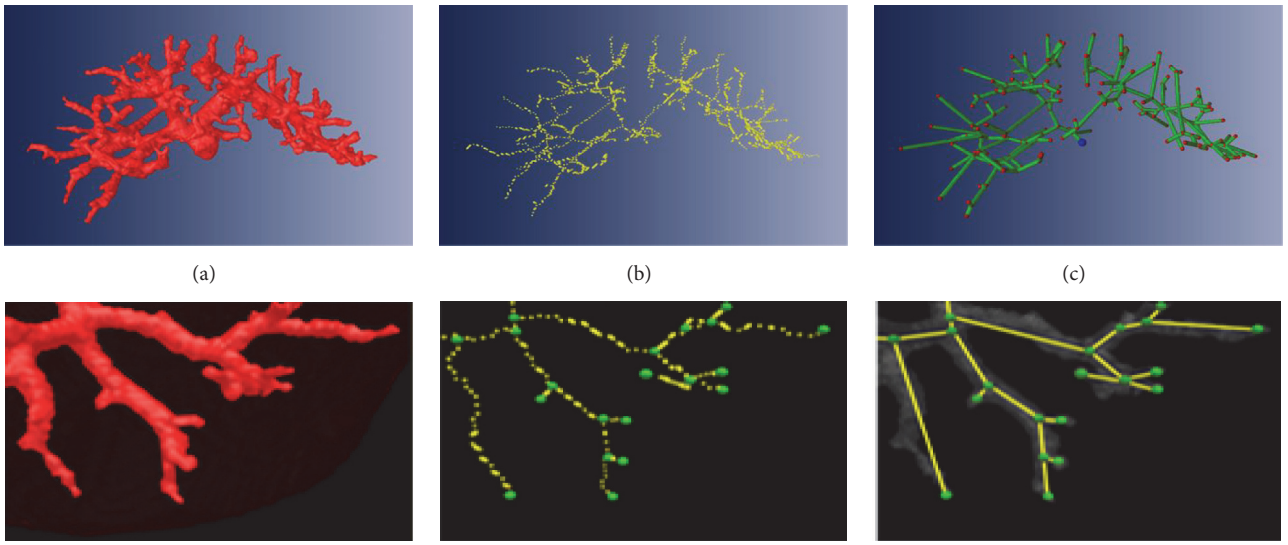
(b) (c)

Figure 8: Skeletonization results and visualization.

average running time for around 70 slices is about 20 s. More experimental analysis can be found in our previous work [32]. The second and third columns present the segmentation of vessel and tumor in liver region on the same slice. The regions of vessels and tumors are marked in green and blue, respectively. Integrating the segmentation results of a series of CT slices, we can form the 3D visualization of the whole liver region; see the last column. We render the liver region in red, tumors in yellow, and vessels in green (both portal vein and hepatic vein). The visualization indicates that the adopted segmentation method is effective in extracting the liver region from original CT images.

Based on the segmented vessel regions, we can construct the skeleton and further the topological graph of vasculature. Various kinds of skeletonization algorithms were applied to build up the vessel skeletons, including distance transform algorithms, Voronoi diagram algorithms, and thinning algorithms. Figure 8(a) shows the vessel skeletons obtained by different skeletonization algorithms. The first column presents three CT images of liver region. The second column illustrates the segmentation of vessel regions. The last three columns show the vessel skeletons generated by distance transform algorithm, Voronoi diagram algorithm, and thinning algorithm, respectively. We find that the thinning algorithm that we use for model implementation can guarantee the connectivity and completeness of the structure of vessel system. Figures 8(b) and 8(c) show 3D visualization of the portal veins and the corresponding skeleton extracted by thinning algorithm.

Compared with the efficiency of different skeletonization algorithms, in 2D space, the average computing time of three algorithms are, respectively, 0.48 s, 0.62 s, and 0.16 s per slice. Taking a CT volume of 124 slices for testing, the computing time of distance transform algorithm is 32.11 s, the thinning algorithm costs 15.79 s, and the Voronoi diagram algorithm runs out of memory. To sum up, the thinning algorithm generates the vessel skeleton in a short time and in the meantime preserves the topology and connectivity of vasculature.



(a) (b) (c)

(d) (e) (f)

Figure 9: Liver vessel tree generation and visualization.

Table 2: Liver segments attributes.

	SegI	SegII	SegIII	SegIV	SegV	SegVI	SegVII	SegVIII
N_{seg}	53062	78287	169190	115116	58760	144737	152095	157323
V_{seg} (mL)	53.06	78.29	169.19	115.12	58.76	144.74	152.10	157.32
R_{seg} (%)	5.71	8.43	18.22	12.40	6.33	15.59	16.38	16.94
R_{tumor} (%)	8.97	0	3.95	0	4.76	0	0	0

As introduced in Section 3, with vessel skeletons, we can construct a directed acyclic graph to present the topology of vasculature. Figure 9 shows the graph representation of the geometric structure of liver portal veins. (a) exhibits the portal veins segmented from liver region. The skeleton result of the portal veins is given in (b). (c) illustrates the directed acyclic graph with the topological voxels marked in red and the tree root marked in blue. Zooming in a local part of liver region, (d) and (e) present the portal veins of local vessel system and its skeleton; (f) shows the induced topological graph. We can find that the proposed method can precisely express the geometric structure of vasculature, even for the minor vessel branches. The time cost for constructing the whole vessel tree is just 1.5 s.

4.2. Test of Liver Segment and Annotation. According to the blood flow, the vessel trees can be divided into two levels. The First Subtree represents the main vessels of liver portal veins and the Second Subtree denotes the minor branches in vasculature. A liver vessel tree constructed in experiments is shown in Figure 10(a). The tree has 192 vertexes marked in green and 191 edges marked in red. Through measuring the radius of vessels, the branches of the tree are categorized into two groups: 26 branches belonging to First Subtree and 165 branches belonging to Second Subtree. The Second Subtree is shown in Figure 10(b). After removing the Micro Subtree, we obtain the final tree as shown in Figure 10(c). The preserved branches of tree are further clustered into 8 classes using K-means++ algorithm. Figure 10(d) illustrates the clustering results, in which the clusters of branches are marked by different colors. From the view of anatomy, each cluster of vessel branches indicates a functional segment. Thus, through clustering the vessel branches, the liver region can be anatomically divided into eight segments as shown in Figure 11. To achieve complete analysis, the liver segments are exhibited from four different views: in visceral surface, hepatic side, hepatic septal, and right lope. The time cost of the whole process is 5 s.

Based on the functional segments, we can compute the basic organ attributes of liver system, such as voxel number of segment N_{seg}, segment volume V_{seg}, volume ratio of segment to liver R_{seg}, and proportion of tumors in segment R_{tumor}. Denoting eight functional segments by SegI~SegVIII, the organ attributes of liver segments shown in Figure 11 are listed in Table 2.

Besides basic organ attributes, we can also compute the attributes of liver lopes to support diagnosis. Table 3 shows the annotation results including the volume information of left/right liver and four liver lopes. It can be inferred from Table 3 that the left lope, which consists of functional segments SegII~SegIV occupies 39.05% of the liver region, and the right lope of segments SegV~SegVIII dominates 55.24%. The statistics are consistent with the anatomical distribution of liver region.

At the final step of the proposed workflow, we should integrate the visualization of liver region, the topological

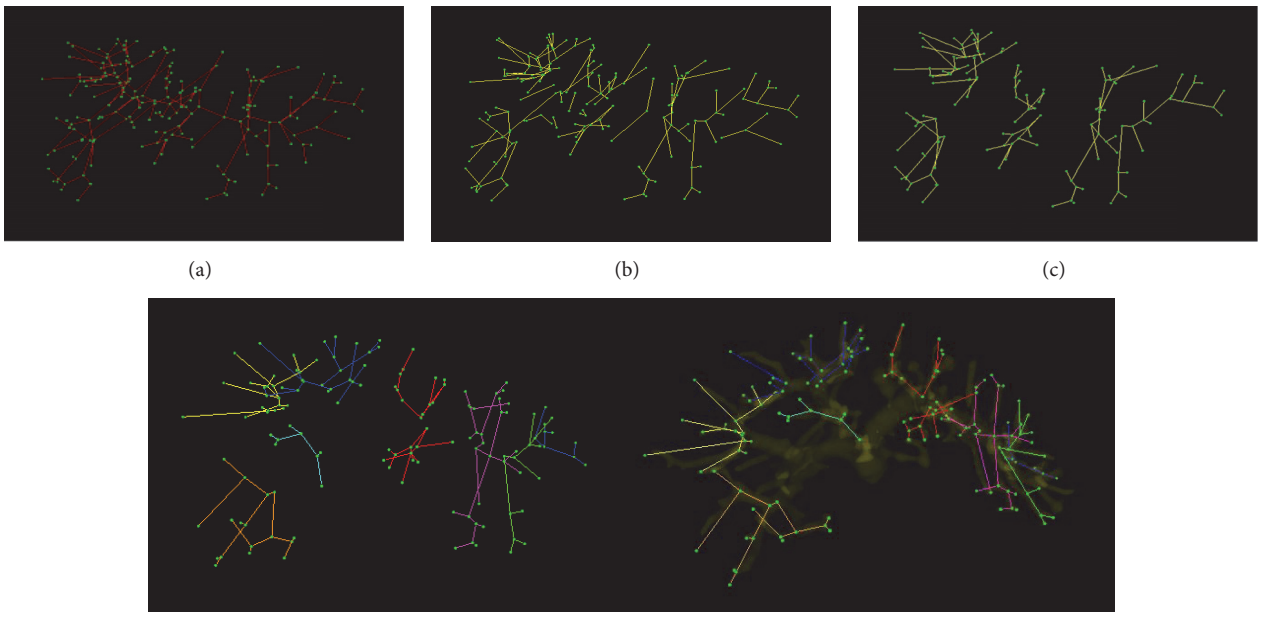


Figure 10: Hierarchical vascular tree generation and division.

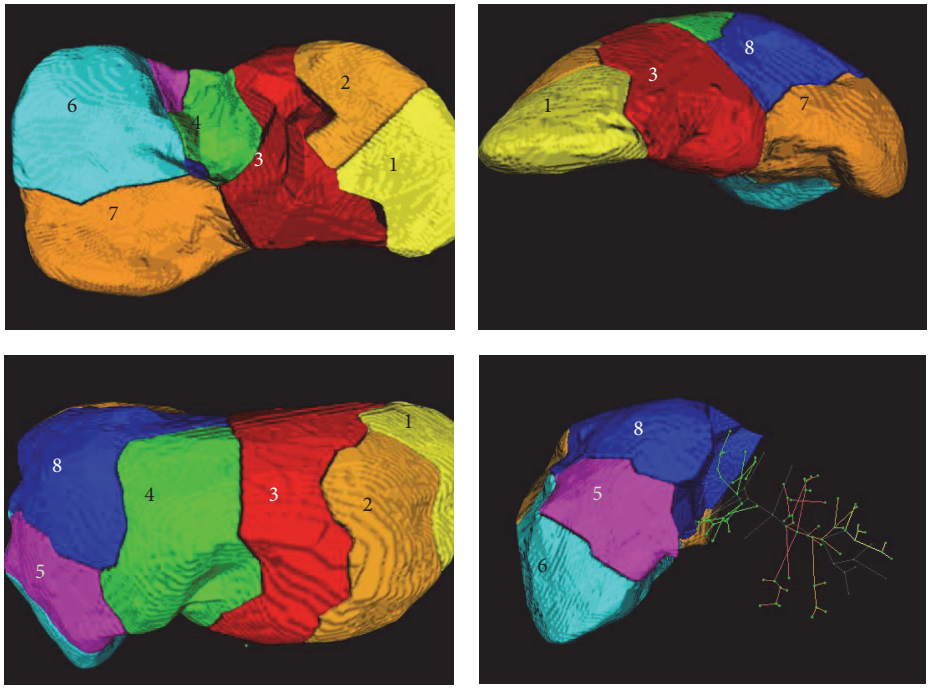
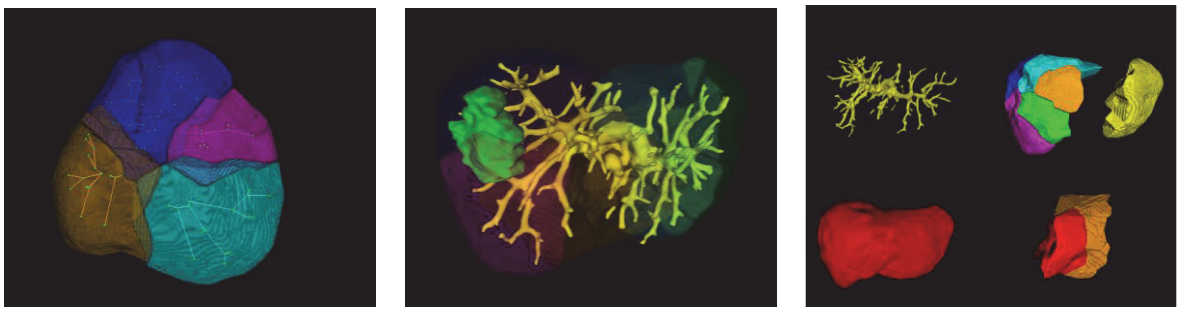


Figure 11: Liver segments and visualization.

structure of vessel tree, the partition of functional segments, and organ attributes to form a report of liver annotation. As shown in Figure 12(a), 3D visualization of liver region intuitively exhibits the spatial relationship between vasculature, lesions, and liver segments. For example, we can easily observe the blood-supply branches of each segment in right liver lobe. Figure 12(b) shows the spatial relationship between tumor, portal vein, and functional segments. Moreover, from the visualization results of liver, vasculature and functional segments can be separated, transformed, rotated, and scaled for complete analysis, as shown in Figure 12(c). Abundant experimental results reveal that the proposed vessel-tree-based liver annotation method can provide visual and measurable information for liver system evaluation and thereby it is effective in supporting diagnosis.

Table 3: Liver annotation results.

Liver			Ratio (%)	
Caudal lobe	SegI		5.71	
Left lobe	Left lateral lobe	SegII	26.65	39.05
		SegIII		
	Left medial lobe	SegIVa	12.40	
		SegIVb		
Right lobe	Right anterior lobe	SegVIII	23.27	55.24
		SegV		
	Right posterior lobe	SegVII	31.97	
		SegVI		



(a) (b) (c)

Figure 12: Three-dimensional visualization of liver.

5. Conclusion

In this paper, we proposed a vessel-tree-based liver annotation method for CT images. At the first step of workflow, the regions of liver, vessels, and tumors are segmented from CT scans. And then a 3D thinning algorithm is applied to obtain the skeleton of liver vessels. Through searching for topological voxels, the skeleton of the portal veins is improved to a directed acyclic graph, that is, vessel tress to present the topology of vasculature. According to the blood flow, the vessel trees are categorized into First and Second Subtrees and the structure of Second Subtrees can indicate the organization of functional segments of liver. In the light of this finding, we cluster the Second Subtrees to partition the liver region into eight functional segments according to Couinaud classification of liver anatomy. Based on the partitioned functional regions, the organ attributes are computed to form quantitative descriptions of liver. At the final step of workflow, we integrate the visualization of liver region, the topological structure of vessel tree, the partition of functional segments, and organ attributes to form a report of liver annotation. Experimental results validate the effectiveness of proposed vessel-tree-based liver annotation method. Our future work will focus on using individual features, such as locational description and shape features, for liver annotation. The liver annotation based on individual organ features is helpful to recognize whether the tumor is benign or malignant.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the Natural Science Foundation of China (nos. 61103070 and 61573235), the Fundamental Research Funds for the Central Universities, and the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University (no. ESSCKF201303).

References

[1] P. Campadelli, E. Casiraghi, and A. Esposito, "Liver segmentation from computed tomography scans: a survey and a new algorithm," *Artificial Intelligence in Medicine*, vol. 45, no. 2-3, pp. 185–196, 2009.

[2] K. Palágyi, J. Tschirren, E. A. Hoffman, and M. Sonka, "Assessment of intrathoracic airway trees: methods and in vivo validation," in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis: ECCV 2004 Workshops CVAMIA and MMBIA, Prague, Czech Republic, May 15, 2004, Revised Selected Papers*, vol. 3117 of *Lecture Notes in Computer Science*, pp. 341–352, Springer, Berlin, Germany, 2004.

[3] S. S. Kumar and D. Devapal, "Survey on recent CAD system for liver disease diagnosis," in *Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies ((ICCICCT '14)*, pp. 763–766, Kanyakumari, India, July 2014.

[4] F. B. Mofrad, R. A. Zoroofi, A. A. Tehrani-Fard, S. Akhlaghpoor, and Y. Sato, "Classification of normal and diseased liver shapes based on spherical harmonics coefficients," *Journal of Medical Systems*, vol. 38, no. 5, article 20, pp. 1–9, 2014.

[5] U. R. Acharya, O. Faust, F. Molinari, S. V. Sree, S. P. Junnarkar, and V. Sudarshan, "Ultrasound-based tissue characterization

and classification of fatty liver disease: a screening and diagnostic paradigm," *Knowledge-Based Systems*, vol. 75, pp. 66–77, 2015.

[6] A. Kumar, S. Dyer, C. Li, P. H. Leong, and J. Kim, "Automatic annotation of liver CT images: the submission of the BMET group to ImageCLEFmed 2014," in *Proceedings of the CLEF Working Notes*, pp. 428–437, Sheffield, UK, September 2014.

[7] I. Nedjar, S. Mahmoudi, A. Chikh, K. Abi-yad, and Z. Boua_a, "Automatic annotation of liver CT image: ImageCLEFmed 2015," in *Proceedings of the Working Notes (CLEF '15)*, pp. 8–11, Toulouse, France, September 2015.

[8] F. Gimenez, J. Xu, Y. Liu et al., "Automatic annotation of radiological observations in liver CT images," in *Proceedings of the AMIA Annual Symposium American Medical Informatics Association*, pp. 257–263, Chicago, Ill, USA, November 2012.

[9] T. Heimann, B. Van Ginneken, M. A. Styner et al., "Comparison and evaluation of methods for liver segmentation from CT datasets," *IEEE Transactions on Medical Imaging*, vol. 28, no. 8, pp. 1251–1265, 2009.

[10] D. Kainmuller, T. Lange, and H. Lamecker, "Shape constrained automatic segmentation of the liver based on a heuristic intensity model," in *Proceedings of the MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pp. 109–116, Brisbane, Australia, October 2007.

[11] T. Heimann, S. Münzing, H. P. Meinzer, and I. Wolf, "A Shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation," in *Proceedings of the Information Processing in Medical Imaging*, pp. 1–12, Kerkrade, The Netherlands, January 2007.

[12] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou, "Towards robust and effective shape modeling: sparse shape composition," *Medical Image Analysis*, vol. 16, no. 1, pp. 265–277, 2012.

[13] H. Ling, S. K. Zhou, Y. Zheng, B. Georgescu, M. Suehling, and D. Comaniciu, "Hierarchical, learning-based automatic liver segmentation," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.

[14] A. Zidan, N. I. Ghali, A. E. Hassanien, H. Hefny, and J. Hemanth, "Level set-based CT liver computer aided diagnosis system," *International Journal of Imaging and Robotics*, vol. 9, no. 1, pp. 26–36, 2012.

[15] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.

[16] N. Xu, N. Ahuja, and R. Bansal, "Object segmentation using graph cuts based active contours," *Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 210–224, 2007.

[17] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 1–27, 2012.

[18] L. Massoptier and S. Casciaro, "Fully automatic liver segmentation through graph-cut technique," in *Proceedings of the Engineering in Medicine and Biology Society*, pp. 5243–5246, Lyon, France, August 2007.

[19] T. Kitrungrotsakul, X.-H. Han, and Y.-W. Chen, "Liver segmentation using superpixel-based graph cuts and restricted regions of shape constrains," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '15)*, pp. 3368–3371, IEEE, Quebec, Canada, September 2015.

[20] C. Florin, N. Paragios, and J. Williams, "Particle filters, a quasi-monte carlo solution for segmentation of coronaries," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (CMICCAI '05)*, pp. 246–253, Palm Springs, Calif, USA, October 2005.

[21] D. Selle, B. Preim, A. Schenk, and H.-O. Peitgen, "Analysis of vasculature for liver surgical planning," *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1344–1357, 2002.

[22] R. Manniesing, M. A. Viergever, and W. J. Niessen, "Vessel enhancing diffusion. A scale space representation of vessel structures," *Medical Image Analysis*, vol. 10, no. 6, pp. 815–825, 2006.

[23] P. T. H. Truc, M. A. U. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, "Vessel enhancement filter using directional filter bank," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 101–112, 2009.

[24] W. H. Hesselink and J. B. T. M. Roerdink, "Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2204–2217, 2008.

[25] A. Beristain and M. Graña, "Pruning algorithm for Voronoi skeletons," *Electronics Letters*, vol. 46, no. 1, pp. 39–41, 2010.

[26] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, "Building skeleton models via 3-D medial surface/axis thinning algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994.

[27] D. Fan, A. Bhalerao, and R. Wilson, "Comparative assessment of retinal vasculature using topological and geometric measures," in *Medical Imaging 2005: Image Processing*, vol. 5747 of *Proceedings of SPIE*, pp. 1104–1111, San Diego, Calif, USA, February 2005.

[28] L. Ruskó and G. Bekes, "Liver segmentation for contrast-enhanced MR images using partitioned probabilistic model," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, no. 1, pp. 13–20, 2011.

[29] D. A. B. Oliveira, R. Q. Feitosa, and M. M. Correia, "Automatic couinaud liver and veins segmentation from CT images," in *Proceedings of the International Conference on Bio-Inspired Systems and Signal*, pp. 249–252, Madeira, Portugal, January 2008.

[30] A. Schenk, S. Zidowitz, H. Bourquain et al., "Clinical relevance of model based computer-assisted diagnosis and therapy," in *Medical Imaging*, vol. 6915 of *Proceedings of SPIE*, San Diego, Calif, USA, March 2008.

[31] S.-H. Huang, B.-L. Wang, M. Cheng, W.-L. Wu, X.-Y. Huang, and Y. Ju, "A fast method to segment the liver according to Couinaud's classification," in *Medical Imaging and Informatics*, X. Gao, H. Müller, M. J. Loomes, R. Comley, and S. Luo, Eds., vol. 4987 of *Lecture Notes in Computer Science*, pp. 270–276, Springer, 2008.

[32] Y. Chen, Z. Wang, J. Hu, W. Zhao, and Q. Wu, "The domain knowledge based graph-cut model for liver CT segmentation," *Biomedical Signal Processing and Control*, vol. 7, no. 6, pp. 591–598, 2012.

[33] H. K. Hahn, B. Preim, D. Selle, and H.-O. Peitgen, "Visualization and interaction techniques for the exploration of vascular

structures," in *Proceedings of the Visualization (VIS '01)*, pp. 395–578, IEEE, San Diego, Calif, USA, October 2001.

[34] H. G. Debarba, D. J. Zanchet, D. Fracaro, A. Maciel, and A. N. Kalil, "Efficient liver surgery planning in 3D based on functional segment classiffication and volumetric information," in *Proceedings of the Engineering in Medicine and Biology Society*, pp. 4797–4800, Buenos Aires, Argentina, August 2010.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 9406259, 14 pages
http://dx.doi.org/10.1155/2016/9406259

Research Article

Convolutional Deep Belief Networks for Single-Cell/Object Tracking in Computational Biology and Computer Vision

Bineng Zhong,[1] Shengnan Pan,[1] Hongbo Zhang,[1] Tian Wang,[1] Jixiang Du,[1] Duansheng Chen,[1] and Liujuan Cao[2]

[1]*Department of Computer Science and Engineering, Huaqiao University, Xiamen, China*
[2]*School of Information Science and Technology, Xiamen University, Xiamen, China*

Correspondence should be addressed to Bineng Zhong; bnzhong@hqu.edu.cn

Received 1 June 2016; Revised 14 August 2016; Accepted 14 September 2016

Academic Editor: Dariusz Mrozek

In this paper, we propose deep architecture to dynamically learn the most discriminative features from data for both single-cell and object tracking in computational biology and computer vision. Firstly, the discriminative features are automatically learned via a convolutional deep belief network (CDBN). Secondly, we design a simple yet effective method to transfer features learned from CDBNs on the source tasks for generic purpose to the object tracking tasks using only limited amount of training data. Finally, to alleviate the tracker drifting problem caused by model updating, we jointly consider three different types of positive samples. Extensive experiments validate the robustness and effectiveness of the proposed method.

1. Introduction

Cell and object tracking have been an active research area in computational biology [1, 2] and computer vision [3–6] with a lot of practical applications, for example, drug discovery, cell biology, intelligence video surveillance, self-driving vehicles, and robotics. Despite much progress made in recent years, designing robust cell and object tracking methods is still a challenging problem due to appearance variations caused by nonrigid deformation, illumination changes, occlusions, dense populations and cluttered scenes, and so forth. Therefore, one key component in cell and object tracking is to build a robust appearance model that can effectively handle the above-discussed challenges.

Over the years, discriminative model based appearance modeling has been popular due to its effectiveness in extrapolating from relatively small number of training samples. Most existing methods focus on two aspects to construct a robust discriminative appearance model: feature representation and classifier construction.

Feature Representation. Tremendous progress has been made in feature representation for cell and object tracking. Typically, a number of cell and object tracking methods employ simple color [7] or intensity [8] histograms for feature representation. Recently, a variety of more complicated handcrafted feature representations has been applied in cell and object tracking, such as subspace-based features [9, 10], Haar features [11–13], local binary pattern (LBP) [14], histogram of gradient (HoG) [15, 16], scale invariant feature transformation (SIFT) [17], and shape features [18]. While the above handcrafted features have achieved great success for their specific tasks and data domains, they are not effective to capture the time-varying properties of cell and object appearances.

Classifier Construction. Designing a good classifier plays another important role in the robust appearance model. The typical classifiers include ensemble learning [19–22], structural learning [18, 23], support vector machine [24], sparse coding [25, 26], coupled minimum-cost flow [27], and semi-supervised learning [28, 29]. However, due to the fact that appearance variations are highly complex, most of these classifiers suffer from their shallow structures.

In this paper, inspired by the remarkable progress in deep learning [30–34] for big data analysis [35], we propose a robust cell and object tracking method (termed CDBN-Tracker) that relies on convolutional deep belief networks

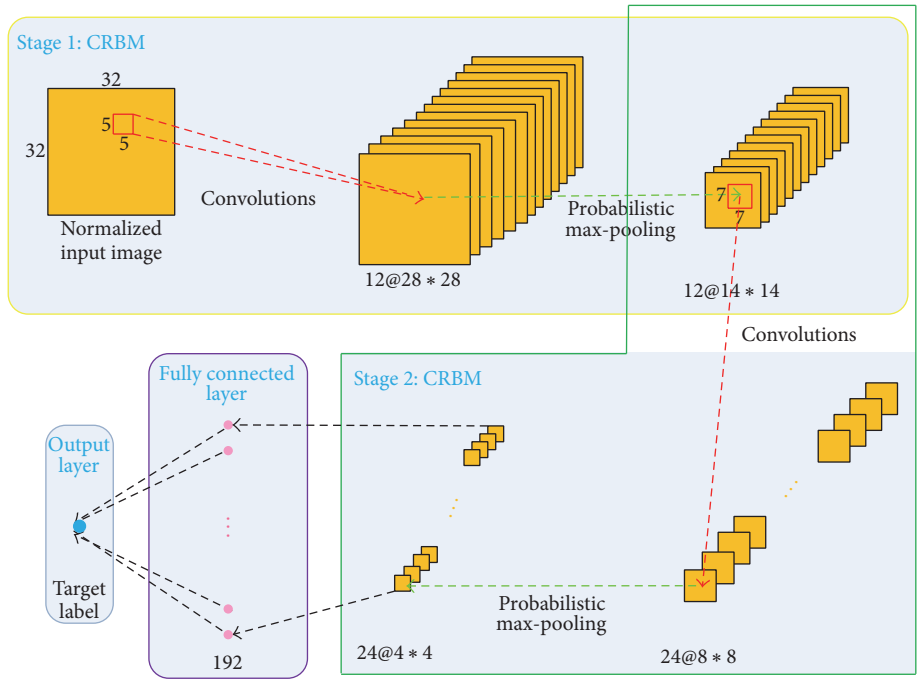


Figure 1: Illustration of how the proposed CDBNTracker constructs an appearance model from a convolutional deep belief network. The raw input image is fed to a 2-stage convolutional deep belief network consisting of two max-pooling CRBMs and one fully connected layer. Each CRBM contains a filter bank layer and a probabilistic max-pooling layer, respectively. The outputs of the second stage are followed by one fully connected layer with 192 units.

(CDBNs) to address both limitations raised from handcrafted feature and shallow classifier designs. As shown in Figure 1, our CDBNTracker is built upon the CDBNs trained from raw pixels, which is composed of two convolutional restricted Boltzmann machines (CRBMs) and one fully connected layer. To the best of our knowledge, it is the first time to apply DBN-like network architectures into cell and object tracking.

The CRBMs are stacked on top of one another, each of which contains a filter bank layer and a probabilistic max-pooling layer, respectively. With end-to-end training, CDBNTracker automatically learns hierarchical features in a supervised manner, making it extremely discriminative in appearance modeling. We further propose a transferring strategy to better reuse the pretrained CDBN features on the cell and object tracking tasks. This allows the CDBNTracker to learn cell or object-specific feature representations.

Last but not least, we propose a systematic and heuristic solution to alleviate the tracker drifting problem for the CDBNTracker. In particular, we classify the positive samples into three categories to update the CDBN-based appearance models, that is, ground-truth samples (nonadaptive samples obtained in the first frame), long-term samples (moderately adaptive samples obtained in the most recent frames), and short-term samples (highly adaptive samples collected in the current frame). The advantages of our CDBNTracker are threefold.

(1) Our CDBNTracker follows the cutting-edge deep learning framework. And the proposed CDBNTracker differs from the recent deep learning-based trackers by using multilayer CDBNs with local tied weights to reduce the model complexity under the scarcity of training samples. Furthermore, we transfer generic visual patterns as good initialization in our tracker to alleviate the "the first frame labeled" problem.

(2) We develop a new model update strategy to effectively alleviate the tracker drift. In addition to short-term and first frame information, long-term information is selectively memorized for updating the current model state to alleviate the abrupt appearance changes.

(3) Different from most previous trackers which use handcrafted features and shallow models, our CDBNTracker is online trained with a multilayer CDBN in a supervised manner which is more discriminative and descriptive.

The rest of the paper is organized as follows. An overview of the related work is given in Section 2. Section 3 introduces how to learn a data-driven cell or object appearance model from a CDBN. The detailed tracking method is then described in Section 4. Experimental results are given in Section 5. Finally, we conclude this work in Section 6.

2. Related Work

Over the past decades, a huge amount of cell and object tracking methods have been proposed [1–6]. Since the proposed tracking method focuses on utilizing deep learning to construct robust appearance models for cell and object tracking, in this section, we firstly review online generative and discriminative tracking methods. Then, cell tracking methods are also briefly introduced. Finally, we discuss the

current progress using deep learning for the cell and object tracking research.

2.1. Online Cell and Object Tracking

2.1.1. Generative Models. Generative tracking models describe the cell and object appearances via a statistical model using the reconstruction errors. Some representative methods include mean shift-based tracker [7], integer programming-based tracker [8], PCA-based tracker [9], sparse coding-based trackers [25, 26], GMM-based tracker [36], multitracker integration [37], and structured learning-based tracker [18]. While generative tracking methods usually succeed in less complex scenes due to the richer appearance models used, they are prone to fail in complex scenes without considering the discriminative information between the foregrounds and backgrounds.

2.1.2. Discriminative Models. On the other hand, discriminative tracking models typically view cell and object tracking as a binary classification task. Thus, they aim to explicitly learn a classifier which can discriminate the cell or object from the surrounding backgrounds. In [38], an ensemble learning-based tracker is proposed, in which a group of weak classifiers is adaptively constructed for object tracking. In [11], an online boosting-based tracker is proposed for object tracking. Grabner and Bischof [11] extend a boosting algorithm for online discriminative tracking. However, online learning-based trackers is prone to the tracker drifting problem. Recently, various discriminative tracking methods have been proposed to alleviate the drifting problem. Using an anchor assumption (i.e., the current tracker does not stray too far from the initial appearance model), Matthews et al. [39] develop a partial solution for the template-based trackers. In [20], a semi-supervised boosting algorithm is applied to online object tracking by using a prior classifier. It is obvious that the semi-supervised boosting-based tracker is not robust to very large changes in appearance. In [28], Babenko et al. present a multiple instance boosting-based tracking method. Hare et al. [12] employ an online kernelized structured output support vector machine for object tracking. In [23], an online structured support vector machine-based tracker is proposed. Duffner and Garcia [29] use a fast adaptive tracking method to track nonrigid objects via cotraining. A number of attempts have been made to apply transfer learning to object tracking [40, 41]. However, they may be limited by using handcrafted features which cannot be simply adapted according to the new observed data obtained during the tracking process.

2.1.3. Cell Tracking Methods. Recently, with the rapid development of cell and computational biology, several cell tracking methods have been proposed. In [8], Li et al. employ integer programming for multiple nuclei tracking in quantitative cancer cell cycle analysis. In [18], Lou et al. propose an active structured learning method for multicell tracking, in which a compatibility function (i.e., global affinity measure) is designed to associate hypotheses and score. In [27], Padfield et al. present a cell tracking method via coupling minimum-cost flow for high-throughput quantitative analysis.

2.2. Deep Learning for Cell and Object Tracking.

Due to the powerful representation abilities, deep learning [33] has recently drawn more and more attention in computational biology, medical imaging analysis [42], computer vision [32, 43], speech recognition [31], natural language processing, and so forth. Deep belief networks [44], autoencoders, and convolutional neural networks [32] are the three representative deep learning methods for computational biology and computer vision.

Despite the fact that tremendous progress has been made in deep learning, only a limited number of tracking methods using the feature representations from deep learning have been proposed so far [42, 45–50]. In [46], a convolutional neural network-based tracking method is proposed for tracking humans. However, once the model is trained, it is fixed during tracking due to the features being learned during offline training. In order to handle the left ventricle endocardium in ultrasound data, Carneiro and Nascimento [42] fuse multiple dynamic models and deep learning architecture in a particle filtering framework. In [51], without using the fully connect layers in convolutional neural networks, a fully convolutional neural network is proposed for object tracking. In [47], a convolutional neural network-based tracking method is presented, in which a pretrained network is transferred to an interested object. Ma et al. [48] combine the pretrained VGG features [52] and correlation filters to improve location accuracy and robustness in object tracking. In [49], a multidomain convolutional neural network-based tracking method is proposed. In [50], Chen et al. propose a convolutional neural network-based tracking method, which transfers the pretrained features from a convolutional neural network to the tracking tasks. Compared to Chen's method using a convolutional neural network, our CDBNTracker explores a different deep learning algorithm (i.e., a convolutional deep belief network, CDBN) for single-cell/object tracking. Instead of using convolutional neural networks, an autoencoder-based tracking method [45] is proposed, in which the generic image features are firstly learned from an offline dataset and then transferred to a specific tracking task.

In this paper, we focus on how to construct an effective CDBN-based appearance model for discriminative single-cell and object tracking in cell biology and computer vision, respectively. To the best of our knowledge, it is the first time to apply DBN-like network architectures to single-cell and object tracking.

3. Object Appearance Model

In this section, we address the problem of how to learn a data-driven appearance model from a CDBN.

3.1. CRBM and CDBN.

The CDBN [43] is a hierarchical generative model composed of one visible (observed) layer and many hidden layers, that is, several CRBMs stacked on top of one another. A statistical relationship between the units in the lower layer is learned by each hidden layer unit; the higher layer representations tend to become more complex and abstract. Following the notations of Lee et al. [43], we briefly review the CRBM and CDBN.

The CRBM is an extension of the RBM which fully connects the hidden layer and visible layer. To capture the 2D structural of image and incorporate translation invariance, the CRBM shares the weights between the hidden units and the visible units among all locations in the hidden units. The CRBM consists of a visible (input) layer and a hidden layer. In this paper, we use real-valued visible units $v \in R^{n_V \times n_V}$ and binary-valued hidden units $h \in \{0,1\}^{n_H \times n_H}$. Denote $W^k \in R^{n_W \times n_W}$ as the kth convolution filter weight between a hidden unit and the visible unit; $b_k \in R$ as a bias variable shared among hidden units and $c \in R$ as a visible bias shared among visible units. The energy function of the probabilistic max-pooling CRBM with real-valued visible units can then be defined as

$$E(v,h) = \frac{1}{2}\sum_{i,j=1}^{n_V} v_{ij}^2 - \sum_{k=1}^{K}\sum_{i,j=1}^{n_H}\sum_{r,s=1}^{n_W} h_{ij}^k W_{rs}^k v_{i+r-1,j+s-1} - \sum_{k=1}^{K} b_k \sum_{i,j=1}^{n_H} h_{ij}^k - c\sum_{i,j=1}^{n_V} v_{ij}, \quad \text{s.t.} \sum_{(i,j)\in B_a} h_{ij}^k \le 1, \ \forall k, a, \tag{1}$$

where K is the number of convolution filters and $B_a = \{(i,j) \mid h_{ij}^k$ belonging to the block $a\}$ is a $C \times C$ block of locally neighboring hidden units h_{ij}^k that are pooled to a pooling unit p_a^k. It should be noted that probabilistic max-pooling enables the CRBM to incorporate max-pooling-like behavior, while allowing probabilistic bottom-up and top-down inference [43]. The conditional probability distributions can be calculated as follows:

$$P\left(h_{ij}^k = 1 \mid v\right) = \frac{\exp\left(I\left(h_{ij}^k\right)\right)}{1 + \sum_{(i',j')\in B_a} \exp\left(I\left(h_{i'j'}^k\right)\right)},$$

$$P\left(v_{ij} \mid h\right) = N\left(\left(\sum_k W^k *_f h^k\right)_{ij} + c, 1\right), \tag{2}$$

$$P\left(p_a^k = 0 \mid v\right) = \frac{1}{1 + \sum_{(i',j')\in B_a} \exp\left(I\left(h_{i'j'}^k\right)\right)},$$

where $I(h_{ij}^k) = (\widetilde{W}^k *_v v)_{ij} + b_k$, $*_f$ is a full convolution, $*_v$ is a valid convolution, and $\widetilde{W}_{ij}^k = W_{n_W-i+1,n_W-j+1}^k$.

Typically, the CRBM is highly overcomplete due to the fact that the hidden layer of the CRBM contains K groups of units, each roughly with size of the visible layer (input image). To avoid the risk of learning trivial solutions by the CRBM, a sparsity penalty term is added to the log-likelihood objective function of the training data. Consequently, each hidden unit group has a mean activation close to a small constant. Finally, after the greedy and layer-wise training, we stack the CRBMs to form a CDBN.

3.2. Learning Cell and Object Appearance Models from CDBNs. In this paper, we view object tracking as an online transfer learning problem and use the CDBN to construct the cell and object appearance model due to its capacity for automatically learning a hierarchical feature representation. As shown in Figure 2, the key idea is to use the internal CDBN features as a generic and middle-level image representation, which can be pretrained on one dataset (the source task here CIFAR-10 [53]) and then reused on the tracking tasks.

More specifically, for the source task, we pretrain a CDBN with two CRBM layers followed by one fully connected layer from the CIFAR-10 natural image dataset [53]. The CIFAR-10 dataset is a labeled subset of the 80 million tiny images, containing 60,000 images and ten classes. Each CRBM layer is composed of a hidden and pooling layer. The first CRBM layer consists of 12 groups of $5*5$ convolution filters, while the second CRBM layer consists of 288 groups of $7*7$ convolution filters. The pooling ratio is set to 2 for each pooling layer. The target sparsity for the first and second CRBM layer is set as 0.003 and 0.005, respectively. The fully connected layer FC3 has 192 units. The output layer has size 10 equal to the number of target categories. It can be seen from Figure 3(a) that the learned filters in first CRBM layer (top) are oriented and localized edge filters, while the learned filters in second CRBM layer (bottom) selectively respond to contours, corners, angles, and surface boundaries in the images.

After pretraining on the source task, the parameters of layers h1, p1, h2, p2, and FC3 are first transferred to the tracking task. Then, we remove the output layer with 10 units and add an output layer with one unit. Finally, the newly designed CDBN is retrained (fine-tuned) on the training data from a specific tracking task to learn a cell or object appearance model. This simple yet effective transferring schema enables the proposed CDBNTracker to tackle the domain changes in training tasks. To empirically illustrate the efficacy of the transfer, we check the fine-tuned filters trained on the training data from a specific tracking task. Figure 3(b) shows the fine-tuned filters trained on the training data from the first frame of the motorRolling sequence [6]. Figure 3(c) shows the fine-tuned filters trained on the training data from the first frame of the Mitocheck sequence [54]. It can be seen from both Figures 3(b) and 3(c) that, in addition to edge, corner, and junction detectors, the transferred CDBN also adaptively learns different and complicated features according to the newly observed data.

4. Single-Cell and Object Tracking via CDBNs (CDBNTracker)

In this section, we present a single-cell and object tracking method, in which the CDBN-based appearance model is effectively incorporated into a particle filtering framework. The particle filtering framework consists of two key components.

(1) A dynamic model $p(x_t \mid x_{t-1})$ generates candidate samples based on previous particles. In this paper, the dynamic model between two consecutive frames is assumed to be a Gaussian distribution: $p(x_t \mid x_{t-1}) = N(x_t; x_{t-1}, \sum)$, where $\sum$ denotes a covariance matrix and $x_t = (p_t^x, p_t^y, w_t, h_t)$ denotes the cell or object state parameters composed of the

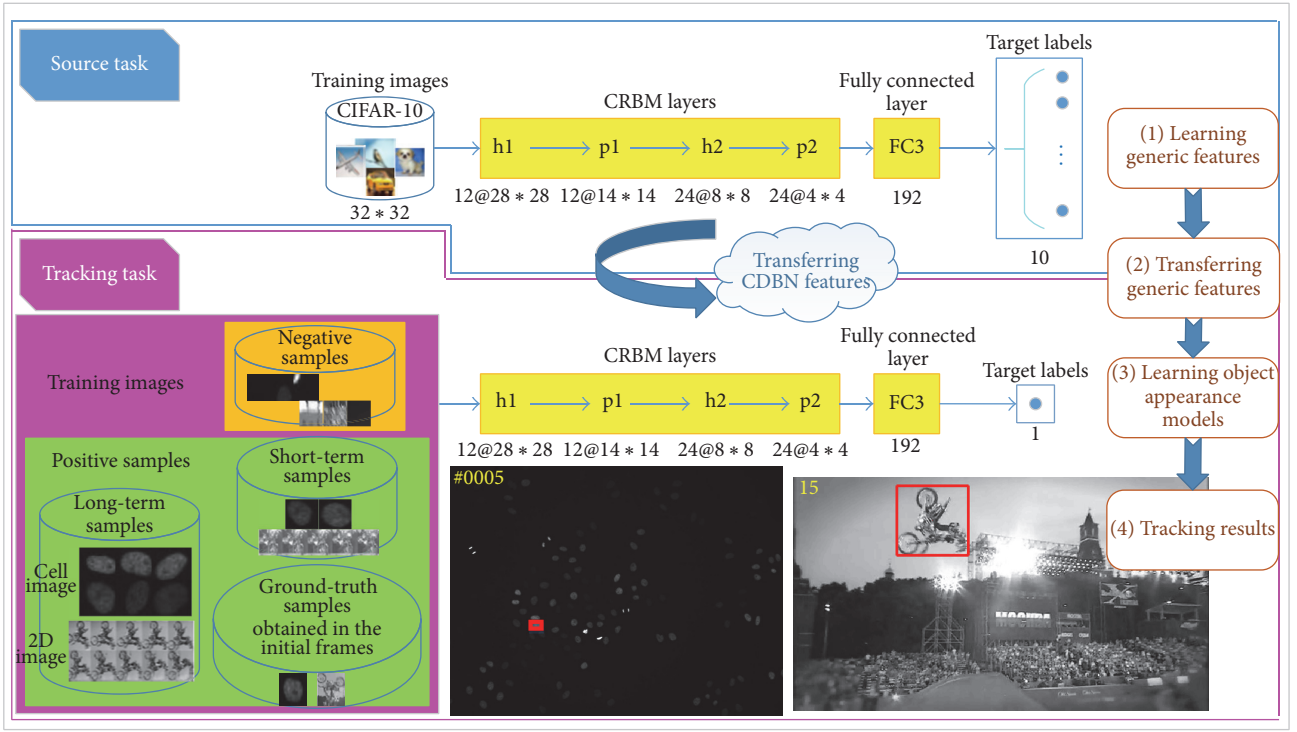


Figure 2: Learning object appearance models by transferring the CDBN features. First, the CDBN is pretrained on the source task (CIFAR-10 classification, top row). Then, the pretrained parameters of the internal layers of the CDBN (h1–FC3) are then transferred to the tracking task (bottom row). To achieve the transfer and construct the cell and object appearance models, we remove the output layer with 10 units and add an output layer with one unit. Furthermore, to alleviate the drifting problem, we treat training samples differently to update the cell and object appearance models.

horizontal coordinate, vertical coordinate, width, and height, respectively.

(2) An observation model $p(y_t \mid x_t)$ calculates the similarity between candidate samples and the cell or object appearance model. In this paper, the proposed CDBN-based appearance model is used to estimate the score of the likelihood function $p(y_t \mid x_t)$.

To capture the appearance variations, the observation model (i.e., the CDBN-based appearance model) needs to be updated over time. Therefore, to alleviate the tracker drifting problem, we classify the positive samples into three categories: ground-truth samples (nonadaptive samples obtained in the first frame), long-term samples (moderately adaptive samples obtained in the most recent frames via FIFO schema), and short-term samples (highly adaptive samples collected in the current frame). We assume the ground-truth set of positive samples obtained in the first frame to be $s_g^+ = \{x_{1,i}^+\}_{i=1}^{N_1^+}$. The long-term set of positive samples obtained in the most recent frames is denoted as $s_{lt}^+ = \{x_{t-i}^+\}_{i=1}^{T}$, where T is the buffer size of temporal sliding window. The sets of negative samples and short-term positive samples collected in the current frame are denoted as $s_t^- = \{x_{t,i}^-\}_{i=1}^{N_t^-}$ and $s_t^+ = \{x_{t,i}^+\}_{i=1}^{N_t^+}$, respectively. At each frame t, we update the CDBN-based appearance model using s_g^+, s_{lt}^+, s_t^+, and s_t^-.

Finally, a summary of our CDBN-based tracking method for single-cell and object tracking is described in Algorithm 1.

Algorithm 1 (single-cell and object tracking via learning and transferring CDBN).

Initialization

(1) Pretrain a CDBN on the CIFAR-10 dataset.

(2) Acquire manual labels in the first frame. Collect the ground-truth set of positive samples s_g^+ and negative samples s_1^-.

(3) Resize each positive/negative image patch to $32 * 32$ pixels.

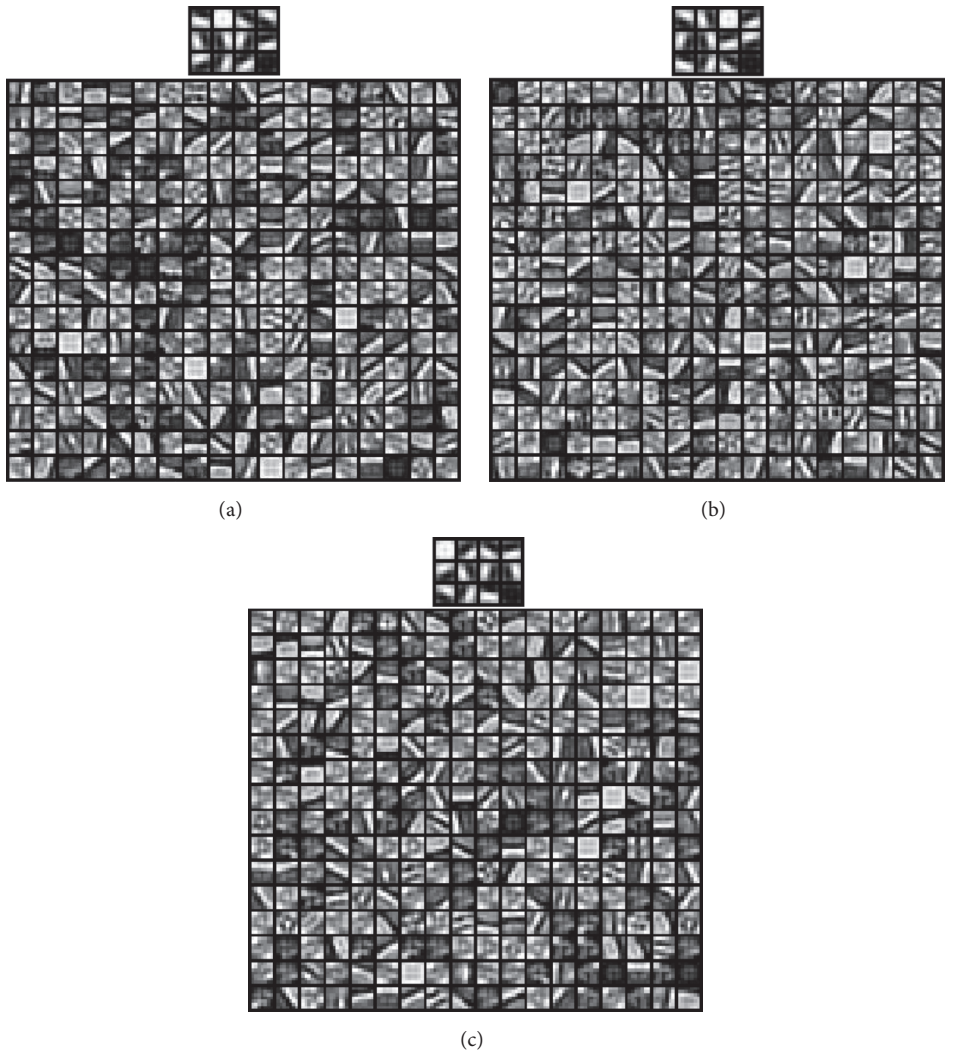
(4) Construct the CDBN-based appearance model via fine-tuning and transferring the pre-trained CDBN using s_g^+ and s_1^-.

(5) Initialize the particle set $\{x_1^i, w_1^i\}_{i=1}^{N_1}$ at time $t = 1$, where $w_1^i = 1/N_1$, $i = 1, \ldots, N_1$

(6) Set the maximum buffer size T for long-term positive samples s_{lt}^+.

For $t = 2$ to the End of the Video

(1) *Prediction*: for $i = 1, \ldots, N_1$, generate $x_t^i \sim p(x_t \mid x_{t-1}^i)$



(a) (b) (c)

Figure 3: (a) The pretrained filters in first (top) and second (bottom) CRBM layer learned from CIFAR-10 natural images. (b) The fine-tuned filters in first (top) and second (bottom) CRBM layer learned from the training data of motorRolling sequence [6]. (c) The fine-tuned filters in first (top) and second (bottom) CRBM layer learned from the training data of Mitocheck sequence [54].

(2) *Likelihood evaluation*: for $i = 1, \ldots, N_1$, let $w_t^i = w_{t-1}^i p(y_t \mid x_t^i)$.

(3) Determine the optimal object state x_t^* as the particle with the maximum weight.

(4) *Resample*: Normalize the weights and compute the covariance of the normalized weights. If this variance exceeds one threshold, then $\beta_j \sim \{w_t^i\}_{i=1}^{N_1}$ and replace $\{x_t^i, w_t^i\}_{i=1}^{N_1}$ with $\{x_t^{\beta_j}, 1/N_1\}_{j=1}^{N_1}$.

(5) *Update*:

(5.1) Set short-term positive samples s_t^+ at time t as the image patches having the 10 highest confidences (estimated by the likelihood evaluation).

(5.2) Select negative samples s_t^- at time t.

(5.3) Update the long-term set of positive samples $s_{lt}^+ = s_{lt}^+ \cup \{x_t^*\}$.

(5.4) If the size of s_{lt}^+ is larger than T, then s_{lt}^+ is truncated to keep the last T elements.

(5.5) Update the CDBN-based appearance model based on s_g^+, s_{lt}^+, s_t^+ and s_t^-.

End For

5. Experiments

In this section, we first introduce the setting of our experiments. Then, we test the proposed CDBNTracker (CDBN-10-2), which has two CRBM layers followed by one fully connected layer and is pretrained on the CIFAR-10 dataset, the Mitocheck dataset [54], and CVPR2013 tracking benchmark [6], respectively. The Mitocheck dataset from the Mitocheck project [54] is a time-lapse microscopic image sequence. The Mitocheck sequence contains higher cell density, larger intensity variability, and illumination variations. The CVPR2013

tracking benchmark contains 50 fully annotated image sequences. Each image sequence is tagged by a number of attributes indicating the presence of different challenging aspects, such as illumination variation, scale variation, occlusion, deformation, and background clutters. To show the advantage of the CDBN-10-2 over the other competing trackers, we compare it with some state-of-the-art tracking methods including a related deep learning tracker (DLT) [45]. Moreover, the efficacy of different positive samples is empirically evaluated by a carefully designed experiment. Finally, to examine the impact of the different training data and CDBN architecture, we evaluate the performance of the proposed CDBNTracker as the amount of training data and the number of CRBM layers in CDBN grow.

5.1. Experiment Setting. The proposed CDBN-10-2 is implemented in Matlab on a HP Z800 workstation with an Intel® Xeon® E5620 2.40 GHz processor and 12 G RAM. The number of particles in particle filtering is set to 1,000. Each image observation of the target object is normalized to a $32 * 32$ patch. The buffer size of temporal sliding window is set as 25. To train the CDBN, we adopt stochastic gradient descent with momentum. In each frame, the number of epochs needed to train the CDBN is 500. The learning rate and momentum are set as $1e-1$ and 0.5, respectively. The average processing speed is about 5 fps at the resolution of $320 * 240$ pixels without using GPUs. Consequently, the proposed CDBN-10-2 can achieve real-time processing speed if the GPUs (e.g., tesla k40) are used. The main memory cost is from the number of parameters in the proposed CDBN model. However, the CDBN shares weights among all locations in an image. Thus, the number of parameters in our CDBN model is significantly reduced (to only $6.9 * 10^4$). We only need a small-scale dataset (e.g., CIFAR-10 with 60,000 images) to pretrain our CDBN model, which can then be effectively transferred to the tracking tasks. The proposed CDBN model can obtain a better performance if we use other large-scale datasets for initialization (e.g., Caltech-256 or ImageNet). In our experiments, if the memory space of one parameter is one byte in Matlab, we find the memory cost is about $6.9 * 10^4/1024 = 70$ KB. We use the same parameters for all of the experiments.

For performance evaluation, we test the proposed CDBN-10-2 on the Mitocheck dataset [54] and CVPR2013 tracking benchmark, respectively. In the CVPR 2013 tracking benchmark, 30 publicly available trackers are evaluated. We follow the protocol used in the benchmark, in which the evaluation is based on two different metrics: the precision plot and success plot. The precision plot shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth, and a representative precision score (threshold = 20 pixels) is used for ranking. Another metric contains the overlap precision over a range of thresholds. The overlap precision is defined as the percentage of frames where the bounding box overlap exceeds a given threshold varied from 0 to 1. In contrast to the precision plot, the trackers are ranked using the area under curve (AUC) in the success plot. In addition, we compare the CDBN-10-2 against the deep learning-based tracker (DLT) of Wang and Yeung [45].

5.2. Comparison with Other Trackers on the CVPR2013 Tracking Benchmark

5.2.1. Quantitative Evaluation. The quantitative comparison results of all the trackers are listed in Figure 4 where only the top 10 trackers are shown for clarity. The values in the legend of the precision plot are the relative number of frames in the 50 sequences where the center location error is smaller than a threshold of 20 pixels. The values in the legend of the success plot are the AUC. In both the precision and success plots, the proposed CDBN-10-2 is the state-of-the-art compared to all alternative methods. Our CDBN-10-2 outperforms Struck by 2.8% in mean distance precision at the threshold of 20 pixels, while it outperforms SCM by 4.3% with the AUC. The robustness of our CDBN-10-2 lies in the hierarchical and deep structure-based appearance model which is discriminatively trained online to account for each variation.

5.2.2. Temporal and Spatial Robustness Evaluation. It is known that a tracker may be sensitive to initialization. To analyse a tracker's robustness to initialization, we follow the evaluation protocol proposed in [6] by perturbing the initialization temporally (referred to as temporal robustness, TRE) and spatially (referred to as spatial robustness, SRE). For TRE, each sequence is partitioned into 20 segments, whereas, for SRE, 12 different bounding boxes are evaluated for each sequence. The precision and success plots for TRE and SRE are shown in Figure 5. The proposed CDBN-10-2 performs favorably compared to other trackers on the temporal and spatial robustness evaluation.

5.2.3. Attribute-Based Evaluation. The object appearance variations may be caused by illumination changes, occlusions, pose changes, cluttered scenes, moving backgrounds, and so forth. To analyse the performance of trackers for each challenging factor, the benchmark annotates the attributes of each sequence and constructs subsets with 11 different dominant attributes, namely, *illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter*, *and low resolution*. We perform a quantitative comparison with the 30 state-of-art tracking methods on the 50 sequences annotated with respect to the aforementioned attributes. Due to space limitation, we show the representative success scores of SRE for different subsets divided based on main variation of the target object in Table 1. As we can see, the proposed CDBN-10-2 performs favorably on the 11 attributes.

5.2.4. Qualitative Evaluation. Qualitative comparison with the top 10 trackers (on four typical sequences) is shown in Figure 6. Meanwhile, for more close-view evaluation, we show the corresponding examples of the center distance error per frame in Figure 7 with the top 10 trackers compared, which show that our method can transfer the pretrained CDBN features to the specific target object well.

Recall that the pretrained CDBN is learned entirely from natural scenes, which are completely unrelated to the tracking task. However, according to the overall tracking results, the proposed CDBN-10-2 outperforms the competing methods.

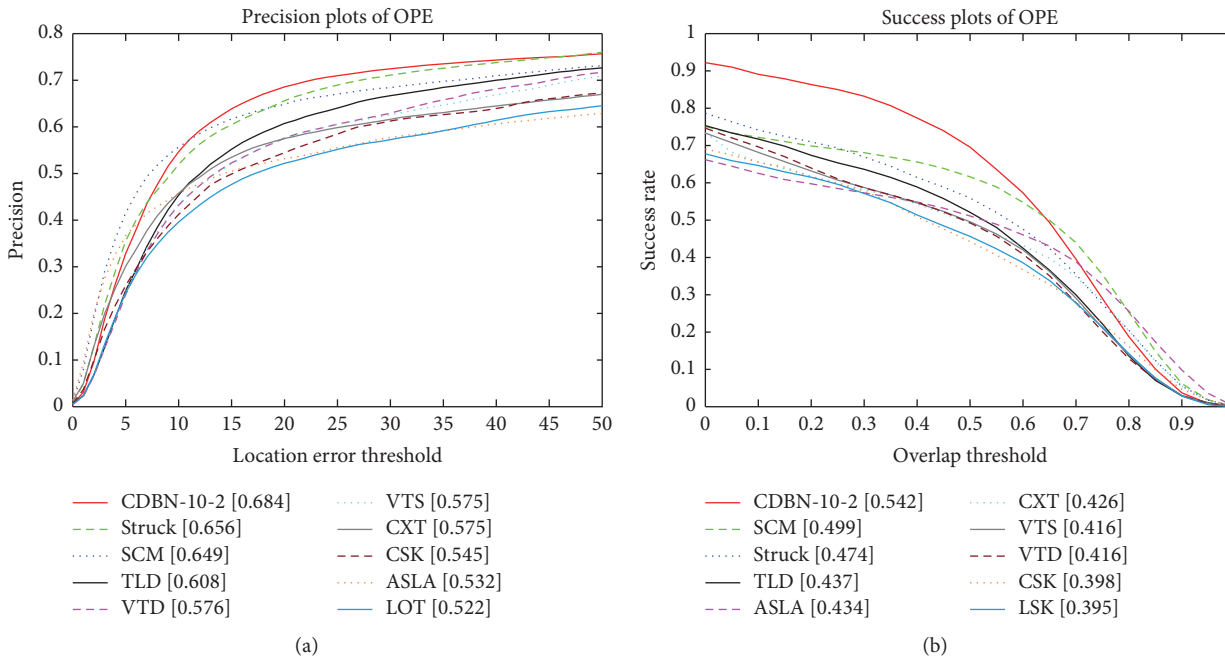


Figure 4: The precision and success plots of quantitative comparison for the 50 sequences in the CVPR2013 tracking benchmark [6]. The performance score of each tracker is shown in the legend. The proposed CDBN-10-2 (in red) obtains better or comparable performance against state-of-the-art tracking methods.

Table 1: A representative success score (AUC) of SRE for different subsets divided based on main variation of the target object. Only the top 5 trackers are displayed for clarity.

Image attributes	Ranking				
	The first	The second	The third	The fourth	The fifth
Fast motion (17)	CDBN-10-2 (0.472)	Struck (0.451)	TLD (0.385)	CXT (0.348)	OAB (0.322)
Background clutter (21)	CDBN-10-2 (0.414)	ASLA (0.410)	Struck (0.408)	SCM (0.387)	VTD (0.377)
Motion blur (12)	CDBN-10-2 (0.530)	Struck (0.452)	TLD (0.392)	CXT (0.354)	DFT (0.325)
Deformation (19)	CDBN-10-2 (0.451)	Struck (0.398)	ASLA (0.386)	DFT (0.364)	CPF (0.362)
Illumination variation (25)	CDBN-10-2 (0.440)	ASLA (0.405)	Struck (0.396)	SCM (0.389)	VTS (0.378)
In-plane rotation (31)	CDBN-10-2 (0.422)	CXT (0.410)	Struck (0.410)	ASLA (0.405)	SCM (0.399)
Low resolution (4)	CDBN-10-2 (0.387)	Struck (0.360)	MTT (0.326)	OAB (0.311)	TLD (0.305)
Occlusion (29)	CDBN-10-2 (0.441)	Struck (0.405)	SCM (0.398)	TLD (0.384)	LSK (0.384)
Out-of-plane rotation (39)	CDBN-10-2 (0.427)	Struck (0.409)	ASLA (0.404)	SCM (0.396)	VTD (0.392)
Out of view (6)	CDBN-10-2 (0.457)	Struck (0.421)	LOT (0.411)	TLD (0.407)	CPF (0.394)
Scale variation (28)	CDBN-10-2 (0.441)	ASLA (0.440)	SCM (0.438)	Struck (0.395)	TLD (0.384)

It implies that our method can construct robust object appearance models by effectively learning and transferring the highly general CDBN features.

5.2.5. Comparison with DLT [45]. To show the advantage of the CDBN-10-2 over other competing trackers based on deep learning, we compare it with the DLT [45]. According the experimental results given in [55], DLT achieves a precision of 0.452 at the threshold of 20 pixels and an AUC of 0.443 on the CVPR 2013 tracking benchmark. Although the DLT has shown good performance in several scenarios, it does not exploit the label information to learn features from the denoising autoencoder and can hardly work well in cluttered background. The proposed CDBN-10-2 outperforms DLT by 23.2% in mean distance precision at the threshold of 20 pixels, while it outperforms it by 9.9% in AUC. This is because the proposed CDBN-10-2 can effectively learn the appearance changes of the target while preserving the ability to discriminate the target from the background via combining the offline and online discriminative learning.

5.3. Efficacy of Different Positive Samples. One big advantage of the proposed CDBN-10-2 lies in that the positive samples are classified into three categories to capture the appearance

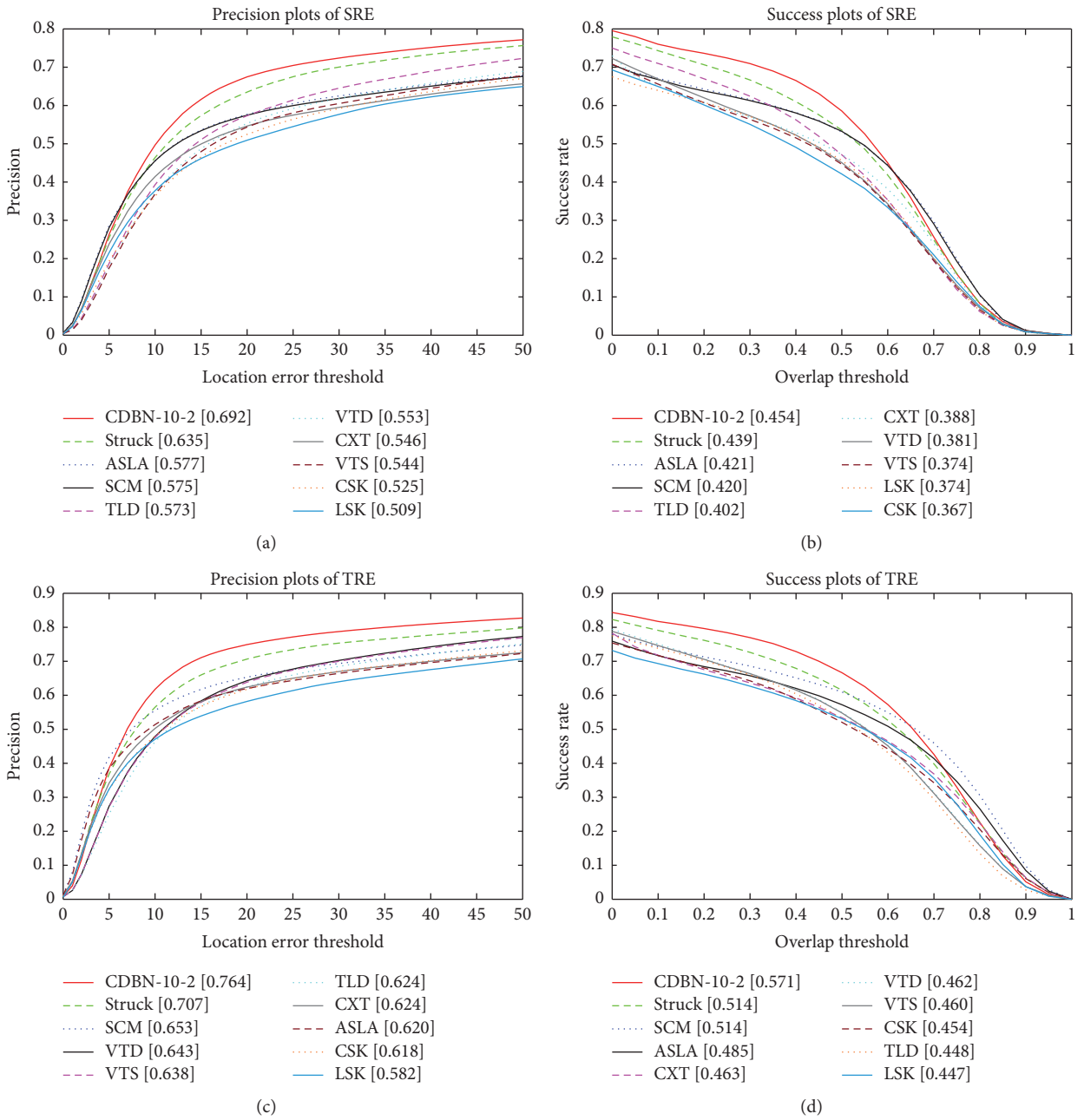


Figure 5: The precision and success plots for TRE and SRE. The proposed CDBN-10-2 (in red) achieves comparable performance in all the evaluations.

variations while alleviating the drifting problem. To verify this advantage, we check the updating process for the positive samples and give several examples in Figure 8. The motorRolling sequence on the first row suffers from large pose and lighting variations. The football sequence on the second row contains a player moving in front of a clutter background. The singer1 sequence on the third row is captured by a PTZ camera and has large illumination changes. The jogging sequence on the fourth row suffers from short-term occlusions, pose, and appearance changes. As shown in Figure 8, it is obvious that the proposed CDBN-10-2 can effectively exploit ground-truth, long-term, and short-term positive samples to incrementally update the CDBN-10-2 to capture object appearance changes while alleviating the drifting problem.

5.4. The Impact of Different Training Data and CDBN Architecture. Since the proposed CDBN-10-2 consists of two CRBM layers followed by one fully connected layer and is pretrained on the CIFAR-10 dataset [53], the following questions arise:

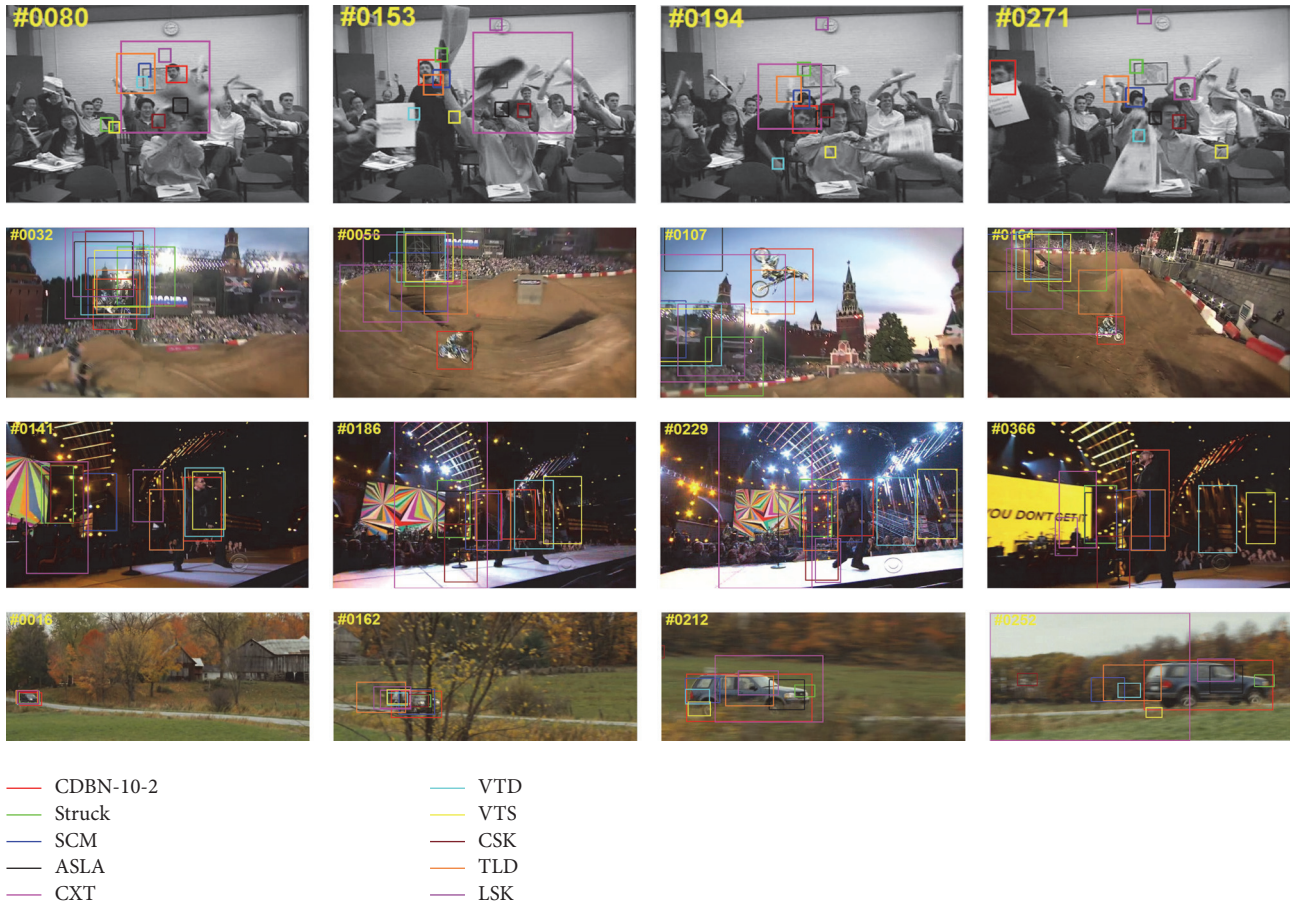


Figure 6: Qualitative comparison on several sequences from [6], that is, the freeman4, motorRolling, singer2, and carScale sequence, respectively.

(1) why the common object recognition dataset is effective for object tracking, even though the dataset does not contain the target objects? (2) Whether the proposed CDBNTracker will continue to improve as data or the number of CRBM layers in CDBN grows? To answer these two questions, we investigate the performance of the proposed CDBNTracker as the amount of training data and the number of CRBM layers in CDBN grow.

Specifically, we first study two simple variations to the CDBN-10-2, namely, CDBN-100-2 and CDBN-tiny-2. They share the same topology of CDBN-10-2 but they are pretrained on either CIFAR-100 or tiny datatset [53]. CIFAR-100 is just like the CIFAR-10, except it has 100 classes containing 600 images each. From the 79 million tiny images, we randomly sample 202,932 images to pretrain the CDBN-tiny-2. Then, we pretrain a CDBNTracker with three CRBM layers followed by one fully connected layer from the CIFAR-10. This version of the CDBNTracker is denoted by CDBN-10-3.

Due to space limitation, we only show the precision and success plots for TRE on the CVPR2013 tracking benchmark in Figure 9. Obviously, the performance of the proposed CDBNTracker continues to improve as data or the number of CRBM layers in CDBN grows. Moreover, although the CDBN is trained offline for other purpose (e.g., object recognition), the proposed CDBNTracker can perform well for the tracking task by using the internal CDBN features as a generic and middle-level image representation. We conjecture that it is because the CDBN features are more effective to represent middle-level concept of target than hand-crafted ones.

5.5. Experimental Results on the Mitocheck Cell Dataset. The qualitative single-cell tracking results of our method on a single-cell from the Mitocheck dataset [54] are shown in Figure 10. Due to space limitations, multiple single-cell tracking results are combined to be shown in Figure 10. It is obviously seen from Figure 10 that the low-quality (low-contrast) images, illumination variations, and large intensity variations challenge the cell tracking methods. Due to the powerful representation learned from multilayer CDBNs with local tied weights to reduce the model complexity under the scarcity of training samples, our method can still provide promising single-cell tracking results.

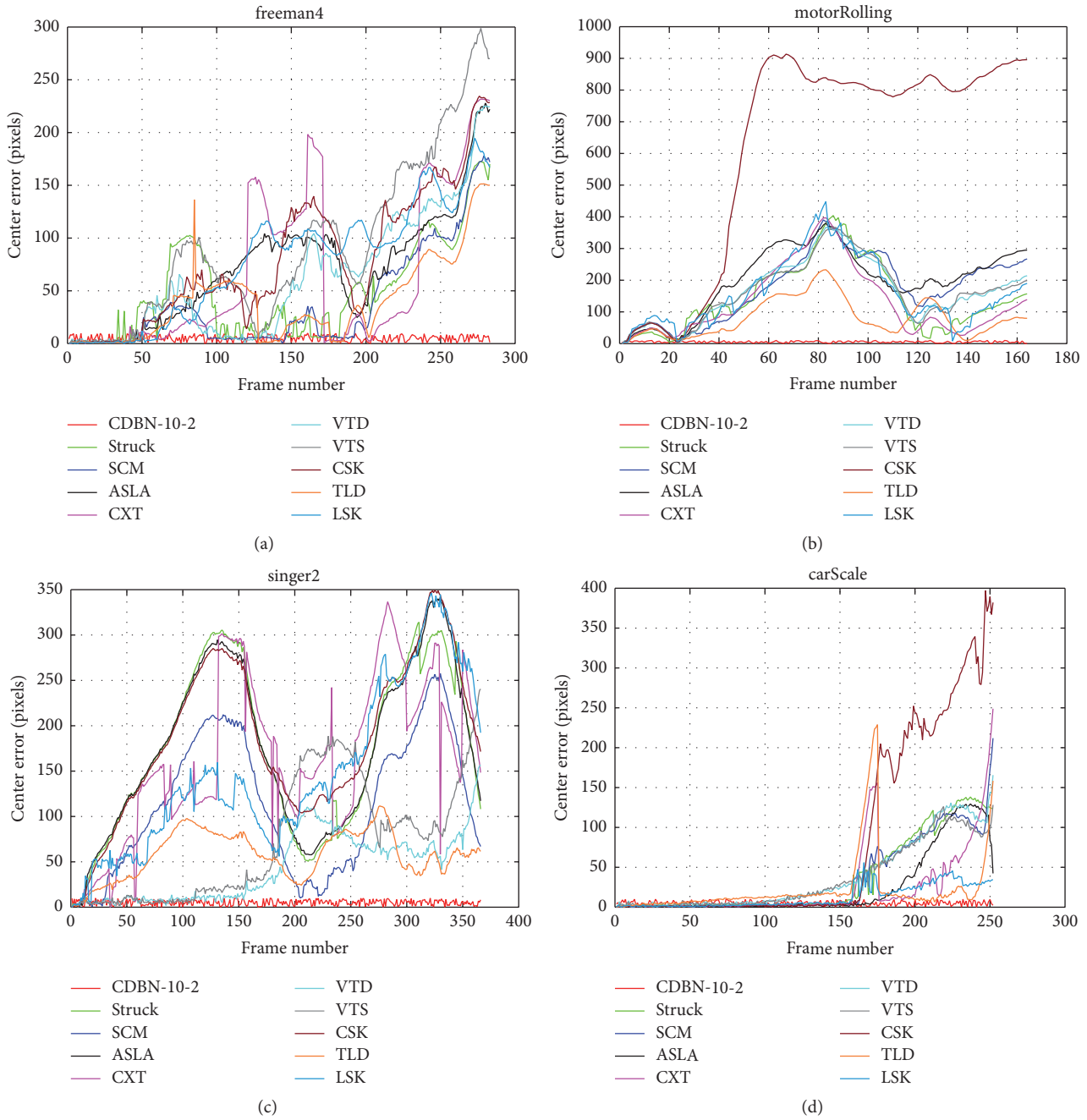


Figure 7: Quantitative comparison on the center distance error per frame for several sequences from [6].

6. Conclusion

In this paper, we have proposed a robust single-cell/object tracking method via learning and transferring CDBN features. The proposed CDBNTracker does not rely on engineered features and automatically learns the most discriminative features in a data-driven way. A simple yet effective method has been used to transfer the generic and midlevel features learned from CDBNs to the single-cell/object tracking task. The drifting problem is alleviated by exploiting ground-truth, long-term, and short-term positive samples. Extensive experiments on the Mitocheck cell dataset and CVPR2013 tracking benchmark have validated the robustness and effectiveness of the proposed CDBNTracker.

Competing Interests

The authors declare that they have no competing interests.

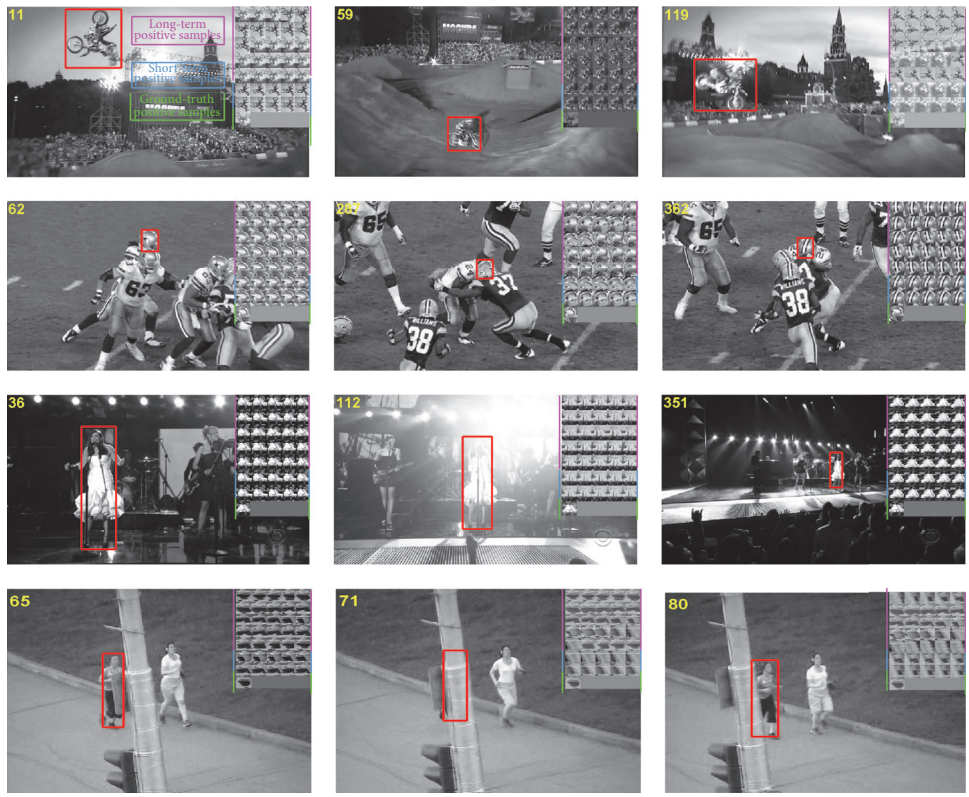


FIGURE 8: Illustration of updating process for the positive samples on several sequences from [6]. Red rectangles represent the bounding boxes of the target objects. The different positive samples are shown in the upper right corner of each image. The first row to the fifth row contain the long-term positive samples which are moderately adaptive. The sixth and seventh row contain the short-term positive samples which are highly adaptive. The last row contains the ground-truth positive samples obtained in the first frame.

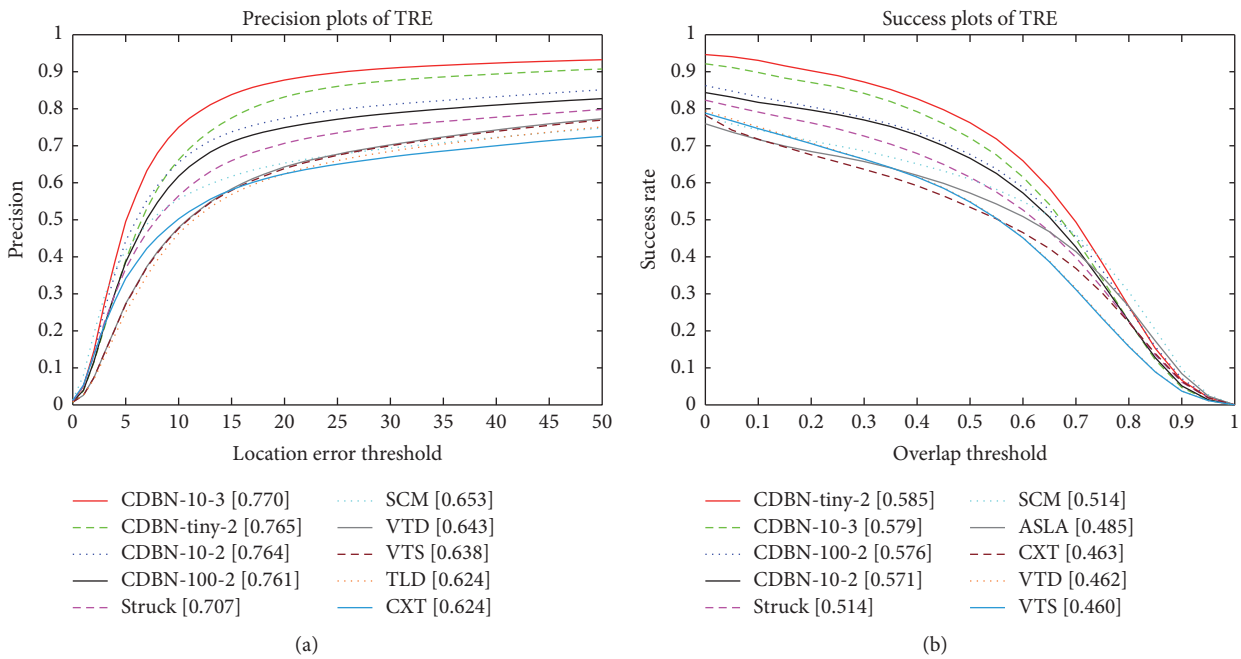


FIGURE 9: We compare the performance of the proposed CDBNTrackers (e.g., CDBN-10-2, CDBN-100-2, CDBN-tiny-2, and CDBN-10-3) as the amount of training data and the number of CRBM layers in CDBN grow.

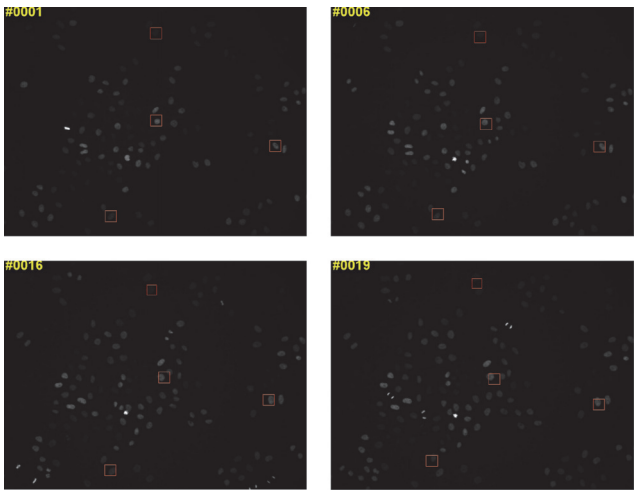


Figure 10: Qualitative comparison on a single-cell from the Mitocheck dataset [54].

Acknowledgments

This work is supported by Natural Science Foundation of China (nos. 61572205, 61572206, 61502182, and 61175121), Natural Science Foundation of Fujian Province (nos. 2015J01257 and 2013J06014), Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (nos. ZQN-PY210 and ZQN-YX108), and 2015 Program for New Century Excellent Talents in Fujian Province University.

References

[1] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen, "Tracking in cell and developmental biology," *Seminars in Cell and Developmental Biology*, vol. 20, no. 8, pp. 894–902, 2009.

[2] T. Kanade, Z. Yin, R. Bise et al., "Cell image analysis: algorithms, system and applications," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV '11)*, Kona, Hawaii, USA, 2011.

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[4] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[5] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. van den Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 1–58, 2013.

[6] Y. Wu, J. W. Lim, and M. H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '13)*, 2013.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[8] F. Li, X. Zhou, J. Ma, and S. T. C. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 96–105, 2010.

[9] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[10] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[11] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 260–267, IEEE, New York, NY, USA, June 2006.

[12] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, November 2011.

[13] R. Yao, Q. F. Shi, C. H. Shen, Y. N. Zhang, and A. Van Den Hengel, "Part-based visual tracking with online latent structural learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2363–2370, Portland, Ore, USA, June 2013.

[14] V. Takala and M. Pietikainen, "Multi-object tracking using color, texture and motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, Minneapolis, Minn, USA, June 2007.

[15] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 81–88, Barcelona, Spain, November 2011.

[16] Y. Lu, T. Wu, and S.-C. Zhu, "Online object tracking, learning, and parsing with and-or graphs," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 3462–3469, June 2014.

[17] J. Fan, X. Shen, and Y. Wu, "Scribble tracker: a matting-based approach for robust tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1633–1644, 2012.

[18] X. Lou, M. Schiegg, and F. A. Hamprecht, "Active structured learning for cell tracking: algorithm, framework, and usability," *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, pp. 849–860, 2014.

[19] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, 2014.

[20] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, Marseille, France, 2008.

[21] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: parallel robust online simple tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 723–730, June 2010.

[22] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[23] L. Zhang and L. Van Der Maaten, "Preserving structure in model-free tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756–769, 2014.

[24] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[25] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, Piscataway, NJ, USA, September-October 2009.

[26] K. H. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part III*, vol. 7574 of *Lecture Notes in Computer Science*, pp. 864–877, Springer, Berlin, Germany, 2012.

[27] D. Padfield, J. Rittscher, and B. Roysam, "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis," *Medical Image Analysis*, vol. 15, no. 4, pp. 650–668, 2011.

[28] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[29] S. Duffner and C. Garcia, "PixelTrack: a fast adaptive algorithm for tracking non-rigid objects," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2480–2487, Sydney, Australia, December 2013.

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, December 2012.

[33] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[34] J. Donahue, Y. Jia, O. Vinyals et al., "DeCAF: a deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 988–996, Beijing, China, June 2014.

[35] Q. Zou, X.-B. Li, W.-R. Jiang, Z.-Y. Lin, G.-L. Li, and K. Chen, "Survey of MapReduce frame operation in bioinformatics," *Briefings in Bioinformatics*, vol. 15, no. 4, pp. 637–647, 2014.

[36] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[37] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, June 2010.

[38] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[39] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[40] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang, "Transferring visual prior for online object tracking," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3296–3305, 2012.

[41] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8691 of *Lecture Notes in Computer Science*, pp. 188–203, 2014.

[42] G. Carneiro and J. C. Nascimento, "Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2592–2607, 2013.

[43] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pp. 609–616, Montreal, Canada, 2009.

[44] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[45] N. Y. Wang and D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, 2013.

[46] J. L. Fan, W. Xu, Y. Wu, and Y. H. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[47] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the International Conference on Machine Learning*, 2015.

[48] C. Ma, J. B. Huang, X. K. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.

[49] H. S. Nam and B. Y. Han, "Learning multi-domain convolutional neural networks for visual tracking," http://arxiv.org/abs/1510.07945.

[50] Y. Chen, X. N. Yang, B. N. Zhong, S. N. Pan, D. S. Chen, and H. Z. Zhang, "CNNTracker: online discriminative object tracking via deep convolutional neural network," *Applied Soft Computing*, vol. 38, pp. 1088–1098, 2016.

[51] L. J. Wang, W. L. Ouyang, X. G. Wang, and H. C. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3119–3127, Santiago, Chile, December 2015.

[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," https://arxiv.org/abs/1409.1556.

[53] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[54] http://www.mitocheck.org/cgi-bin/mtc?action=show_movie;query=243867.

[55] N. Wang and D.-Y. Yeung, "Ensemble-based tracking: aggregating crowdsourced structured time series data," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 2807–2817, Beijing, China, June 2014.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 7639397, 9 pages
http://dx.doi.org/10.1155/2016/7639397

Research Article

An Approach for Predicting Essential Genes Using Multiple Homology Mapping and Machine Learning Algorithms

Hong-Li Hua,[1,2] Fa-Zhan Zhang,[1,2] Abraham Alemayehu Labena,[1,2] Chuan Dong,[1,2] Yan-Ting Jin,[1,2] and Feng-Biao Guo[1,2]

[1]*Center of Bioinformatics, School of Life Science and Technology, Key Laboratory for Neuroinformation of the Ministry of Education, University of Electronic Science and Technology of China, Chengdu, China*
[2]*Center of Information in Biomedicine, University of Electronic Science and Technology of China, Chengdu, China*

Correspondence should be addressed to Feng-Biao Guo; fbguo@uestc.edu.cn

Received 30 May 2016; Revised 25 July 2016; Accepted 4 August 2016

Academic Editor: Yungang Xu

Investigation of essential genes is significant to comprehend the minimal gene sets of cell and discover potential drug targets. In this study, a novel approach based on multiple homology mapping and machine learning method was introduced to predict essential genes. We focused on 25 bacteria which have characterized essential genes. The predictions yielded the highest area under receiver operating characteristic (ROC) curve (AUC) of 0.9716 through tenfold cross-validation test. Proper features were utilized to construct models to make predictions in distantly related bacteria. The accuracy of predictions was evaluated via the consistency of predictions and known essential genes of target species. The highest AUC of 0.9552 and average AUC of 0.8314 were achieved when making predictions across organisms. An independent dataset from *Synechococcus elongatus*, which was released recently, was obtained for further assessment of the performance of our model. The AUC score of predictions is 0.7855, which is higher than other methods. This research presents that features obtained by homology mapping uniquely can achieve quite great or even better results than those integrated features. Meanwhile, the work indicates that machine learning-based method can assign more efficient weight coefficients than using empirical formula based on biological knowledge.

1. Introduction

Essential genes are those genes which are indispensable for the basic activities of organisms under certain growth conditions [1]. The proteins coded by essential genes are considered to carry out the fundamental biological functions. Therefore, these essential genes are regarded as the basis of life [2]. Knowing more about necessity of genes can help researchers find out the existence form of microbes [3], construct the minimal gene subset [4], discover potential drug targets, and design reasonable and effective drugs to resist microbial pathogens [5]. In addition, these genes are more inclined to be related to basic cellular processes such as duplication and translation, which would lead essential genes to be more stringent than nonessential genes when negative (purifying) selection occurs [6, 7]. Because of their tremendous functions in cells, the research of essential genes has become hotspot in bioinformatics and genomics. A series of experimental approaches such as single gene knockouts [8], conditional knockouts [9], RNA interference [10], and transposon mutagenesis [11] have been provided to identify microbic essential genes. While experimental techniques may be reliable, these methods have significant shortcomings, such as high cost and long duration.

As an alternative way, computational methods do not have the above-mentioned drawbacks. Hence, some researchers attempt to use computational techniques combining with biological characteristics to identify essential genes. To some extent, some of these methods have obtained satisfactory results. Deng et al. trained classifier on the basis of several biological features including intrinsic and context-dependent genomic features. The results of their method yielded AUC scores between 0.86 and 0.93 through tenfold cross-validation test in the same organism and 0.69 to 0.89 for cross organism predictions [12]. Song et al. used Z-curve and some other features which derived from sequences

combining with their linear method to predict essential genes. They acquired AUC scores between 0.8042 and 0.9319 in 12 organisms [13]. These biological features often used can be summarized as three types: intrinsic genomic features like GC content, derived features from sequence like codon adaption index, experimental data like gene expression profile, as well as other features like gene ontology and functional gene network [14–18]. Although these features are associated with gene essentiality, the majority of them cannot be collected or available in most of microbes, nor does every feature have high predictive power in identification of essential genes. Besides, these features may increase biological redundancy. Therefore, most of the methods based on various biological characteristics just could be developed in merely a handful of species. However, essential genes tend to be conserved during the long-term evolution [19]; thus sequence alignment is of great significance for molecular function prediction [20]. Considering these factors, our group previously developed a universal tool named Geptop (gene essentiality prediction tool based on orthology and phylogeny) to predict gene essentiality [21]. This approach uses reciprocal best hit (RBH) method to obtain the results of homology mapping and considers the distance of phylogeny as the weight of orthology variables. Through a series of tests, Geptop, a method designed only based on biological knowledge, has obtained quite better results than those based on integrated features.

In this study, we attempted to investigate the optimized weight and find whether it can further improve the predictions or not. In recent years, the machine learning-based methods have shown significant performance in many prediction researches [22]. Therefore, we put forward a new approach to identify essential genes based on multiple homology mapping and machine learning technique. For a given organism, the greatest weight is the evolutionary distance between it and its closest related organism in Geptop. However, in our method, it was measured by the feature with the best ability to distinguish positive and negative sample sets.

2. Materials and Methods

2.1. Data Sources. Annotations of essential genes were downloaded from the latest version of Database of Essential Genes (DEG) at http://tubic.tju.edu.cn/deg/. 39 bacterial essential gene sets are included in DEG database, but not all of them are reliable because of the limitations of wet-lab technologies. Additionally, an organism may have different batches of data accompanying with different accuracy, and some genomes contain many conditional essential genes which are specific in these organisms. Therefore, we excluded those inappropriate datasets and finally chose 25 essential gene sets as positive datasets. Meanwhile, we downloaded another annotation data of *Escherichia coli* K_12 *(E. coli)* from Profiling of *E. coli* Chromosome (PEC) [23] for extra study. Those genes, which cannot be found in essential gene sets, are regarded as nonessential genes and are used to construct negative datasets for each organism. Therefore, each gene of the target species was assigned a Boolean value to label the essentiality (essential: +1; nonessential: −1). The complete protein coding sequences with fasta format of 25 organisms were obtained from NCBI Genomes. We listed all species used in this work in Table 1. In order to describe these species conveniently, we gave an abbreviated name for each of them (Table 1).

2.2. Homology Mapping. Essential genes tend to be more conserved than nonessential genes in the process of long-term evolutionary. Hence, they should be kept in most of the bacteria [24]. This property of essential genes constitutes the basis of our method. In our work, we used the method of reciprocal best hit (RBH) to identify orthologs between the selected organisms through pair-wise comparison. For two given organisms, one was used as the query species Q and the other was used as the referential species R. Firstly, we queried an CDS_i (coding DNA sequence) of Q against all CDS_s in R by Blastp program with a default E-value threshold of 10 and yielded a set of hits {M}. Then, we queried CDS_j with the lowest E-value in {M} against all CDS_s in Q by the same way and yielded a set of hits {N}. A pair of proteins (CDS_i, CDS_j) are considered orthologs if we queried CDS_i with the lowest E-value in {N}. Therefore, CDS_j is assumed as orthologous-essential gene if the referential gene is annotated as essential by experimental methods in species R. And the value of sample CDS_i for feature R is marked as 1 if CDS_i has orthologous-essential gene in species R. Otherwise, it is marked as 0. Finally, a gene can be represented by a series of binary values. Specially, for training sets, the feature names are described by the names of referential species, and binary vectors are values of these features.

2.3. Method of Geptop. Geptop calculates gene essentiality through combining the features which computed by homology mapping with corresponding evolutionary distance between query species and its referential species. The evolutionary distance is calculated by composition vector (CV) [25]; it is employed as the weight of orthologous-essential gene. For a given genome, Geptop took multiple homology mapping to obtain the orthologous-essential genes of different referential species. Then, it computed the CV distances between the query genome and the other referential ones. The gene essentiality of the given organism is calculated by the following equation:

$$S_i = 1 - \left(\sum_{j=1}^{N} E_{ij} \times D_j \right)^{1/N}, \quad (j = 1, 2, 3, \ldots, N), \tag{1}$$

where S_i represents the essentiality score of ith gene of query species, E_{ij} represents the essentiality of the optimal orthologous gene of ith query gene in jth referential species, D_j is the distance between the query species and the jth referential species, and N is the total number of referential species. Thus, Geptop method can decide whether a gene is essential or not according to the essentiality score.

2.4. Method Based on Sequence Feature and Machine Learning Algorithm

2.4.1. Support Vector Machine. Support vector machine (SVM) [26], an efficient machine learning method, has been

Table 1: Bacteria used in this work.

Species	Abbreviation	Number of essential genes	Number of total genes
Acinetobacter ADP1	*ACA*	499	3307
Bacillus subtilis 168	*BAS*	271	4175
Bacteroides thetaiotaomicron VPI 5482	*BAT*	325	4778
Burkholderia thailandensis E264	*BUT*	406	5632
Caulobacter crescentus NA1000	*CAC*	480	3885
Campylobacter jejuni NCTC 11168 ATCC 700819	*CAJ*	222	1572
Escherichia coli K-12 MG1655	*ESC*	296	4140
Escherichia coli K-12 in PEC database	*ESC_PEC*	287	4146
Francisella novicida U112	*FRN*	390	1719
Mycobacterium tuberculosis H37Rv	*MYT*	611	3906
Mycoplasma genitalium G37	*MYG*	378	475
Mycoplasma pulmonis UAB CTIP	*MYP*	310	782
Porphyromonas gingivalis ATCC 33277	*POG*	463	2089
Pseudomonas aeruginosa UCBPP PA14	*PSA*	335	5892
Salmonella enterica serovar Typhimurium 14028S	*SA14028S*	105	5315
Salmonella enterica serovar Typhimurium LT2	*SAL*	230	4451
Salmonella enterica serovar Typhimurium SL1344	*SAS*	353	4446
Salmonella enterica serovar Typhi Ty2	*SAT*	358	4352
Shewanella oneidensis MR 1	*SHO*	402	4065
Sphingomonas wittichii RW1	*SPW*	535	4850
Staphylococcus aureus N315	*STN315*	302	2582
Staphylococcus aureus NCTC 8325	*STNCTC*	346	2767
Streptococcus pneumonia TIGR4	*STT*	111	2105
Streptococcus pneumonia R6	*STR*	127	1814
Streptococcus sanguinis SK36	*STS*	218	2270
Vibrio cholerae O1 biovar El Tor N16961	*VIC*	591	3503

The number of essential genes and total genes are counted after filtering unmatched data.

widely used in classification and pattern recognition. It adopts the principle of structural risk minimization and belongs to supervised learning method. SVM maps features into a high-dimensional feature space by kernel function. In the high-dimensional space, the samples with different attributions can be separated easily. In the present work, we adopted LibSVM [27] to perform SVM algorithm with RBF kernel function. It gave evaluation index for each feature that differed from the weight given by Geptop method.

2.4.2. Feature Selection. In order to measure the contribution of each feature in a test process, we utilized F-score algorithm [28] to estimate the importance of them. F-score is a simple and quite effective arithmetic to discriminate two sets of real data. The larger the score is, the more significant contribution the feature makes. For training vectors $x_k (k = 1, 2, \ldots, p)$ the F-score is defined as followings:

$$F(i) = \frac{\left(\overline{x}_i^{(+)} - \overline{x}_i\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x}_i\right)^2}{\left(1/\left(n_+ - 1\right)\right) \sum_{k=1}^{n_+} \left(x_{k,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + \left(1/\left(n_- - 1\right)\right) \sum_{k=1}^{n_-} \left(x_{k,i}^{(-)} - \overline{x}_i^{(-)}\right)^2}, \quad (2)$$

where n_+ and n_- are the number of positive and negative samples, respectively; $\overline{x}_i$, $\overline{x}_i^{(+)}$, and $\overline{x}_i^{(-)}$ are the mean of the ith feature of the total, positive and negative samples, respectively; $x_{k,i}^{(+)}$ is the ith feature of the kth positive sample; $x_{k,i}^{(-)}$ is the ith feature of the kth negative sample.

2.4.3. Classifier Design and Performance Evaluation. For a species under test, the homology mapping was implemented between the query species and other 24 organisms, and then 24 features were obtained to train the classifier. We used the classic machine learning method SVM to train the model and predict essential genes. Gaussian kernel function

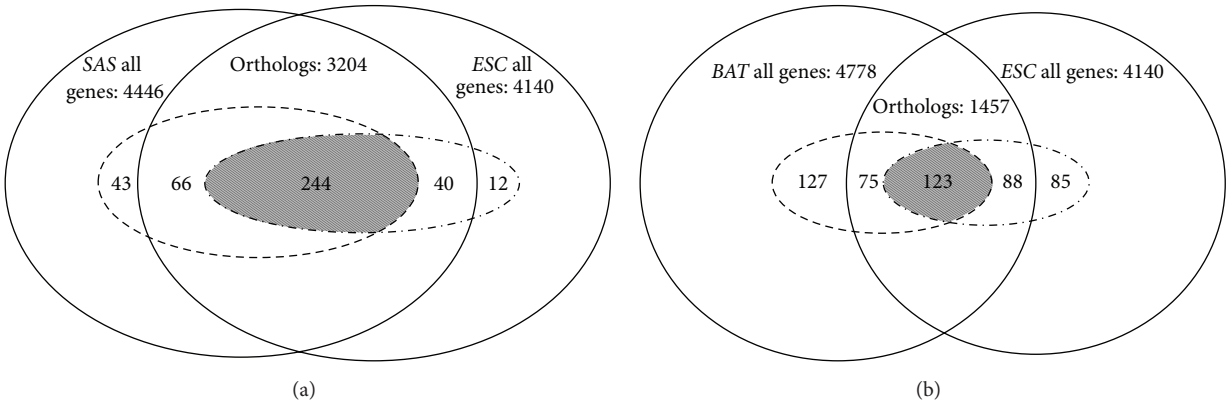


Figure 1: Comparison of the number of conserved genes and essential genes between two organisms. (a) We compared the difference between *SAS* and *ESC*, two relatively closely related organisms. They shared 3204 orthologous genes and 244 common essential genes. The broken circle represents 353 *SAS* essential genes, and the dash dotted line circle represents 296 *ESC* essential genes. For *ESC*, there are 310 orthologous-essential genes in *SAS*. (b) We compared the difference between *BAT* and *ESC*, two relatively distantly related organisms. They shared 1457 orthologous genes and 123 common essential genes. The broken circle represents 325 *BAT* essential genes, and the dash dotted line circle represents 296 *ESC* essential genes. For *ESC*, there are 198 orthologous-essential genes in *BAT*. Obviously, the closer species may have more orthologous sequence and more common essential genes with the target species than the distant one.

was selected to project the original features into a high-dimensional space. Gridding search method was adopted to search the best penalty parameter C and γ. Cross-validation and receiver operating characteristic (ROC) curve is the usual performance evaluation method for predictions [29]. Therefore, we used the area under ROC curve (AUC) of tenfold cross-validation to evaluate the performance of our classifier. For 10-fold cross-validation, the training data were randomly divided into 10 equal parts. Nine parts were used to train the classifiers and the remaining part was used for testing. This process was repeated until each part was taken as test set. For prediction in cross organisms, we chose the feature sets of the closest organism or them of the greatest contributed feature/organism to train the model and used the same characteristic variables of test organism to make prediction. The predictions were compared with the known gene essentialities which have been determined by experimental method.

3. Results

3.1. Evolutionary Distance and Orthology in Cross Species. If two organisms have closer evolutionary distance, they may share more orthologous genes or more common essential genes, relatively. We used an online web server CVTree [30] to establish phylogenetic tree for the selected 25 species. The phylum each organism belongs to and the distance relationships among them were illustrated in Supplementary Data Figure 1 in Supplementary Material available online at http://dx.doi.org/10.1155/2016/7639397. We discovered that only one organism belongs to *Actinobacteria*, other 24 organisms are from 4 different phyla, and each phylum has more than one organism.

Essential genes are always inclined to be conserved because of their important functions, while conserved genes across species are not necessarily essential. We compared the number of conserved genes and essential genes between two organisms to illustrate the orthologous relationships across species (Figure 1). We analyzed these relationships in two groups. One pair contains relatively closely related organisms: *ESC* and *SAS* (Figure 1(a)), and the other pair contains relatively distantly related organisms: *ESC* and *BAT* (Figure 1(b)). The evolutionary distances of *SAS-ESC* and *BAT-ESC* are 0.3273 and 0.4976, respectively. There are 3204 orthologous genes between *ESC* and *SAS*, in which there are 244 common essential genes, accounting for around 77.39% and 5.89% in the total number of *ESC* genes, respectively. In the other group, 1457 genes are orthologous between *ESC* and *BAT*, in which there are 123 common essential genes, accounting for only 35.19% and 2.97% in the total number of *ESC* genes, respectively. These two results exemplified that the closer the evolutionary distance between species is, the more orthologous genes or common essential genes they would share. Meanwhile, an organism may share different essential genes with different organisms. Hence, we need to use multiple genomes to implement homology mapping, which we called multiple homology mapping.

3.2. Classifier Training towards 25 Genomes and 10-Fold Cross-Validations of the Classifier. We gave the flowchart to display how this work was implemented in Figure 2. In this study, features we used were derived from protein sequences via homology mapping, which are easier to be acquired compared with other various biological features. Totally, 24 features were used as input variables for SVM classifier. The input data contained 24 features and the class labels for the species under test. Each genome was taken as the test species, and each of their AUC score of 10-fold cross-validations was acquired. All of their results of prediction yielded AUC scores between 0.5700 and 0.9716 (Figure 3) and accuracy scores between 0.7980 and 0.9805 (Supplementary Data Table 1). In addition, the AUC score of *ESC_PEC (E. coli)* whose data were

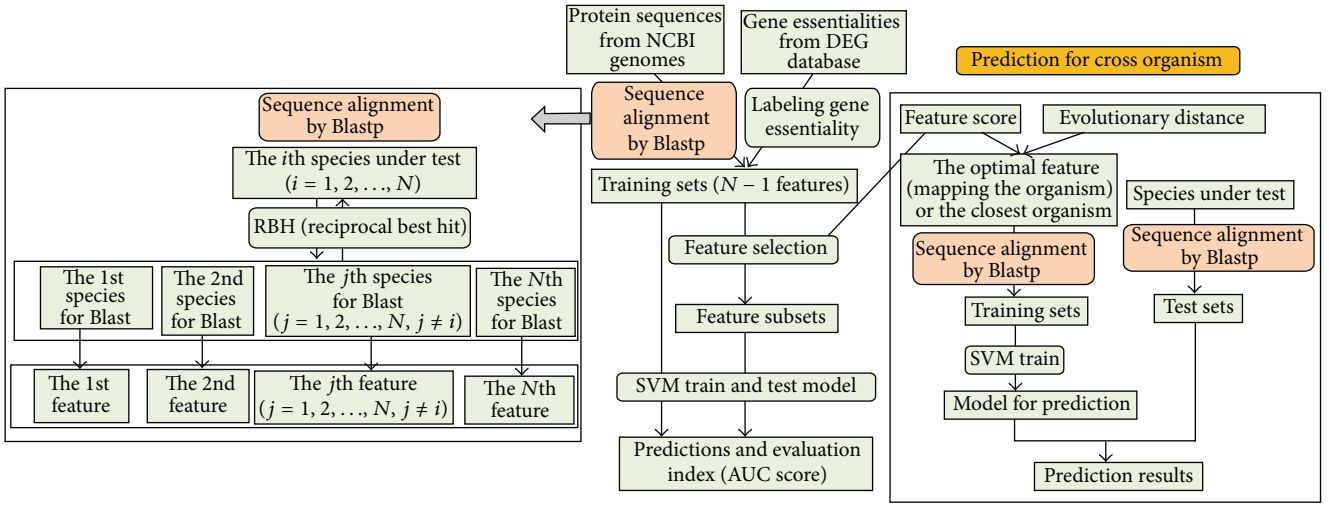


Figure 2: The flowchart for obtaining training sets by multiple homology mapping and training the model to predict essential genes. For a species under test, it was used for sequence alignment towards other 24 species, respectively, and each result was used as a training feature. The training sets obtained from multiple sequence alignment were used to train and test the prediction model by SVM. Meanwhile, we used the F-score to evaluate the discriminative capability of each feature. The optimal feature subsets were selected to train and test the model. Tenfold cross-validation was utilized to assess the performance of the classifier. For predicting essential genes in cross organisms, the feature sets of the closest organism or those of the organism/feature which has the biggest F-score for the target species were selected as the training sets to train model, and then this model was used to predict essential genes in target species.

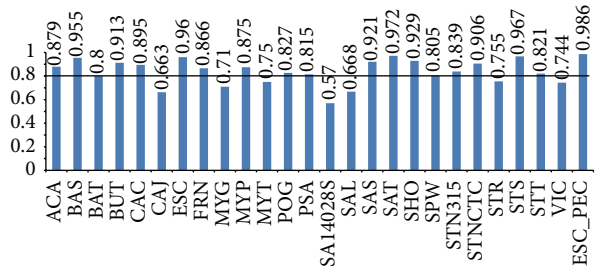


Figure 3: 26 AUC scores of 10-fold cross-validation within 25 bacteria as well as *ESC_PEC*, respectively. The last AUC score belongs to *ESC* whose data are obtained from PEC database. More than 70% of the results exceed the AUC score of 0.80, and 9 organisms' results of prediction yielded AUC scores more than 0.90.

obtained from PEC database is 0.9864 by this classifier. Previously, Deng et al. [12] got AUC score of 0.93 by 10-fold cross-validation for the same dataset through combining 4 machine learning methods and 13 features including codon bias index, Aromaticity, and Paralogy. Absolutely, our method achieved better performance than theirs. In addition, more than 70% of the bacteria exceed AUC score of 0.80, and merely 12% of all are less than 0.70. These results demonstrate that our classifiers have quite great performance.

3.3. Predictions after Feature Selection. In the above work, all features obtained by homology mapping were utilized to train the classifiers, but not every feature has high predictive power for identification of essential genes. Therefore, ranking the features in order and filtering out the useless features played an important role in prediction [29]. We used the method of feature selection to choose appropriate feature subsets to reappraise the performance of classifiers. Based on F-score algorithm [28], we got a vector composed of 24 feature scores in descending order for an organism under test. Another feature selection method named DX score [31] was also applied to measure feature score, and the same order of features were acquired. It indicated that the features' order we obtained was reliable. Feature subsets were constructed through appending a feature one by one in accordance with the descending order, and each feature subset was utilized to train and test the model. Finally, the optimal feature subset and its predictions would be chosen on the basis of AUC score. After selecting optimal feature subsets, most of predictions have corresponding improvement. The AUC scores of 8 species were improved by more than 1%, and the average AUC score was improved around 1% among 25 species comparison with the results of using all features. In addition, contribution score of each feature was obtained by F-score. For each organism under test, we analyzed the correlation between feature scores and evolutionary distances through Pearson correlation analysis (Table 2). We discovered that 22 species among all presented negative correlations, in which 15 results presented significant negative correlations. That is to say the importance of features evaluated by machine learning method is consistent with the evolutionary distance between them in general.

3.4. Predicting Essential Genes across Organisms. It is necessary to use the suitable model to predict essential genes across distantly related bacteria. The relatively closely related organisms may have similar patterns in developing model for predicting gene essentiality when using orthologs to some extent. For a species with unknown gene essentiality,

Table 2: Correlations between evolutionary distances and feature scores for each target organism.

Organisms	Correlations	P value
ACA*	−0.45204	0.0266
BAS**	−0.37043	0.0075
BAT*	−0.50001	0.0128
BUT*	−0.41124	0.0459
CAC*	−0.41482	0.0439
CAJ	−0.23786	0.2631
ESC*	−0.50218	0.0120
ESC_PEC**	−0.52353	0.0087
FRN	−0.39292	0.0575
MYT	−0.35883	0.0851
MYG*	−0.49728	0.0134
MYP**	−0.54766	0.0056
POG*	−0.46123	0.0233
PSA**	−0.60836	0.0016
SA14028S**	−0.60533	0.0017
SAL	−0.28669	0.1744
SAS	−0.24910	0.2405
SAT	−0.31248	0.1371
SHO*	−0.50456	0.0119
SPW**	−0.65577	0.0005
STN315	−0.11162	0.6036
STNCTC	0.03883	0.8570
STR	0.11619	0.5887
STS	0.24868	0.2413
STT	0.19591	0.3589
VIC*	−0.50718	0.0114

$*$ represents that the correlation is significant at the 0.05 level; $**$ represents that the correlation is significant at the 0.01 level.

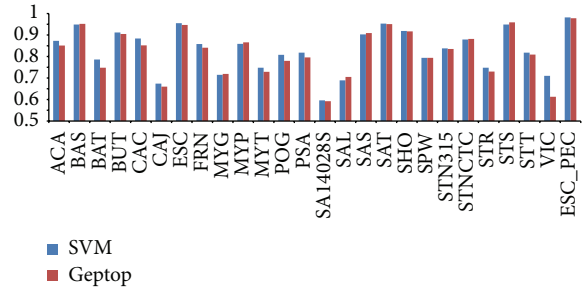


Figure 4: Comparison AUC scores of interspecies prediction for 25 bacteria between SVM and Geptop. The last AUC score belongs to *ESC* whose data are obtained from PEC database. The vertical axis, in the range from 0.5 to 1, represents AUC scores. More than 65% of the results exceed the AUC score of 0.80, and 8 organisms' results of prediction yielded AUC scores more than 0.90. For 26 genomes including *ESC_PEC*, 18 of all are better than Geptop.

using the features of closest organism as training set to train model is available. However, the closest species may not be the most important contributor in machine learning methods. Besides, if the data quality of this species is poor, it could cause bad effects for the classifier and the classifier could not give the best predictions. Therefore, we chose the characteristic set of closest species and the characteristic set of the greatest contributor to train the model, respectively. The better outputs of these diverse training set were chosen as the results of prediction. For example, *Salmonella enterica* serovar Typhimurium LT2 *(SAL)* is the closest related bacterium for *ESC*, but the classifier built by its features made a low AUC score of 0.7894. This may be related to the low data quality of *SAL* itself (Supplementary Data Table 2). Nevertheless, we chose the features of *Salmonella enterica* serovar Typhi SL1344 *(SAS)* as training set to train model for predicting essential genes of *ESC*, because feature SAS has maximum F-score among all features of *ESC*. AUC score of 0.9552 and precision (or PPV) of 0.7330 were acquired, and the classifier identified 258 true essential genes from its 352 positive outputs. These two values are greater than the AUC score of 0.9470 and precision (or PPV) of 0.6574 through Geptop method, which identified 236 true essential genes from its 359 positive outputs. We applied this method to implement interspecies prediction among other 24 organisms. As a result, AUC scores between 0.5957 and 0.9552 were obtained for 25 bacteria except *ESC_PEC* (Figure 4, Supplementary Data Table 3). For 26 genomes including *ESC_PEC*, 18 results of prediction are better than Geptop. And the average AUC was improved by 1.14% compared with Geptop.

A newly determined essential gene set of *Synechococcus elongatus* PCC 7942 *(SYE)* by experimental technology are acquirable for independent testing [32]. Totally, 674 genes are annotated as essential in its genome. This bacterium belongs to *Cyanophyta*, a quite different phylum compared with the bacteria already used in this study. We collected this dataset to further evaluate the performance of our classifier. The evolutionary distances between *SYE* and the existing organisms were calculated by composition vector (CV) method. The nearest species was determined as *Caulobacter crescentus* NA1000 *(CAC)*. We chose the features of *CAC* to train a classifier and obtained values of the same features of *SYE* as test set through homology mapping between it and other 24 species except *CAC*. Through SVM, we achieved AUC score of 0.7855 (Figure 5) and precision (or PPV) of 0.8105. In order to assess performance of machine learning method, Geptop method was implemented to predict essential genes on *SYE*, and then it obtained AUC score of 0.7578 and precision (or PPV) of 0.8484. Although the precision of Geptop is higher than that of SVM, SVM method identified 325 true essential genes from its 401 positive outputs, which are more than 263 true essential genes identified by Geptop.

4. Discussion

We make predictions based on gene essentialities which have been determined by experimental technology. There is no doubt that the results of prediction would be influenced by quality of experimental data. For example, if insertion was avoided accidentally, transposon mutagenesis technique would be likely to mislabel short genes [33]. Therefore, the reason why those three species *(CAJ, SAL*, and *SA14028S)*

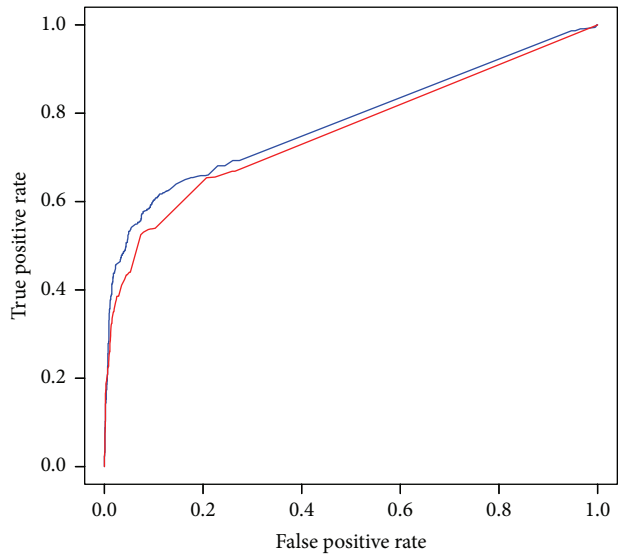


Figure 5: Comparison the ROC curve of Geptop and SVM for *SYE*. The blue curve represents the results obtained through SVM, and the area under it is 0.7855. The red curve represents the results obtained through Geptop, and the area under it is 0.7578.

have AUC lower than 0.70 may be that a mass of essential genes were mislabeled when they were identified by experimental techniques. This has been discussed in Geptop. Additionally, according to the fact that few contributions *SA14028S* provided for almost all organisms and few contributions these features provided for it, we can further presume that the data of this species have low quality.

Geptop gave weights for features only based on evolutionary distance, which would ignore the data quality. Our machine learning method measures the features from an integrated view. In order to investigate the reason why our method performs better than Geptop, we ranked the features based on theirs abilities of differentiate between positive and negative sample sets, which were measured by F-score in our method, and we also ranked them based on weights which were measured by evolutionary distance in Geptop. The rank changes of features from Geptop to SVM were calculated. We analyzed the relationships between AUC scores of 10-fold cross-validation and rank changes using Pearson correlation method (Table 3). We find that 23 among 25 results present significant positive correlations. It indicates that rank changes of features from Geptop to SVM are consistent with AUC values. Taking *ACA* as example (Supplementary Data Table 4), the most important feature in Geptop is ranked as 11th in SVM. Specially, the 6th important feature in Geptop is ranked as 24th in SVM. In addition, this feature has no contribution for predicting. In truth, the 24th feature for *ACA* in SVM is *SA14028S*, which has been discussed that it has low data quality in the above paragraph. Furthermore, the important features in Geptop with relatively high AUC like *ESC* and *SAS* have no or few rank changes. In other words, our method can take the predictive power of every feature into full account. Simultaneously, as a machine learning method, SVM has its inherent advantages. For instance, the final decision function is dominantly determined by a few support vectors, which not only can help us acquire key samples, but also has good robustness. Thus, our method can achieve better predictions than Geptop; the latter makes prediction completely depending on biological knowledge.

Table 3: Correlations between rank changes and AUC scores of 10-fold cross-validation.

Organisms	Correlations	P value
*ACA***	0.66774	0.0005
*BAS**	0.50475	0.0140
*BAT***	0.61735	0.0017
*BUT***	0.64683	0.0009
*CAC***	0.64730	0.0008
*CAJ***	0.52945	0.0094
*ESC***	0.69090	0.0003
*FRN***	0.70211	0.0002
*MYT***	0.58115	0.0036
*MYG***	0.58017	0.0037
*MYP***	0.67868	0.0004
*POG***	0.58558	0.0033
*PSA***	0.62930	0.0013
SA14028S	0.36461	0.0872
*SAL***	0.66214	0.0006
*SAS***	0.69220	0.0003
*SAT***	0.70091	0.0002
*SHO***	0.73831	$5.77E-05$
*SPW***	0.66206	0.0006
*STAN315**	0.50613	0.0137
*STANCTC***	0.54941	0.0066
STR	0.37110	0.0813
*STS***	0.58267	0.0035
*STT***	0.70091	0.0002
*VIC***	0.65402	0.0007

$*$ represents that the correlation is significant at the 0.05 level; $**$ represents that the correlation is significant at the 0.01 level.

5. Conclusion

Our classifier is designed based on RBH method, which can reflect substantive characteristics of orthologous sequence. Although the homology mapping method may ignore the species-specific essential genes, it still can identify reasonable number of essential genes. Superiority of multiple homology mapping has been presented in Geptop, which acquired satisfactory results in predicting essential genes. Besides, these features could be extracted for almost all sequenced bacterial genomes. We utilized this method incorporating with SVM to train classifier and predict essential genes, and then the classifier achieved better performance than Geptop. It probably gives the credit to the fact that our method can measure predictive ability of each feature through machine learning method, which differs from Geptop that only considers evolutionary distance; thus our method can train the better model. Geptop has been designed as a web server; our

method also has potential to be developed as a tool to provide service for users.

In conclusion, through multiple homology mapping and machine learning method, we provide a significant alternative method to predict essential genes. The results of prediction yield higher AUC scores than those integrated approaches as well as Geptop method. Simultaneously, this work reveals that machine learning method may perform better than the method using empirical formula; the latter was developed completely based on biological knowledge. With more reliable and available experimental essential gene sets, the performance of our method will be improved to an even better level.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge Dr. Shuo Liu in University of Electronic Science and Technology for checking the paper and kindly providing some suggestions.

References

[1] C.-T. Zhang and R. Zhang, "Gene essentiality analysis based on DEG, a database of essential genes," *Methods in Molecular Biology*, vol. 416, pp. 391–400, 2007.

[2] K. Kobayashi, S. D. Ehrlich, A. Albertini et al., "Essential Bacillus subtilis genes," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 8, pp. 4678–4683, 2003.

[3] M. Juhas, D. R. Reuß, B. Zhu, and F. M. Commichau, "*Bacillus subtilis* and *Escherichia coli* essential genes and minimal cell factories after one decade of genome engineering," *Microbiology*, vol. 160, pp. 2341–2351, 2014.

[4] M. Itaya, "An estimation of minimal genome size required for life," *FEBS Letters*, vol. 362, no. 3, pp. 257–260, 1995.

[5] J. E. Dickerson, A. Zhu, D. L. Robertson, and K. E. Hentges, "Defining the role of essential genes in human disease," *PLoS ONE*, vol. 6, no. 11, Article ID e27368, 2011.

[6] I. K. Jordan, I. B. Rogozin, Y. I. Wolf, and E. V. Koonin, "Essential genes are more evolutionarily conserved than are nonessential genes in bacteria," *Genome Research*, vol. 12, no. 6, pp. 962–968, 2002.

[7] H. Luo, F. Gao, and Y. Lin, "Evolutionary conservation analysis between the essential and nonessential genes in bacterial genomes," *Scientific Reports*, vol. 5, article 13210, 2015.

[8] G. Giaever, A. M. Chu, L. Ni et al., "Functional profiling of the *Saccharomyces cerevisiae* genome," *Nature*, vol. 418, pp. 387–391, 2002.

[9] T. Roemer, B. Jiang, J. Davison et al., "Large-scale essential gene identification in *Candida albicans* and applications to antifungal drug discovery," *Molecular Microbiology*, vol. 50, no. 1, pp. 167–181, 2003.

[10] L. M. Cullen and G. M. Arndt, "Genome-wide screening for gene function using RNAi in mammalian cells," *Immunology and Cell Biology*, vol. 83, no. 3, pp. 217–223, 2005.

[11] Y. Veeranagouda, F. Husain, E. L. Tenorio, and H. M. Wexler, "Identification of genes required for the survival of *B. fragilis* using massive parallel sequencing of a saturated transposon mutant library," *BMC Genomics*, vol. 15, article 429, 2014.

[12] J. Deng, L. Deng, S. Su et al., "Investigating the predictability of essential genes across distantly related organisms using an integrative approach," *Nucleic Acids Research*, vol. 39, no. 3, pp. 795–807, 2011.

[13] K. Song, T. Tong, and F. Wu, "Predicting essential genes in prokaryotic genomes using a linear method: ZUPLS," *Integrative Biology*, vol. 6, no. 4, pp. 460–469, 2014.

[14] J. Cheng, W. Wu, Y. Zhang et al., "A new computational strategy for predicting essential genes," *BMC Genomics*, vol. 14, no. 1, article 910, 2013.

[15] A. M. Gustafson, E. S. Snitkin, S. C. J. Parker, C. DeLisi, and S. Kasif, "Towards the identification of essential genes using targeted genome sequencing and comparative analysis," *BMC Genomics*, vol. 7, article 265, 2006.

[16] M. L. Acencio and N. Lemke, "Towards the prediction of essential genes by integration of network topology, cellular localization and biological process information," *BMC Bioinformatics*, vol. 10, article 1471, p. 290, 2009.

[17] Y. Xu, M. Guo, X. Liu, C. Wang, and Y. Liu, "Inferring the soybean (*Glycine max*) microrna functional network based on target gene network," *Bioinformatics*, vol. 30, no. 1, pp. 94–103, 2014.

[18] Y. Xu, M. Guo, W. Shi, X. Liu, and C. Wang, "A novel insight into Gene Ontology semantic similarity," *Genomics*, vol. 101, no. 6, pp. 368–375, 2013.

[19] C. G. Acevedo-Rocha, G. Fang, M. Schmidt, D. W. Ussery, and A. Danchin, "From essential to persistent genes: a functional approach to constructing synthetic life," *Trends in Genetics*, vol. 29, no. 5, pp. 273–279, 2013.

[20] Q. Zou, Q. Hu, M. Guo, and G. Wang, "HAlign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[21] W. Wei, L.-W. Ning, Y.-N. Ye, and F.-B. Guo, "Geptop: a gene essentiality prediction tool for sequenced bacterial genomes based on orthology and phylogeny," *PLoS ONE*, vol. 8, no. 8, article e72343, 2013.

[22] Q. Zou, J. Li, Q. Hong et al., "Prediction of microRNA-disease associations based on social network analysis methods," *BioMed Research International*, vol. 2015, Article ID 810514, 9 pages, 2015.

[23] Y. Yamazaki, H. Niki, and J.-I. Kato, "Profiling of *Escherichia coli* Chromosome database," in *Microbial Gene Essentiality: Protocols and Bioinformatics*, A. L. Osterman and S. Y. Gerdes, Eds., vol. 416 of *Methods in Molecular Biology™*, pp. 385–389, 2008.

[24] F. M. Mobegi, A. Zomer, M. I. de Jonge, and S. A. van Hijum, "Advances and perspectives in computational prediction of microbial gene essentiality," *Briefings in Functional Genomics*, 2016.

[25] J. Qi, B. Wang, and B.-I. Hao, "Whole proteome prokaryote phylogeny without sequence alignment: a K-string composition approach," *Journal of Molecular Evolution*, vol. 58, no. 1, pp. 1–11, 2004.

[26] E. Byvatov and G. Schneider, "Support vector machine applications in bioinformatics," *Appl Bioinformatics*, vol. 2, no. 2, pp. 67–77, 2003.

[27] M. Pirooznia and Y. Deng, "SVM Classifier—a comprehensive java interface for support vector machine classification of microarray data," *BMC Bioinformatics*, vol. 7, supplement 4, article S25, 2006.

[28] Y.-W. Chen and C.-J. Lin, "Combining SVMs with various feature selection strategies," *Studies in Fuzziness and Soft Computing*, vol. 207, pp. 315–324, 2006.

[29] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[30] Z. Xu and B. Hao, "CVTree update: a newly designed phylogenetic study platform using composition vectors and whole genomes," *Nucleic Acids Research*, vol. 37, no. 2, pp. W174–W178, 2009.

[31] H. Tan, J. Bao, and X. Zhou, "A novel missense-mutation-related feature extraction scheme for 'driver' mutation identification," *Bioinformatics*, vol. 28, no. 22, pp. 2948–2955, 2012.

[32] B. E. Rubin, K. M. Wetmore, M. N. Price et al., "The essential gene set of a photosynthetic organism," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 112, no. 48, pp. E6634–E6643, 2015.

[33] S. Y. Gerdes, M. D. Scholle, J. W. Campbell et al., "Experimental determination and system level analysis of essential genes in Escherichia coli MG1655," *Journal of Bacteriology*, vol. 185, no. 19, pp. 5673–5684, 2003.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 6802832, 10 pages
http://dx.doi.org/10.1155/2016/6802832

Research Article

ProFold: Protein Fold Classification with Additional Structural Features and a Novel Ensemble Classifier

Daozheng Chen,[1] Xiaoyu Tian,[1] Bo Zhou,[2] and Jun Gao[1]

[1]*College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China*
[2]*Shanghai University of Medicine & Health Sciences, Shanghai 201318, China*

Correspondence should be addressed to Bo Zhou; zhoub@sumhs.edu.cn and Jun Gao; jungao@shmtu.edu.cn

Received 1 June 2016; Revised 15 July 2016; Accepted 7 August 2016

Academic Editor: Dariusz Mrozek

Protein fold classification plays an important role in both protein functional analysis and drug design. The number of proteins in PDB is very large, but only a very small part is categorized and stored in the SCOPe database. Therefore, it is necessary to develop an efficient method for protein fold classification. In recent years, a variety of classification methods have been used in many protein fold classification studies. In this study, we propose a novel classification method called proFold. We import protein tertiary structure in the period of feature extraction and employ a novel ensemble strategy in the period of classifier training. Compared with existing similar ensemble classifiers using the same widely used dataset (DD-dataset), proFold achieves 76.2% overall accuracy. Another two commonly used datasets, EDD-dataset and TG-dataset, are also tested, of which the accuracies are 93.2% and 94.3%, higher than the existing methods. ProFold is available to the public as a web-server.

1. Introduction

Protein fold classification is a crucial problem in structural bioinformatics. Protein folding information is helpful in identifying the tertiary structure and functional information of a protein [1]. In recent years, many protein fold classification studies have been performed. The methods proposed by researchers can be roughly divided into two categories: one is template-based method [2–7], and the other is taxonomy-based method [8–15]. Recently, taxonomy-based methods have attracted more attention due to their relatively excellent performance.

The taxonomy-based method was proposed by Dubchak et al. [8, 9] in 1995 for the first time. Many taxonomy-based methods classify a query protein to a known folding type. This nonmanual label method contributes to the growth of the quantity of protein in Structural Classification of Proteins (SCOP) [16] and could narrow the gap between the number of proteins in SCOP and Protein Data Bank (PDB). In this paper, the taxonomy-based method is equivalent to the classification problem in machine learning. There are two significant problems in classification tasks: one is feature extraction, and the other is machine learning method.

In terms of feature extraction, most of the researchers extract multidimensional numerical feature vectors from amino acid sequences. In 1995, Dubchak et al. [8, 9] extracted global description of amino acid sequence for the first time. Since then, in order to improve the accuracy of classification, some researchers have put forward other feature extraction methods, such as pseudoamino acid composition [12, 17], pairwise frequency information [18], Position Specific Scoring Matrix (PSSM) [17], structural properties of amino acid residues and amino acid residue pairs [19], and hidden Markov model structural alphabet [20, 21]. Except for extracting features from amino acid sequence directly, some features are extracted from evolution information combining the functional domain and the sequential evolution information [22] and predicted secondary structure [14, 23, 24]. Although the classification accuracy can be improved after combining these features together [20, 25], it is still not good enough.

For protein fold classification, many classifiers have been used, such as neutral network (NNs) [8, 13], SVMs [10, 13, 18–21, 24, 26–33], k-nearest neighbors (k-NN) [12], probabilistic multiclass multikernel classifier [25], random forest [23, 34–37], rotation forest [38], and a variety of ensemble classifiers [11, 12, 14, 18, 22, 39–41].

Up to 28th April, 2016, PDB had 109850 protein structures (http://www.rcsb.org/pdb/home/home.do). However, Structural Classification of Proteins- extended (SCOPe) [42] only had 77439 PDB entries (http://scop.berkeley.edu/statistics/ver=2.06). Therefore, there still exists a great number of protein structures which do not have their structure classification labels in the SCOPe database. What is more, most protein structures in SCOPe are classified manually, so it requires a lot of manual labor. In this study, we start from the PDB file 3D structure studying the protein fold classification. In terms of feature extraction, we use a new feature extraction method, combining the existing methods of the global description of amino acid sequence [13], PSSM [43], and protein functional information [22] proposed by other researchers. The new feature extraction method extracts eight types of secondary structure states from PDB files by the Definition of Secondary Structure in Proteins (DSSP) software [44]. In terms of machine learning classifiers, we propose a novel ensemble strategy. With the new added feature extracted from DSSP and the novel ensemble strategy we propose, our method can achieve 1–3% higher accuracy than similar methods.

As demonstrated by a series of recent publications [45–55] in compliance with Chou's 5-step rule [56], to establish a really useful machine learning classifier for a biological system, we should follow the following five guidelines: (a) benchmark dataset construction or selection for training and testing the model; (b) extract features from the biological sequence samples with effective methods that can truly reflect their intrinsic correlation with the target to be predicted; (c) introduce or develop a powerful algorithm (or engine) to operate the classifier; (d) properly perform cross-validation tests and test on independent dataset to objectively evaluate the anticipated accuracy of the classifier; (e) establish a user-friendly web-server (http://binfo.shmtu.edu.cn/profold/) for the classifier that is accessible to the public. In the following, we are to describe how to deal with these steps one-by-one.

2. Materials and Methods

2.1. Data Sets. In this study, three benchmark datasets are used, respectively: (1) Ding and Dubchak (DD) [13], (2) Taguchi and Gromiha (TG) [58], and (3) Extended DD (EDD) [10]. DD-dataset was proposed by Ding and Dubchak in 2001 and modified by Shen and Chou in 2006 [12]. Since then, DD-dataset has been used in many protein fold classification studies [11, 18, 20–24, 26, 32–36, 38, 40, 57, 59]. There are 311 protein sequences in the training set and 386 protein sequences in the testing set with no two proteins having more than 35% of sequence identity. The protein sequences in DD-dataset were selected from 27 SCOP [35] folds comprehensively, which belong to different structural classes containing α, β, α/β, and $\alpha + \beta$.

TG-dataset contains 30 SCOP folds and 1612 protein sequences with no two protein having more than 25% sequence identity.

EDD-dataset contains 27 SCOP folds, like DD-dataset. There are 3418 protein sequences with no protein having more than 40% sequence identity.

These three datasets can be downloaded directly from our website (http://binfo.shmtu.edu.cn/profold/benchmark.html).

2.2. Feature Extraction Method. With the rapid growth of biological sequences in the postgenomic age, one of the most important but also most difficult problems in computational biology is how to represent a biological sequence with a discrete model or a vector. Therefore Chou's PseAAC [60–62] was proposed. Encouraged by the successes of using PseAAC to deal with protein/peptide sequences, three web-servers [63–65] were developed for generating various feature vectors for DNA/RNA sequences. Particularly, recently a powerful web-server called Pse-in-One [66] has been established that can be used to generate any desired feature vectors for protein/peptide and DNA/RNA sequences according to the need of users' studies. Inspired by this, in this study, we extract four feature groups, including the DSSP feature, the amino acid composition and physicochemical properties (AAsCPP) feature, the PSSM feature, and the functional domain (FunD) composition feature. These feature extraction methods will be described as follows.

2.2.1. Definition of Secondary Structure in Proteins. The DSSP program was designed by Kabsch and Sander [44] and used to standardize protein secondary structure. The DSSP program works by calculating the most likely protein secondary structure given by the protein 3-dimensional structure. The specific principle of the DSSP program is calculating the H-bond energy between every two atoms by the atomic position in a PDB file, and then the most likely class of secondary structure for each residue can be determined by the best two H-bonds of each atom.

The DSSP feature extraction process is as follows. Firstly, DSSP entries are calculated from PDB entries by DSSP program. Secondly, the corresponding DSSP sequences from DSSP entries are obtained. DSSP sequence contains eight states (T, S, G, H, I, B, E, —), which can be divided into four groups, as shown in Table 1. Finally, according to the eight states and four groups, a 40D feature vector can be extracted from a DSSP sequence. The detail of the description and dimension of the features are shown in Table 2.

2.2.2. Amino Acids Composition and Physicochemical Properties. As effective features to describe a protein, the amino acid composition and physiochemical properties have reached good predict result, respectively [13, 34, 35]. Ding and Dubchak [13] tried to integrate the features for the first time and achieved a good result. Later, many other researchers proposed other feature integration methods. In 2013, Lin et al. [41] used a 188D feature vector combining amino acid composition and physiochemical properties. The 188D feature extraction method is also used in this paper.

The eight physiochemical properties of amino acids are hydrophobicity, van der Waals volume, polarity, polarizability, charge, surface tension, surface tension, and solvent accessibility. Different kinds of amino acids have different physiochemical properties so that they can be divided into three groups [13, 41], as shown in Table 3.

Table 1: The eight states of DSSP feature in four groups.

Eight-state SS	Code	Description	Four groups
3_{10} helix (G)	G	Helix-3	First
Alpha-helix (H)	H	Alpha helix	
pi-helix (I)	I	Helix-5	
Beta-strand (E)	E	Strand	Second
Beta-bridge (B)	B	Beta bridge	
Beta-turn (T)	T	Turn	Third
High curvature loop (S)	S	Bend	
Irregular (L)	—	Empty, no secondary structure assigned	Fourth

Table 2: The description and dimension of the DSSP feature.

Features description	Dimension
State composition	8
Group composition	4
Number of continuous states	8
Number of continuous groups	4
Number of continuous state compositions	8
Number of continuous group compositions	4
Alternate frequency between groups	4

The percentage composition of the 20 amino acids in the query protein forms a 20D feature vector. The group composition of amino acids (3D), the pairwise frequency between every two groups (3D), and the distribution pattern of constituents (where the first, 25%, 50%, 75%, and 100% of a given constituent are contained) (5 × 3D) from each physiochemical property are extracted. Therefore, we can get a 168D feature vector from a protein sequence according to the eight physiochemical properties. Adding up the 20D amino acid composition feature and the 168D physiochemical feature, we can get a 188D feature vector altogether. The name and the dimensions of the features are listed in Table 4.

2.2.3. Position Specific Scoring Matrix. PSSM is a relatively common feature. In addition to protein fold type classification research area, there are some studies on protein structural class prediction [67, 68] which used this feature. PSSM is derived from PSI-BLAST (Position Specific Iterative Basic Local Alignment Search Tool) [43] by taking the multiple sequence alignment of sequences in nonredundant protein sequence database (nrdb90) [69]. The iteration number is 3 and the cutoff E-value is 0.001. Two $L \times 20$ matrices can be obtained by PSI-BLAST, in which L represents the length of the query amino acid sequence, and 20 represents the 20 amino acids. One of the two matrices contains conservation scores of a given amino acid at a given position in sequence, and the other provides probability of occurrence of a given amino acid at a given position in the sequence. The PSSM feature is extracted from the former matrix. Suppose that the parameter in the matrix is S_{ij} ($i = 1, 2, \ldots, L$; $j = 1, 2, \ldots, 20$). Then the feature can be calculated by (1). That is to calculate the average value of each column in the matrix and form a 20D feature vector.

$$P_{\text{pssm}} = \left[\frac{\sum_{i=1}^{L} S_{i1}}{L}, \frac{\sum_{i=1}^{L} S_{i1}}{L}, \ldots, \frac{\sum_{i=1}^{L} S_{i20}}{L} \right]^{T}. \quad (1)$$

2.2.4. Functional Domain Composition. Proteins always contain some modules or domains, which involve different evolution resources and functions. Therefore, we can extract features in some FunD databases. There are some different FunD databases: SMART [70], Pfam [71], COG [72], KOG [72], and CDD [73]. In 2009, Shen and Chou [22] considered CDD as a relatively more complete functional domain database, and they used CDD to extract features. In this study, we used CDD (version 2.11), which co ntains 17402 common protein domains and families. Taking each of protein domains as a vector-base, we can extract a 17402D feature vector. Specific process is as follows. Firstly, use RPS-BLAST program [74] to compare the protein sequence with each of the 17402 domain sequences. Secondly, if the significance threshold value (expect value) is no more than 0.001, this component of the protein in the 17402D feature vector is assigned 1; otherwise, it is assigned 0. In this way, we can extract a 17402D feature vector, and each component of the feature can be either 1 or 0.

2.3. The Proposed Ensemble Classifier. In this study, we propose a novel ensemble strategy which includes 5 individual steps. Step 1: 10 widely used machine learning classifiers, LMT [75], RandomForest [34], LibSVM [76], SimpleLogistic [75], RotationForest [38], SMO [77], NaiveBayes [78], RandomTree [79], FT [80], and SimpleCart [81], are selected, and a 5-fold cross validation is implemented on the DD-dataset. Step 2: the classifier with the highest accuracy in each feature group is chosen. Step 3: corresponding models by training each feature group with the chosen classifier are selected. The four models are DSSP classification model, AAsCPP classification model, PSSM classification model, and FunD classification model. Detailed process is shown in Figure 1. Step 4: features from the test dataset are extracted and the classification result P_{ij} by calculating the corresponding models is obtained, i represents a kind of classification model ranging from 1 to 4, and j represents a kind of fold index, ranging from 1 to the total number of the fold classes (e.g.,

Table 3: The 20 amino acids divided into 3 groups according to their physiochemical properties.

Physicochemical property	The 1st group	The 2nd group	The 3rd group
Hydrophobicity	RKEDQN	GASTPHY	CVLIMFW
Van der Waals volume	GASCTPD	NVEQIL	MHKFRYW
Polarity	LIFWCMVF	PATGS	HQRKNED
Polarizability	GASDT	CPNVEQIL	KMHFRYW
Charge	KR	ANCQGHILMFPSTWYV	DE
Surface tension	GQDNAHR	KTSEC	ILMFPWYV
Secondary structure	EALMQKRH	VIYCWFT	GNPSD
Solvent accessibility	ALFCGIVW	RKQEND	MPSTHY

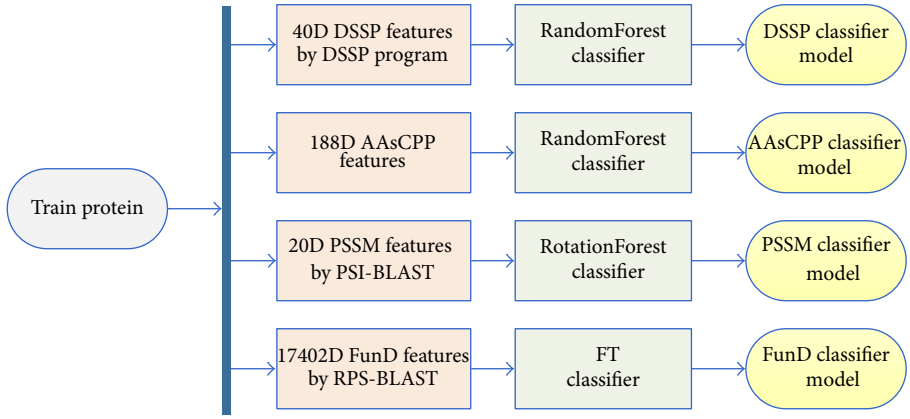


Figure 1: The training process of the four feature groups through the corresponding classifier.

Table 4: The name and the dimension of the amino acids composition and physiochemical features.

Feature name	Dimension
Amino acids composition	20
Hydrophobicity	21
Van der Waals volume	21
Polarity	21
Polarizability	21
Charge	21
Surface tension	21
Secondary structure	21
Solvent accessibility	21

the value of j ranges from 1 to 27 on DD-dataset). Step 5: the average of the probabilities of the four models in each fold class is calculated. The fold class with the highest probability will be chosen as the classification result. Detailed process is shown in Figure 2.

The machine learning tool we used is WEKA (Waikato Environment for Knowledge Analysis) [56], a collection of machine learning classifiers for data mining tasks based on Java.

2.4. Measurement. In this study, the standard Q percentage accuracy is used to test the effect of the proposed classification method, which helped us to compare our result with other researchers' results [12, 13, 34]. The definition of the standard Q percentage accuracy is described in

$$N = n_1 + n_2 + \cdots + n_i + \cdots + n_k,$$
$$C = c_1 + c_2 + \cdots + c_i + \cdots + c_k, \quad (2)$$
$$Q = \frac{C}{N},$$

where n_i represents the number of the proteins which belong to class i, c_i represents the correct number in n_i test data, c_i/n_i represents the classification accuracy of class i, k represents the total number of classes, N represents the total number of tests, C represents the total number of the correct classified data, and Q represents the classification accuracy.

3. Results and Discussion

3.1. Performance of ProFold. In order to test the performance of proFold, we first select the widely used DD-dataset for evaluation. The overall accuracy is 76.2%. Comparison with existing ensemble learning methods on DD-dataset is shown in Table 5. From Table 5, we can see that the accuracy of the other methods are under 75%, and the accuracy of our method is 3% higher than PFPA (2015) [40], which is the best one in the other methods.

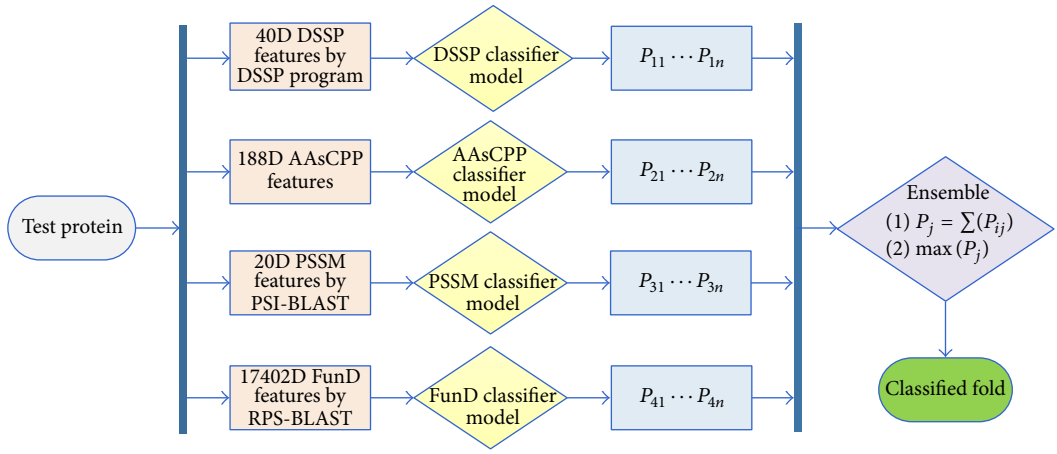


Figure 2: The ensemble process of calculating the test data through the models.

Table 5: Comparison with existing ensemble learning methods on DD-dataset.

Methods	References	Overall accuracy (%)
PFP-Pred	[12]	62.1
GAOEC	[11]	64.7
ThePFP-FunDSeqE	[22]	70.5
Dehzangi et al.	[34]	62.7
Dehzangi et al.	[38]	62.4
MarFold	[18]	71.7
PFP-RFSM	[35]	73.7
Feng and Hu	[36]	70.2
Feng et al.	[23]	70.8
PFPA	[40]	73.6
proFold (the proposed method)	*This paper*	*76.2*

In order to further evaluate the performance of proFold, we also select another two large scale datasets: EDD-dataset and TG-dataset. Training and testing dataset are not clearly distinguished in the two datasets, so a k-fold cross validation is implemented on them.

We calculated the classification accuracy of EDD-dataset by 10-fold cross validation for 10 times and compared the result with other methods. The results are shown in Table 6. We can see from the table that only the accuracies of Paliwal et al. and Lyons et al. are more than 90%, which are lower than that of proFold. The result showed that the advantage of proFold is obvious when larger scale datasets are used for validation.

Regarding TG-dataset, we also took experiments by 10-fold cross validation for 10 times and compared the results with other methods. The results are shown in Table 7. We can see from the table that HMMFold (2015) method achieved the highest accuracy, which is 93.8%. The accuracy of our method is 94.3%, which is higher than HMMFold. TG-dataset has threefold classes more than DD-dataset and its scale is twice larger than DD-dataset. The result showed that the advantage of proFold is obvious when the dataset with more fold classes is tested.

Table 6: Comparison with the different methods on EDD-dataset by 10-fold cross validation.

Methods	References	Overall accuracy (%)
Paliwal et al.	[29]	90.6
Paliwal et al.	[30]	86.2
Dehzangi et al.	[31]	88.2
HMMFold	[32]	86.0
Saini et al.	[33]	89.9
Lyons et al.	[21]	92.9
proFold (the proposed method)	*This paper*	*93.2*

Table 7: Comparison with the different methods on TG-dataset by 10-fold cross validation.

Methods	References	Overall accuracy (%)
Paliwal et al.	[29]	77.0
Paliwal et al.	[30]	73.3
Dehzangi et al.	[31]	73.8
HMMFold	[32]	93.8
Saini et al.	[33]	74.5
NiRecor	[57]	84.6
Lyons et al.	[21]	85.6
proFold (the proposed method)	*This paper*	*94.3*

3.2. Performance of the Proposed Ensemble Classifier. In the field of protein fold classification, many researchers used ensemble learning methods [11, 18, 22, 23, 34–36, 38, 46, 51, 54, 79, 82–89]. The specific process of those ensemble strategies is as follows. (1) Integrate all features. (2) Select several basic classifiers for training. (3) Propose an ensemble classifier

Table 8: The accuracy of 5-fold cross validation on the features extracted from DD-dataset using 10 basic classifiers.

Feature groups	Basic classifiers	Fivefold CV accuracy (%)
DSSP	LMT	43.0
	RandomForest*	51.3
	LibSVM	46.4
	SimpleLogistic	43.0
	RotationForest	49.7
	SMO	36.4
	NaiveBayes	43.4
	RandomTree	32.8
	FT	42.4
	SimpleCart	37.7
AAsCPC	LMT	32.5
	RandomForest*	35.4
	LibSVM	34.4
	SimpleLogistic	32.5
	RotationForest	27.7
	SMO	34.4
	NaiveBayes	28.3
	RandomTree	11.6
	FT	34.4
	SimpleCart	20.6
PSSM	LMT	56.3
	RandomForest	53.7
	LibSVM	57.2
	SimpleLogistic	55.9
	RotationForest*	56.1
	SMO	30.2
	NaiveBayes	42.4
	RandomTree	29.6
	FT	49.5
	SimpleCart	33.4
FunD	LMT	42.1
	RandomForest	43.1
	LibSVM	21.2
	SimpleLogistic	43.1
	RotationForest	41.8
	SMO	38.9
	NaiveBayes	38.3
	RandomTree	39.9
	FT*	44.1
	SimpleCart	34.7

*The basic classifier of each feature group with the highest accuracy.

according to the classification result probability of each basic classifier. In this study, we find that the redundancies of the features will influence the performance of those methods. Therefore, we propose a novel ensemble strategy.

We took experiments on DD-dataset. Firstly, extract four feature groups which have been tested in 10 basic classifiers by cross validation. The detailed information of the test results is listed in Table 8. We can see from Table 8 that the best classifier is RandomForest using the DSSP feature group and AAsCPP feature group. The best classifiers are RotaionForest and FT when PSSM and FunD features are implemented, respectively. Secondly, train the four feature groups with corresponding basic classifiers and get four models. Finally, test the models on DD-dataset. The overall accuracy is 76.2%. Our method improves the accuracy effectively compared with other existing ensemble learning methods.

Table 9: Comparison with the different ensemble strategies on three datasets.

Datasets	The accuracy of traditional ensemble strategy (%)	The accuracy of this paper ensemble strategy (%)
DD	72.5	76.2
EDD	89.9	93.2
TG	91.7	94.3

In order to compare our ensemble strategy with the traditional ensemble strategy, we took experiments on the four feature groups with traditional ensemble strategy. (1) Integrate the four feature groups. (2) Train the models with RandomForest, RotationForest, and FT respectively. (3) Test the models on DD-dataset, EDD-dataset, and TG-dataset. The classification accuracy of our ensemble strategy has increased by 3% to 4%, as shown in Table 9. The result showed that our ensemble strategy has a better classification performance.

3.3. Accuracy Improvements with the DSSP Feature. In order to evaluate the influence on importing the DSSP feature, we calculated the classification accuracy of each fold class with and without the DSSP feature, respectively, using the DD-dataset. The accuracies are shown in Table 10. From the table, we can see that the accuracies of some fold classes, such as Fold number 2, number 4, number 6, number 12, number 23, and number 26, have increased obviously after importing the DSSP feature. The overall accuracy has increased from 71.3% to 76.2%. For example, the protein chain 1FAPB in DD-dataset was incorrectly classified into Fold number 5 before importing the DSSP feature, and it was reclassified into Fold number 4 correctly after importing the DSSP feature. The results showed that the DSSP feature has a significant effect on protein structure classification.

As we know that PDB files contain protein 3D structure information, we started from the PDB file of the protein in this study. The DSSP feature is extracted from the 3D structure in PDB and the 3D structure of a protein is more stable. Thus it explains why the DSSP feature has a significant effect on the protein structure classification.

4. Conclusion

In this study, we proposed a novel method called proFold. ProFold is an ensemble classifier combining the protein structural and functional information. In terms of feature extraction, we imported the DSSP feature into protein fold

Table 10: The accuracy of each fold class with and without the DSSP feature.

Fold number	The accuracy without the DSSP feature	The accuracy with the DSSP feature
1	100.0	100.0
2*	88.9	100.0
3*	55.0	60.0
4*	62.5	87.5
5	88.9	88.9
6*	66.7	77.8
7*	77.3	84.1
8	66.7	66.7
9	92.3	92.3
10	66.7	66.7
11	50.0	50.0
12*	47.4	68.4
13	100.0	100.0
14	50.0	50.0
15	100.0	100.0
16*	91.7	93.8
17*	83.3	91.7
18*	38.5	46.2
19	85.2	85.2
20	50.0	50.0
21	87.5	87.5
22	58.3	58.3
23*	57.1	71.4
24	100.0	100.0
25	25.0	25.0
26*	44.4	59.3
27*	92.6	96.3
Overall	71.3	76.2

*The fold class of which the accuracy has increased significantly after importing the DSSP feature.

classification for the first time. Experiments showed that the classification accuracy will increase by about 5% using the DD-dataset by importing the DSSP feature. In terms of classification method, we proposed a novel ensemble classifier and improved the classification accuracy with this method. The classification accuracies of proFold on DD-, EDD-, and TG-dataset are 76.2%, 93.2%, and 94.3%, respectively, which are higher than the existing similar methods. The results showed that proFold is a relatively better classifier.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by grants from National Natural Science Foundation of China (Grant no. 61303099).

References

[1] H. S. Chan and K. A. Dill, "The protein folding problem," *Physics Today*, vol. 46, no. 2, pp. 24–32, 1993.

[2] D. T. Jones, W. R. Taylor, and J. M. Thornton, "A new approach to protein fold recognition," *Nature*, vol. 358, no. 6381, pp. 86–89, 1992.

[3] J. Xu, M. Li, D. Kim, and Y. Xu, "RAPTOR: optimal protein threading by linear programming," *Journal of Bioinformatics and Computational Biology*, vol. 1, no. 1, pp. 95–117, 2003.

[4] J. Söding, A. Biegert, and A. N. Lupas, "The HHpred interactive server for protein homology detection and structure prediction," *Nucleic Acids Research*, vol. 33, supplement 2, pp. W244–W248, 2005.

[5] J. Cheng and P. Baldi, "A machine learning information retrieval approach to protein fold recognition," *Bioinformatics*, vol. 22, no. 12, pp. 1456–1463, 2006.

[6] W. Zhang, S. Liu, and Y. Zhou, "SP^5: improving protein fold recognition by using torsion angle profiles and profile-based gap penalty model," *PLoS ONE*, vol. 3, no. 6, Article ID e2325, 2008.

[7] R.-X. Yan, J.-N. Si, C. Wang, and Z. Zhang, "DescFold: a web server for protein fold recognition," *BMC Bioinformatics*, vol. 10, article 416, 2009.

[8] I. Dubchak, I. Muchnik, S. R. Holbrook, and S.-H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 92, no. 19, pp. 8700–8704, 1995.

[9] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.-H. Kim, "Recognition of a protein fold in the context of the SCOP classification," *Proteins: Structure, Function, and Bioinformatics*, vol. 35, no. 4, pp. 401–407, 1999.

[10] Q. Dong, S. Zhou, and J. Guan, "A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation," *Bioinformatics*, vol. 25, no. 20, pp. 2655–2662, 2009.

[11] X. Guo and X. Gao, "A novel hierarchical ensemble classifier for protein fold recognition," *Protein Engineering, Design and Selection*, vol. 21, no. 11, pp. 659–664, 2008.

[12] H.-B. Shen and K.-C. Chou, "Ensemble classifier for protein fold pattern recognition," *Bioinformatics*, vol. 22, no. 14, pp. 1717–1722, 2006.

[13] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001.

[14] K. Chen and L. Kurgan, "PFRES: protein fold classification by using evolutionary information and predicted secondary structure," *Bioinformatics*, vol. 23, no. 21, pp. 2843–2850, 2007.

[15] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identificationof human microRNAs by incorporatinga high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.

[16] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, no. 4, pp. 536–540, 1995.

[17] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, 2001.

[18] T. Yang, V. Kecman, L. Cao, C. Zhang, and J. Zhexue Huang, "Margin-based ensemble classifier for protein fold recognition," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12348–12355, 2011.

[19] M. T. A. Shamim, M. Anwaruddin, and H. A. Nagarajaram, "Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs," *Bioinformatics*, vol. 23, no. 24, pp. 3320–3327, 2007.

[20] P. Deschavanne and P. Tufféry, "Enhanced protein fold recognition using a structural alphabet," *Proteins: Structure, Function and Bioinformatics*, vol. 76, no. 1, pp. 129–137, 2009.

[21] J. Lyons, K. K. Paliwal, A. Dehzangi, R. Heffernan, T. Tsunoda, and A. Sharma, "Protein fold recognition using HMM–HMM alignment and dynamic programming," *Journal of Theoretical Biology*, vol. 393, pp. 67–74, 2016.

[22] H.-B. Shen and K.-C. Chou, "Predicting protein fold pattern with functional domain and sequential evolution information," *Journal of Theoretical Biology*, vol. 256, no. 3, pp. 441–446, 2009.

[23] Z. Feng, X. Hu, Z. Jiang, H. Song, and M. A. Ashraf, "The recognition of multi-class protein folds by adding average chemical shifts of secondary structure elements," *Saudi Journal of Biological Sciences*, vol. 23, no. 2, pp. 189–197, 2016.

[24] J.-Y. Yang and X. Chen, "Improving taxonomy-based protein fold recognition by using global and local features," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. 7, pp. 2053–2064, 2011.

[25] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics*, vol. 24, no. 10, pp. 1264–1270, 2008.

[26] W. Chmielnicki and K. Stąpor, "A hybrid discriminative/generative approach to protein fold recognition," *Neurocomputing*, vol. 75, no. 1, pp. 194–198, 2012.

[27] L. Liu, X.-Z. Hu, X.-X. Liu, Y. Wang, and S.-B. Li, "Predicting protein fold types by the general form of Chou's pseudo amino acid composition: approached from optimal feature extractions," *Protein and Peptide Letters*, vol. 19, no. 4, pp. 439–449, 2012.

[28] A. Sharma, J. Lyons, A. Dehzangi, and K. K. Paliwal, "A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition," *Journal of Theoretical Biology*, vol. 320, pp. 41–46, 2013.

[29] K. K. Paliwal, A. Sharma, J. Lyons, and A. Dehzangi, "Improving protein fold recognition using the amalgamation of evolutionary-based and structural based information," *BMC Bioinformatics*, vol. 15, supplement 16, p. S12, 2014.

[30] K. K. Paliwal, A. Sharma, J. Lyons, and A. Dehzangi, "A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition," *IEEE Transactions on Nanobioscience*, vol. 13, no. 1, pp. 44–50, 2014.

[31] A. Dehzangi, K. Paliwal, J. Lyons, A. Sharma, and A. Sattar, "A segmentation-based method to extract structural and evolutionary features for protein fold recognition," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 510–519, 2014.

[32] J. Lyons, A. Dehzangi, R. Heffernan et al., "Advancing the accuracy of protein fold recognition by utilizing profiles from hidden markov models," *IEEE Transactions on NanoBioscience*, vol. 14, no. 7, pp. 761–772, 2015.

[33] H. Saini, G. Raicar, A. Sharma et al., "Probabilistic expression of spatially varied amino acid dimers into general form of Chou's pseudo amino acid composition for protein fold recognition," *Journal of Theoretical Biology*, vol. 380, pp. 291–298, 2015.

[34] A. Dehzangi, S. Phon-Amnuaisuk, and O. Dehzangi, "Using random forest for protein fold prediction problem: an empirical study," *Journal of Information Science and Engineering*, vol. 26, no. 6, pp. 1941–1956, 2010.

[35] J. Li, J. Wu, and K. Chen, "PFP-RFSM: protein fold prediction by using random forests and sequence motifs," *Journal of Biomedical Science and Engineering*, vol. 6, no. 12, pp. 1161–1170, 2013.

[36] Z. Feng and X. Hu, "Recognition of 27-class protein folds by adding the interaction of segments and motif information," *BioMed Research International*, vol. 2014, Article ID 262850, 9 pages, 2014.

[37] L. Wei, M. Liao, X. Gao, and Q. Zou, "An improved protein structural classes prediction method by incorporating both sequence and structure information," *IEEE Transactions on Nanobioscience*, vol. 14, no. 4, pp. 339–349, 2015.

[38] A. Dehzangi, S. Phon-Amnuaisuk, M. Manafi, and S. Safa, "Using rotation forest for protein fold prediction problem: an empirical study," in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, C. Pizzuti, M. D. Ritchie, and M. Giacobini, Eds., vol. 6023 of *Lecture Notes in Computer Science*, pp. 217–227, Springer, Berlin, Germany, 2010.

[39] G. Bologna and R. D. Appel, "A comparison study on protein fold recognition," in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP '02)*, pp. 2492–2496, IEEE, Singapore, November 2002.

[40] L. Wei, M. Liao, X. Gao, and Q. Zou, "Enhanced protein fold prediction method through a novel feature extraction technique," *IEEE Transactions on NanoBioscience*, vol. 14, no. 6, pp. 649–659, 2015.

[41] C. Lin, Y. Zou, J. Qin et al., "Hierarchical classification of protein folds using a novel ensemble classifier," *PLoS ONE*, vol. 8, no. 2, article e56499, 2013.

[42] N. K. Fox, S. E. Brenner, and J.-M. Chandonia, "SCOPe: structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures," *Nucleic Acids Research*, vol. 42, no. 1, pp. D304–D309, 2014.

[43] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[44] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.

[45] H. Lin, E.-Z. Deng, H. Ding, W. Chen, and K.-C. Chou, "IPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition," *Nucleic Acids Research*, vol. 42, no. 21, pp. 12961–12972, 2014.

[46] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "iPPI-Esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC," *Journal of Theoretical Biology*, vol. 377, pp. 47–56, 2015.

[47] W.-R. Qiu, X. Xiao, and K.-C. Chou, "iRSpot-TNCPseAAC: identify recombination spots with trinucleotide composition

and pseudo amino acid components," *International Journal of Molecular Sciences*, vol. 15, no. 2, pp. 1746–1766, 2014.

[48] W. Chen, H. Ding, P. Feng, H. Lin, and K.-C. Chou, "iACP: a sequence-based tool for identifying anticancer peptides," *Oncotarget*, vol. 7, no. 13, pp. 16895–16909, 2016.

[49] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "iSuc-PseOpt: identifying lysine succinylation sites in proteins by incorporating sequence-coupling effects into pseudo components and optimizing imbalanced training dataset," *Analytical Biochemistry*, vol. 497, pp. 48–56, 2016.

[50] B. Liu, L. Fang, F. Liu, X. Wang, and K.-C. Chou, "IMiRNA-PseDPC: microRNA precursor identification with a pseudo distance-pair composition approach," *Journal of Biomolecular Structure and Dynamics*, vol. 34, no. 1, pp. 220–232, 2016.

[51] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "pSuc-Lys: predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach," *Journal of Theoretical Biology*, vol. 394, pp. 223–230, 2016.

[52] B. Liu, L. Fang, R. Long, X. Lan, and K.-C. Chou, "iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 32, no. 3, pp. 362–369, 2016.

[53] J. Jia, Z. Liu, X. Xiao, B. Liu, and K. Chou, "iCar-PseCp: identify carbonylation sites in proteins by Monto Carlo sampling and incorporating sequence coupled effects into general PseAAC," *Oncotarget*, vol. 7, no. 23, pp. 34558–34570, 2016.

[54] B. Liu, R. Long, and K.-C. Chou, "iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework," *Bioinformatics*, vol. 32, no. 16, pp. 2411–2418, 2016.

[55] W.-R. Qiu, B.-Q. Sun, X. Xiao, D. Xu, and K.-C. Chou, "iPhos-PseEvo: identifying human phosphorylated proteins by incorporating evolutionary information into general PseAAC via grey system theory," *Molecular Informatics*, 2016.

[56] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 273, no. 1, pp. 236–247, 2011.

[57] N. J. Cheung, X.-M. Ding, and H.-B. Shen, "Protein folds recognized by an intelligent predictor based-on evolutionary and structural information," *Journal of Computational Chemistry*, vol. 37, no. 4, pp. 426–436, 2016.

[58] J. Lyons, N. Biswas, A. Sharma, A. Dehzangi, and K. K. Paliwal, "Protein fold recognition by alignment of amino acid residues using kernelized dynamic time warping," *Journal of Theoretical Biology*, vol. 354, pp. 137–145, 2014.

[59] K. Kavousi, B. Moshiri, M. Sadeghi, B. N. Araabi, and A. A. Moosavi-Movahedi, "A protein fold classifier formed by fusing different modes of pseudo amino acid composition via PSSM," *Computational Biology and Chemistry*, vol. 35, no. 1, pp. 1–9, 2011.

[60] P. Du, X. Wang, C. Xu, and Y. Gao, "PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions," *Analytical Biochemistry*, vol. 425, no. 2, pp. 117–119, 2012.

[61] D.-S. Cao, Q.-S. Xu, and Y.-Z. Liang, "Propy: a tool to generate various modes of Chou's PseAAC," *Bioinformatics*, vol. 29, no. 7, pp. 960–962, 2013.

[62] S.-X. Lin and J. Lapointe, "Theoretical and experimental biology in one—a symposium in honour of Professor Kuo-Chen Chou's 50th anniversary and Professor Richard Giegé's 40th anniversary of their scientific careers," *Journal of Biomedical Science and Engineering*, vol. 6, no. 4, pp. 435–442, 2013.

[63] W. Chen, T.-Y. Lei, D.-C. Jin, H. Lin, and K.-C. Chou, "PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition," *Analytical Biochemistry*, vol. 456, no. 1, pp. 53–60, 2014.

[64] W. Chen, X. Zhang, J. Brooker, H. Lin, L. Zhang, and K.-C. Chou, "PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions," *Bioinformatics*, vol. 31, no. 1, pp. 119–120, 2015.

[65] B. Liu, F. Liu, L. Fang, X. Wang, and K.-C. Chou, "RepDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects," *Bioinformatics*, vol. 31, no. 8, pp. 1307–1309, 2015.

[66] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K.-C. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. 1, pp. W65–W71, 2015.

[67] L. Zhang, X. Zhao, and L. Kong, "Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 355, pp. 105–110, 2014.

[68] S. Zhang, "Accurate prediction of protein structural classes by incorporating PSSS and PSSM into Chou's general PseAAC," *Chemometrics and Intelligent Laboratory Systems*, vol. 142, pp. 28–35, 2015.

[69] L. Holm and C. Sander, "Removing near-neighbour redundancy from large protein sequence collections," *Bioinformatics*, vol. 14, no. 5, pp. 423–429, 1998.

[70] I. Letunic, R. R. Copley, B. Pils, S. Pinkert, J. Schultz, and P. Bork, "SMART 5: domains in the context of genomes and networks," *Nucleic Acids Research*, vol. 34, supplement 1, pp. D257–D260, 2006.

[71] R. D. Finn, J. Mistry, B. Schuster-Böckler et al., "Pfam: clans, web tools and services," *Nucleic Acids Research*, vol. 34, supplement 1, pp. D247–D251, 2006.

[72] R. L. Tatusov, N. D. Fedorova, J. D. Jackson et al., "The COG database: an updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, article 41, 2003.

[73] A. Marchler-Bauer, J. B. Anderson, M. K. Derbyshire et al., "CDD: a conserved domain database for interactive domain family analysis," *Nucleic Acids Research*, vol. 35, no. 1, pp. D237–D240, 2007.

[74] A. A. Schäffer, L. Aravind, T. L. Madden et al., "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements," *Nucleic Acids Research*, vol. 29, no. 14, pp. 2994–3005, 2001.

[75] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[76] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[77] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.

[78] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, Morgan Kaufmann, 1995.

[79] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: a new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1619–1630, 2006.

[80] J. Gama, "Functional trees," *Machine Learning*, vol. 55, no. 3, pp. 219–250, 2004.

[81] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[82] K.-C. Chou and H.-B. Shen, "Predicting protein subcellular location by fusing multiple classifiers," *Journal of Cellular Biochemistry*, vol. 99, no. 2, pp. 517–527, 2006.

[83] K.-C. Chou and H.-B. Shen, "Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-nearest neighbor classifiers," *Journal of Proteome Research*, vol. 5, no. 8, pp. 1888–1897, 2006.

[84] K.-C. Chou and H.-B. Shen, "Signal-CF: a subsite-coupled and window-fusing approach for predicting signal peptides," *Biochemical and Biophysical Research Communications*, vol. 357, no. 3, pp. 633–640, 2007.

[85] H.-B. Shen and K.-C. Chou, "Gpos-PLoc: an ensemble classifier for predicting subcellular localization of Gram-positive bacterial proteins," *Protein Engineering, Design and Selection*, vol. 20, no. 1, pp. 39–46, 2007.

[86] H.-B. Shen and K.-C. Chou, "Hum-mPLoc: an ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites," *Biochemical and Biophysical Research Communications*, vol. 355, no. 4, pp. 1006–1011, 2007.

[87] H.-B. Shen, J. Yang, and K.-C. Chou, "Euk-PLoc: an ensemble classifier for large-scale eukaryotic protein subcellular location prediction," *Amino Acids*, vol. 33, no. 1, pp. 57–67, 2007.

[88] H.-B. Shen and K.-C. Chou, "QuatIdent: a web server for identifying protein quaternary structural attribute by fusing functional domain and sequential evolution information," *Journal of Proteome Research*, vol. 8, no. 3, pp. 1577–1584, 2009.

[89] J. Jia, Z. Liu, X. Xiao, B. Liu, and K.-C. Chou, "iPPBS-Opt: a sequence-based ensemble classifier for identifying protein-protein binding sites by optimizing imbalanced training datasets," *Molecules*, vol. 21, no. 1, p. 95, 2016.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 8527435, 7 pages
http://dx.doi.org/10.1155/2016/8527435

Research Article

Recombination Hotspot/Coldspot Identification Combining Three Different Pseudocomponents via an Ensemble Learning Approach

Bingquan Liu,[1] Yumeng Liu,[2] and Dong Huang[2,3]

[1]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang, China*
[2]*School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China*
[3]*Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China*

Correspondence should be addressed to Bingquan Liu; liubq@hit.edu.cn

Received 30 May 2016; Accepted 11 July 2016

Academic Editor: Qin Ma

Recombination presents a nonuniform distribution across the genome. Genomic regions that present relatively higher frequencies of recombination are called hotspots while those with relatively lower frequencies of recombination are recombination coldspots. Therefore, the identification of hotspots/coldspots could provide useful information for the study of the mechanism of recombination. In this study, a new computational predictor called SVM-EL was proposed to identify hotspots/coldspots across the yeast genome. It combined Support Vector Machines (SVMs) and Ensemble Learning (EL) based on three features including basic kmer (Kmer), dinucleotide-based auto-cross covariance (DACC), and pseudo dinucleotide composition (PseDNC). These features are able to incorporate the nucleic acid composition and their order information into the predictor. The proposed SVM-EL achieves an accuracy of 82.89% on a widely used benchmark dataset, which outperforms some related methods.

1. Introduction

Meiotic recombination describes the process of alleles' exchange between homologous chromosomes during meiosis [1]. It can provide material for natural selection by producing diverse gametes. It might also contribute to the evolution of the genome via gene conversion or mutagenesis [2–4].

Although the exact location where recombination happens in the genome and the mechanism of recombination are still unclear, it has been assured that recombination plays an important role in promoting genome evolution. Therefore, several studies have been performed on chromosomes [5–7] and found that recombination presents a nonuniform distribution across the genome. Genomic regions that present relatively higher frequencies of recombination are called hotspots while those with relatively lower frequencies of recombination are called recombination coldspots [8, 9]. With the number of the sequenced genomes showing explosive growth, more reliable methods are urgently needed to be developed to identify the recombination spots.

The prediction of recombination hotspots or coldspots is still a challenging task, although much information can be acquired from the experiments. Recently, several computational models have been presented to identify the recombination hotspots/coldspots. For example, Liu et al. [10], based on sequence Kmer frequencies, proposed a model which combines the increment of diversity with quadratic discriminant analysis (IDQD). Later, this method was improved by adding gaps into the kmers [11]. Chen et al. presented a predictor called iRSpot-PseDNC trained with pseudo dinucleotide composition features [12].

The aforementioned methods extracted the features from DNA sequences in different aspects. For example, the model based on oligonucleotide frequencies considers the nucleic acid composition information. The iRSpot-PseDNC incorporates both the local nucleic acid composition information and

the global information of the protein sequences. Therefore, it is reasonable to combine these complementary predictors to further improve the performance of recombination hotspot/coldspot identification. In this regard, three basic predictors trained with basic kmer (Kmer) [13], dinucleotide-based auto-cross covariance (DACC) [14, 15], and pseudo dinucleotide composition (PseDNC) [16], respectively, were combined via the framework of ensemble learning approach, and a novel predictor called SVM-EL was proposed. All these features can be easily generated by a recently proposed tool called Pse-in-One [17], which is able to generate various features only based on the DNA, RNA, or protein sequence information.

2. Materials and Methods

2.1. Benchmark Dataset. The benchmark datasets $\mathbf{S}$ was obtained from Liu et al. [10]:

$$\mathbf{S} = \mathbf{S}^{+} \cup \mathbf{S}^{-}, \tag{1}$$

where the subset $\mathbf{S}^{+}$ contains 490 recombination hotspots, the subset $\mathbf{S}^{-}$ contains 591 recombination coldspots, and the symbol $\cup$ represents the "union" in the set theory.

2.2. Feature Vectors Generated by Pse-in-One. SVM-EL is developed by combining the outcomes of three individual predictors which were trained by different features, including basic kmer (Kmer) [13], dinucleotide-based auto-cross covariance (DACC) [14, 15], and pseudo dinucleotide composition (PseDNC). These basic features can be generated by using Pse-in-One [17] which provides two approaches to generate feature vectors. One way is through the web server (http://bioinformatics.hitsz.edu.cn/Pse-in-One/) and another way is through the stand-alone tool (http://bioinformatics.hitsz.edu.cn/Pse-in-One/download/).

Suppose a DNA sequence $\mathbf{D}$ is

$$\mathbf{D} = R_1 R_2 R_3 R_4 R_5 \cdots R_L, \tag{2}$$

where L represents the DNA sequence length and R_i ($i = 1, 2 \cdots L$) is the nucleic acid at the position i. Therefore, three basic features used in the current study can be described as follows.

2.2.1. Kmer. Kmer [13] is an approach representing DNA sequences by the occurrence frequencies of kmers. The Kmer contains the local sequence-order information and it can be generated with the help of Pse-in-One by the following steps.

For web server approach, firstly, choose DNA sequences (PseDAC-General), then select Kmer in the tab of Mode, and set the value of k. Secondly, input or upload the DNA sequence file in FASTA format, click the Submit button, and then you will see the results and you can download them as a text file (Figure 1).

For stand-alone approach, Kmer features can be easily generated by using the following command line:

'./kmer.py –f svm –l +1 3 DNA'

where –f svm represents the format of the output file which is the LIBSVM training data format, –l +1 represents the input file that contains positive samples only, k equals 3, and the sequence type is DNA.

FIGURE 1: An example of the kmer features' generation by using Pse-in-One.

2.2.2. Dinucleotide-Based Auto-Cross Covariance (DACC). Dinucleotide-based auto-cross covariance (DACC) [14, 15] is the combination of DAC [14, 15, 19] and DCC [14, 15]. The DAC measures the correlation between two dinucleotides for one DNA property [17]. The DCC approach measures the correlation between two dinucleotides for two different properties [17].

Given a DNA sequence $\mathbf{D}$ represented as (2), the DAC feature can be calculated as [17]

$$\begin{aligned} &\mathrm{DAC}(\mu, \mathrm{lag}) \\ &= \sum_{i=1}^{L-\mathrm{lag}-1} \frac{\left(P_\mu\left(R_i R_{i+1}\right) - \overline{P}_\mu\right)\left(P_\mu\left(R_{i+\mathrm{lag}} R_{i+\mathrm{lag}+1}\right) - \overline{P}_\mu\right)}{L - \mathrm{lag} - 1}, \\ &\overline{P}_\mu = \sum_{j=1}^{L-1} \frac{P_\mu\left(R_j R_{j+1}\right)}{L-1}, \end{aligned} \tag{3}$$

where μ is the dinucleotide property index; L is the length of DNA sequence; lag represents the distance between two dinucleotides; $P_\mu(R_i R_{i+1})$ represents the value of dinucleotide $R_i R_{i+1}$ at position i for the dinucleotide property index μ; $\overline{P}_\mu$ represents the average value of $P_\mu(R_i R_{i+1})$ for a DNA sequence.

Given a DNA sequence $\mathbf{D}$ represented as (2), the DCC feature can be calculated as [17]

$$\begin{aligned} &\mathrm{DCC}(\mu_1, \mu_2, \mathrm{lag}) \\ &= \sum_{i=1}^{L-\mathrm{lag}-1} \frac{\left(P_{\mu_1}\left(R_i R_{i+1}\right) - \overline{P}_{\mu_1}\right)\left(P_{\mu_2}\left(R_{i+\mathrm{lag}} R_{i+\mathrm{lag}+1}\right) - \overline{P}_{\mu_2}\right)}{L - \mathrm{lag} - 1}, \\ &\overline{P}_\mu = \sum_{j=1}^{L-1} \frac{P_\mu\left(R_j R_{j+1}\right)}{L-1}, \end{aligned} \tag{4}$$

where μ_1 and μ_2 are two different dinucleotide property indices; L is the DNA sequence length; lag is the distance between two dinucleotides; $P_{\mu_1}(R_i R_{i+1})(P_{\mu_2}(R_i R_{i+1}))$ represents the value of dinucleotide $R_i R_{i+1}$ at position i for the

Table 1: The values of fifteen DNA dinucleotide properties.

	AA/TT	AC/GT	AG/CT	AT	CA/TG	CC/GG	CG	GA/TC	GC	TA
F-roll	0.04	0.06	0.04	0.05	0.04	0.04	0.04	0.05	0.05	0.03
F-tilt	0.08	0.07	0.06	0.10	0.06	0.06	0.06	0.07	0.07	0.07
F-twist	0.07	0.06	0.05	0.07	0.05	0.06	0.05	0.06	0.06	0.05
F-slide	6.69	6.80	3.47	9.61	2.00	2.99	2.71	4.27	4.21	1.85
F-shift	6.24	2.91	2.80	4.66	2.88	2.67	3.02	3.58	2.66	4.11
F-rise	21.34	21.98	17.48	24.79	14.51	14.25	14.66	18.41	17.31	14.24
Roll	1.05	2.01	3.60	0.61	5.60	4.68	6.02	2.44	1.70	3.50
Tilt	−1.26	0.33	−1.66	0.00	0.14	−0.77	0.00	1.44	0.00	0.00
Twist	35.02	31.53	32.29	30.72	35.43	33.54	33.67	35.67	34.07	36.94
Slide	−0.18	−0.59	−0.22	−0.68	0.48	−0.17	0.44	−0.05	−0.19	0.04
Shift	0.01	−0.02	−0.02	0.00	0.01	0.03	0.00	−0.01	0.00	0.00
Rise	3.25	3.24	3.32	3.21	3.37	3.36	3.29	3.30	3.27	3.39
Energy	−1.00	−1.44	−1.28	−0.88	−1.45	−1.84	−2.17	−1.30	−2.24	−0.58
Enthalpy	−7.60	−8.40	−7.80	−7.20	−8.50	−8.00	−10.60	−8.20	−9.80	−7.20
Entropy	−21.30	−22.40	−21.00	−20.40	−22.70	−19.90	−27.20	−22.20	−24.40	−21.30

dinucleotide property index $\mu_1(\mu_2)$; $\overline{P}_{\mu_1}(\overline{P}_{\mu_2})$ represents the average value of $P_{\mu_1}(R_iR_{i+1})(P_{\mu_2}(R_iR_{i+1}))$ for a DNA sequence.

The features of DACC contain global sequence-order information, and it can be generated via Pse-in-One [17] which includes two generation approaches. The generation steps of DACC feature can be described as follows.

For web server approach, firstly, choose the DNA sequences (PseDAC-General) option, then select DACC in the tab of Mode, and set the value of lag. Secondly, upload a user-defined physicochemical index file called user_property and the values of fifteen dinucleotide physicochemical properties are shown in Table 1. Finally, input or upload the DNA sequence file in FASTA format, click the Submit button, and then you will see the results and you can download them as a text file (Figure 2).

For stand-alone approach, DACC features can be easily generated by using the following command line:

'./acc.py −e user_property −f svm −l +1 3 DNA DACC'

where −e user_property represents the user-defined physicochemical index file, −f svm and −l +1 have the same meaning with the above command line, the parameter lag equals 3, the sequence type is DNA, and the method used is DACC.

2.2.3. Pseudo Dinucleotide Composition (PseDNC). Given a DNA sequence **D** represented as (2), the PseDNC feature vector **D** can be defined as [17]

$$\mathbf{D} = \left[d_1 \;\; d_2 \;\; d_3 \;\; \cdots \;\; d_{16} \;\; d_{16+1} \;\; \cdots \;\; d_{16+\lambda}\right]^{\mathbf{T}}, \quad (5)$$

where

$$d_k = \begin{cases} \dfrac{f_k}{\sum_{i=1}^{16} f_i + w\sum_{j=1}^{\lambda}\theta_j}, & (1 \le k \le 16), \\ \dfrac{w\theta_{k-16}}{\sum_{i=1}^{16} f_i + w\sum_{j=1}^{\lambda}\theta_j}, & (17 \le k \le 16+\lambda), \end{cases} \quad (6)$$

where f_k $(1 \le k \le 16)$ represents the normalized frequency of dinucleotides along the DNA sequence; w $(0 \le w \le 1)$ represents the weight factor; λ is the top counted tiers of the correlation in a DNA, θ_j $(1 \le j \le \lambda)$ measures the correlation between dinucleotides in the DNA, which is defined as

$$\begin{aligned} \theta_1 &= \frac{1}{L-2}\sum_{i=1}^{L-2}\Theta\left(R_iR_{i+1}, R_{i+1}R_{i+2}\right), \\ \theta_2 &= \frac{1}{L-3}\sum_{i=1}^{L-3}\Theta\left(R_iR_{i+1}, R_{i+2}R_{i+3}\right), \\ \theta_3 &= \frac{1}{L-4}\sum_{i=1}^{L-4}\Theta\left(R_iR_{i+1}, R_{i+3}R_{i+4}\right), \\ &\vdots \\ \theta_\lambda &= \frac{1}{L-1-\lambda}\sum_{i=1}^{L-1-\lambda}\Theta\left(R_iR_{i+1}, R_{i+\lambda}R_{i+\lambda+1}\right), \\ &\qquad (\lambda < L), \end{aligned} \quad (7)$$

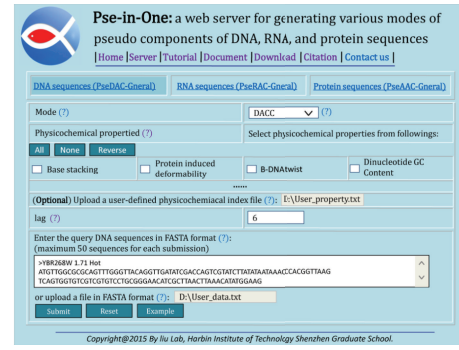


Figure 2: An example of the DACC features' generation by using Pse-in-One.

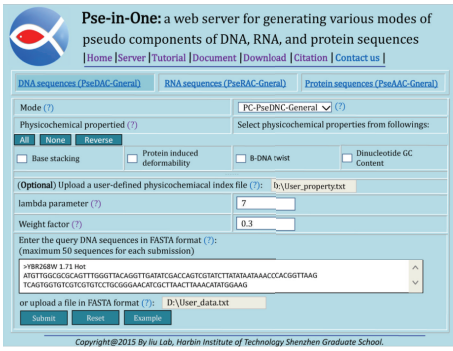


Figure 3: An example of the PseDNC features' generation by using Pse-in-One.

where

$$\Theta\left(R_iR_{i+1}, R_jR_{j+1}\right) = \frac{1}{\mu}\sum_{\mu=1}^{\mu}\left[P_\mu\left(R_iR_{i+1}\right) - P_\mu\left(R_jR_{j+1}\right)\right]^2, \quad (8)$$

where μ represents the indices of the dinucleotide property; $P_\mu(R_iR_{i+1})(P_\mu(R_jR_{j+1}))$ represents the value of dinucleotide $R_iR_{i+1}(R_jR_{j+1})$ at position $i(j)$ for the dinucleotide property index μ.

Pseudo dinucleotide composition (PseDNC) [17] not only incorporates the local nucleic acid composition information and the global or long range information along the DNA sequences, but also incorporates the dinucleotide properties into feature vectors.

For web server approach, the generation steps of the feature vectors are similar to those of the DACC's. For web server approach, an example is shown in Figure 3.

For stand-alone approach, the command line is

'./pse.py −e user_property −f svm −l +1 7 0.3 DNA PseDNC'

where −e user_property, −f svm, and −l +1 have the same meaning with the above command line, lambda equals 7, the value of weight equals 0.3, the sequence type is DNA, and the method used is PseDNC.

The meanings of all the parameters for these scripts are described in [17].

2.3. Support Vector Machine (SVM). Support Vector Machine (SVM) is a kind of algorithm based on statistical learning theory proposed by Vapnik [20–22], which has been widely used for many bioinformatics tasks [23–27].

In the current study, the LIBSVM package version 3.21 [18] has been employed. The SVM parameters, the kernel width parameter γ and the regularization parameter C, were optimized via the grid tool provided by LIBSVM [18].

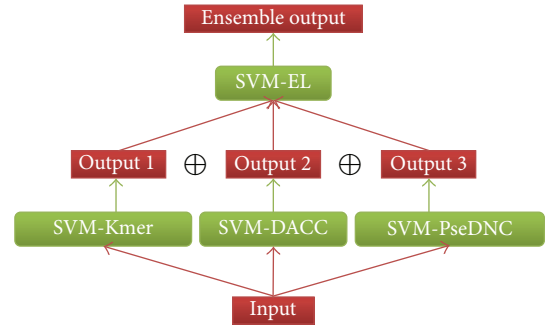


Figure 4: The basic framework for an ensemble classifier.

In the current study, three basic predictors are proposed, including SVM-Kmer, SVM-DACC, and SVM-PseDNC. The values of SVM-Kmer's parameters are shown as follows:

$$C = 2^7, \quad \gamma = 2, \quad k = 6. \quad (9)$$

The values of SVM-DACC's parameters are shown as follows:

$$C = 2^3, \quad \gamma = 2^{-3}, \quad \text{lag} = 6. \quad (10)$$

The values of SVM-PseDNC's parameters are shown as follows:

$$C = 2^{13}, \quad \gamma = 2^3, \quad \lambda = 7, \quad w = 0.3. \quad (11)$$

2.4. Ensemble Learning. In machine learning, ensemble learning is the process by which multiple classifiers are constructed and combined based on the same dataset to obtain a better performance than a single classifier [28, 29] and existing popular multiobjective optimization evolutionary algorithms can be used for ensemble learning [30, 31]. Ensemble classifier also performed well in several bioinformatics problems. In the current study, the basic framework for an ensemble classifier is illustrated in Figure 4. The final results are obtained by fusing three individual classifier outcomes, as illustrated below.

Suppose the ensemble classifier $\mathbb{C}$ is defined as

$$\mathbb{C} = \mathbb{C}_1 \oplus \mathbb{C}_2 \oplus \mathbb{C}_3, \quad (12)$$

where $\mathbb{C}_1$ represents the classifier SVM-Kmer, $\mathbb{C}_2$ represents the classifier SVM-DACC, and $\mathbb{C}_3$ represents the classifier SVM-PseDNC. The symbol $\oplus$ denotes the fusing operator.

Therefore, the process of the ensemble classifier can be formulated as follows:

$$R_j = \frac{1}{3} P_i\left(\mathbf{S}, L_j\right), \quad \left(i = 1, 2, 3;\ j = 1, 2\right), \tag{13}$$

where L_1 is the set only containing recombination hotspots and L_2 is the set of recombination coldspots. $P_i(\mathbf{S}, L_j)$ is the probability for DNA sequence $\mathbf{S}$ which belongs to category L_i obtained by the ith basic classifier.

Thus, which category the query DNA $\mathbf{S}$ belongs to is to be determined by using its average probability calculated by (13); that is, suppose that

$$R_\mu = \text{Max}\left\{R_1, R_2\right\}, \tag{14}$$

where the operator max represents selecting a lager value in the brackets, and the subscript μ represents the query DNA $\mathbf{S}$ belonging to category L_μ.

2.5. Criteria for Performance Evaluation. The prediction results can be divided into true positive (TP), false negative (FN), false positive (FP), and true negative (TN) [32]. In the current study, jackknife test [33–37] was employed and four kinds of evaluation indexes were adopted, including Sensitivity (Se), Specificity (Sp), Accuracy (Acc), and Matthew's Correlation Coefficient (Mcc). They are described as

$$\begin{aligned}
\text{Se} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \\
\text{Sp} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%, \\
\text{Acc} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \\
\text{Mcc} &= \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}.
\end{aligned} \tag{15}$$

3. Results and Discussion

3.1. Performance of the Three Basic Classifiers. As an inherent property, sequence-order is important for the classification of DNA sequences. So, three basic methods based on sequence-order information are adopted to identify recombination hotspots/coldspots. Table 2 shows the performance of the three methods. According to the table, we can see that SVM-DACC and SVM-PseDNC outperform SVM-Kmer on the prediction accuracy index. The main reason is that SVM-Kmer is only based on local sequence-order information, while both of SVM-DACC and SVM-PseDNC also contain global sequence-order information.

3.2. The Performance of the Three Basic Predictors Can Be Further Improved by Using Ensemble Learning. Based on the analysis above, we have proposed three basic predictors for identifying recombination hotspots/coldspots. These methods capture DNA information from different aspects.

Table 2: Results on benchmark dataset for different predictors proposed in the current study.

Predictor	Test method	Se (%)	Sp (%)	Acc (%)	MCC
SVM-Kmer[a]	Jackknife	75.92	86.29	81.59	0.628
SVM-DACC[b]	Jackknife	76.12	87.99	82.61	0.649
SVM-PseDNC[c]	Jackknife	72.04	90.69	82.24	0.644
SVM-EL	Jackknife	76.33	88.33	82.89	0.654

[a]The parameters used are $k = 6$ for SVM-Kmer and $C = 2^7$ and $\gamma = 2$ for LIBSVM [18].
[b]The parameters used are lag $= 6$ for SVM-DACC and $C = 2^3$ and $\gamma = 2^{-3}$ for LIBSVM [18].
[c]The parameters used are $\lambda = 7$ and $w = 0.3$ for SVM-PseDNC and $C = 2^{13}$ and $\gamma = 2^3$ for LIBSVM [18].

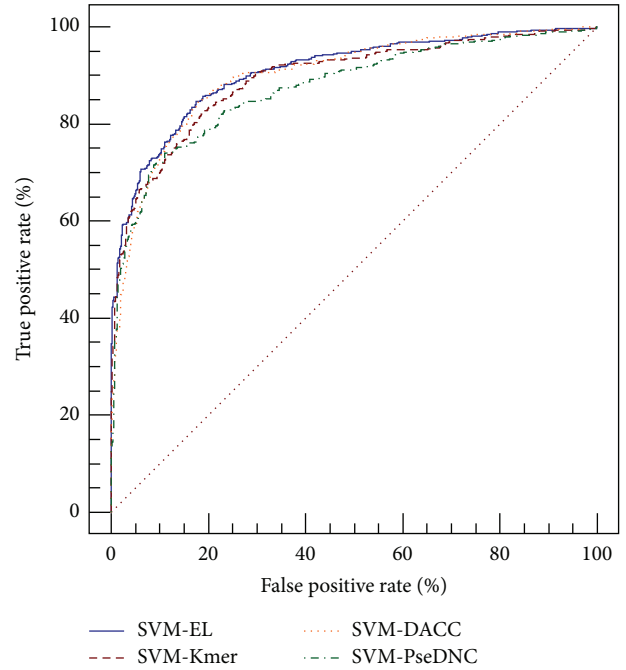


Figure 5: The comparison of different predictors for hotspots/coldspots identification. The areas under ROC curves (AUC) of SVM-EL, SVM-DACC, SVM-Kmer, and SVM-PseDNC are 0.91, 0.90, 0.89, and 0.87, respectively.

Therefore, we presented a complementary method SVM-EL which can fuse these basic methods to improve the prediction performance. The performance of SVM-EL is shown in Table 2, from which we can see that SVM-EL outperforms the three basic methods. Besides, the corresponding receiver operating characteristic (ROC) curves of the four classifiers were drawn in Figure 5. AUC, the area under the ROC curve, is often used to indicate the performance of a classifier: the larger the value, the better the classifier.

As shown in Figure 5, the predictor SVM-EL showed the top performance, outperforming three basic methods: SVM-Kmer, SVM-DACC, and SVM-PseDNC.

3.3. Comparison with Other Related Predictors. Two state-of-the-art methods, IDQD [10] and iRSpot-PseDNC, were

Table 3: Results on benchmark dataset for different predictors.

Predictor	Test method	Se (%)	Sp (%)	Acc (%)	MCC
IDQD[a]	5-fold	79.40	81.00	80.30	0.603
iRSpot-PseDNC[b]	Jackknife	73.06	89.49	82.04	0.638
SVM-EL	Jackknife	76.33	88.33	82.89	0.654

[a]From Liu et al. [10].
[b]From Chen et al. [12].

selected to compare with the proposed SVM-EL. Table 3 shows the results of various methods on the benchmark dataset.

According to Table 3, we can see that SVM-EL outperforms the other methods. The main reason is that IDQD and SVM-Kmer only consider local sequence-order information, and iRSpot-PseDNC, SVM-DACC, and SVM-PseDNC improved them by incorporating global sequence-order information. However, SVM-EL not only incorporates the local nucleic acid information, but also incorporates the global information. Therefore, we conclude that SVM-EL would be a useful tool for hotspots/coldspots identification.

4. Conclusion

In this article, we proposed a predictor called SVM-EL for yeast hotspot/coldspot identification, which combines Support Vector Machine (SVM) with Ensemble Learning (EL). The approach combined with different predictors trained by different features contributes to the improvement of prediction accuracy. SVM-EL is trained by different features, including basic kmer (Kmer), dinucleotide-based auto-cross covariance (DACC), and pseudo dinucleotide composition (PseDNC). All these features can be generated by Pse-in-One [17], which is a powerful web server for generating various DNA, RNA, or protein features. It also provides a stand-alone version to users, which is easy to use. Via jackknife test, it was observed that the predictor outperforms other predictors. In the future, we will consider using other approaches for yeast hotspot/coldspot identification, such as bioinspired computing models [38–45].

Competing Interests

The authors declare no competing financial interests.

Authors' Contributions

Bingquan Liu conceived the study and designed the experiments and participated in designing the study, drafting the manuscript, and performing the statistical analysis. Yumeng Liu participated in coding the experiments and drafting the manuscript. Dong Huang participated in performing the statistical analysis. All authors read and approved the final manuscript. Bingquan Liu and Yumeng Liu contributed equally to this paper.

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (863 Program) (2015AA015405), the National Natural Science Foundation of China (nos. 61300112, 61573118, 61272383, and 61572151), the Natural Science Foundation of Guangdong Province (2014A030313695), Guangdong Natural Science Funds for Distinguished Yong Scholars (2016A030306008), and Scientific Research Foundation in Shenzhen (Grant no. JCYJ20150626110425228).

References

[1] A. Lynn, T. Ashley, and T. Hassold, "Variation in human meiotic recombination," *Annual Review of Genomics and Human Genetics*, vol. 5, pp. 317–349, 2004.

[2] C. C. A. Spencer, P. Deloukas, S. Hunt et al., "The influence of recombination on human genetic diversity," *PLoS Genetics*, vol. 2, no. 9, article e148, 2006.

[3] N. Galtier, G. Piganeau, D. Mouchiroud, and L. Duret, "GC-content evolution in mammalian genomes: the biased gene conversion hypothesis," *Genetics*, vol. 159, no. 2, pp. 907–911, 2001.

[4] M. J. Lercher and L. D. Hurst, "Human SNP variability and mutation rate are higher in regions of high recombination," *Trends in Genetics*, vol. 18, no. 7, pp. 337–340, 2002.

[5] F. Baudat and A. Nicolas, "Clustering of meiotic double-strand breaks on yeast chromosome III," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, no. 10, pp. 5213–5218, 1997.

[6] S. Klein, D. Zenvirth, V. Dror, A. B. Barton, D. B. Kaback, and G. Simchen, "Patterns of meiotic double-strand breakage on native and artificial yeast chromosomes," *Chromosoma*, vol. 105, no. 5, pp. 276–284, 1996.

[7] D. Zenvirth, T. Arbel, A. Sherman, M. Goldway, S. Klein, and G. Simchen, "Multiple sites for double-strand breaks in whole meiotic chromosomes of *Saccharomyces cerevisiae*," *The EMBO Journal*, vol. 11, no. 9, pp. 3441–3447, 1992.

[8] E. Mancera, R. Bourgon, A. Brozzi, W. Huber, and L. M. Steinmetz, "High-resolution mapping of meiotic crossovers and non-crossovers in yeast," *Nature*, vol. 454, no. 7203, pp. 479–485, 2008.

[9] J. L. Gerton, J. DeRisi, R. Shroff, M. Lichten, P. O. Brown, and T. D. Petes, "Global mapping of meiotic recombination hotspots and coldspots in the yeast Saccharomyces cerevisiae," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 21, pp. 11383–11390, 2000.

[10] G. Liu, J. Liu, X. Cui, and L. Cai, "Sequence-dependent prediction of recombination hotspots in *Saccharomyces cerevisiae*," *Journal of Theoretical Biology*, vol. 293, pp. 49–54, 2012.

[11] R. Wang, Y. Xu, and B. Liu, "Recombination spot identification Based on gapped k-mers," *Scientific Reports*, vol. 6, article 23934, 2016.

[12] W. Chen, P.-M. Feng, H. Lin, and K.-C. Chou, "IRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition," *Nucleic Acids Research*, vol. 41, no. 6, article e68, 2013.

[13] B. Liu, F. Liu, L. Fang, X. Wang, and K.-C. Chou, "repDNA: a Python package to generate various modes of feature vectors for

DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects," *Bioinformatics*, vol. 31, no. 8, pp. 1307–1309, 2015.

[14] Q. Dong, S. Zhou, and J. Guan, "A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation," *Bioinformatics*, vol. 25, no. 20, pp. 2655–2662, 2009.

[15] W. Chen, X. Zhang, J. Brooker, H. Lin, L. Zhang, and K.-C. Chou, "PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions," *Bioinformatics*, vol. 31, no. 1, pp. 119–120, 2015.

[16] B. Liu, R. Long, and K.-C. Chou, "iDHS-EL: identifying DNase I hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework," *Bioinformatics*, vol. 32, no. 16, pp. 2411–2418, 2016.

[17] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K.-C. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. 1, pp. W65–W71, 2015.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, article 27, 2011.

[19] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[20] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[21] Y. Wu and S. Krishnan, "Combining least-squares support vector machines for classification of biomedical signals: a case study with knee-joint vibroarthrographic signals," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 23, no. 1, pp. 63–77, 2011.

[22] B. Liu, L. Fang, R. Long, X. Lan, and K. Chou, "iEnhancer-2L: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 32, no. 3, pp. 362–369, 2016.

[23] J. Chen, X. Wang, and B. Liu, "IMiRNA-SSF: improving the identification of MicroRNA precursors by combining negative sets with different distributions," *Scientific Reports*, vol. 6, article 19062, 2016.

[24] B. Liu, D. Zhang, R. Xu et al., "Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection," *Bioinformatics*, vol. 30, no. 4, pp. 472–479, 2014.

[25] Q. Zou, Y. Mao, L. Hu, Y. Wu, and Z. Ji, "miRClassify: an advanced web server for miRNA family classification and annotation," *Computers in Biology and Medicine*, vol. 45, no. 1, pp. 157–160, 2014.

[26] W. Chen and H. Lin, "Prediction of midbody, centrosome and kinetochore proteins based on gene ontology information," *Biochemical and Biophysical Research Communications*, vol. 401, no. 3, pp. 382–384, 2010.

[27] D. Li, Y. Ju, and Q. Zou, "Protein folds prediction with hierarchical structured SVM," *Current Proteomics*, vol. 13, no. 2, pp. 79–85, 2016.

[28] B. Liu, S. Wang, and X. Wang, "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation," *Scientific Reports*, vol. 5, Article ID 15479, 2015.

[29] M. Wu, L. Liao, X. Luo et al., "Analysis and classification of stride patterns associated with children development using gait signal dynamics parameters and ensemble learning algorithms," *BioMed Research International*, vol. 2016, Article ID 9246280, 8 pages, 2016.

[30] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.

[31] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[32] Y. Wu, P. Chen, X. Luo et al., "Quantification of knee vibroarthrographic signal irregularity associated with patellofemoral joint cartilage pathology based on entropy and envelope amplitude measures," *Computer Methods and Programs in Biomedicine*, vol. 130, pp. 1–12, 2016.

[33] B. Liu, J. Xu, X. Lan et al., "iDNA-Prot—dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition," *PLoS ONE*, vol. 9, no. 9, Article ID e106691, 2014.

[34] W. Chen and H. Lin, "Identification of voltage-gated potassium channel subfamilies from sequence information using support vector machine," *Computers in Biology and Medicine*, vol. 42, no. 4, pp. 504–507, 2012.

[35] W. Chen, P. Feng, and H. Lin, "Prediction of ketoacyl synthase family using reduced amino acid alphabets," *Journal of Industrial Microbiology and Biotechnology*, vol. 39, no. 4, pp. 579–584, 2012.

[36] B. Liu, J. Chen, and X. Wang, "Application of learning to rank to protein remote homology detection," *Bioinformatics*, vol. 31, no. 21, pp. 3492–3498, 2014.

[37] B. Liu, L. Fang, F. Liu, X. Wang, J. Chen, and K.-C. Chou, "Identification of real microRNA precursors with a pseudo structure status composition approach," *PLoS ONE*, vol. 10, no. 3, Article ID e0121501, 2015.

[38] T. Song, J. Xu, and L. Pan, "On the universality and non-universality of spiking neural P systems with rules on synapses," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 960–966, 2015.

[39] T. Song and L. Pan, "Spiking neural P systems with request rules," *Neurocomputing*, vol. 193, pp. 193–200, 2016.

[40] X. Wang, T. Song, F. Gong, and P. Zheng, "On the computational power of spiking neural P systems with self-organization," *Scientific Reports*, vol. 6, Article ID 27624, 2016.

[41] X. Zhang, L. Pan, and A. Păun, "On the universality of axon P systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2816–2829, 2015.

[42] X. Zeng, L. Xu, X. Liu, and L. Pan, "On languages generated by spiking neural P systems with weights," *Information Sciences*, vol. 278, pp. 423–433, 2014.

[43] X. Zeng, X. Zhang, T. Song, and L. Pan, "Spiking neural P systems with thresholds," *Neural Computation*, vol. 26, no. 7, pp. 1340–1361, 2014.

[44] X. Zhang, Y. Liu, B. Luo, and L. Pan, "Computational power of tissue P systems for generating control languages," *Information Sciences*, vol. 278, pp. 285–297, 2014.

[45] B. Liu, S. Wang, Q. Dong, S. Li, and X. Liu, "Identification of DNA-binding proteins by combining auto-cross covariance transformation and ensemble learning," *IEEE Transactions on NanoBioscience*, vol. 15, no. 4, pp. 328–334, 2016.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 5313050, 7 pages
http://dx.doi.org/10.1155/2016/5313050

Research Article

Statistical Approaches for the Construction and Interpretation of Human Protein-Protein Interaction Network

Yang Hu,[1] Ying Zhang,[2] Jun Ren,[1] Yadong Wang,[3] Zhenzhen Wang,[4] and Jun Zhang[2]

[1]*School of Life Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[2]*Department of Pharmacy, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin 150088, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[4]*College of Bioinformatics Science and Technology, Harbin Medical University, Harbin 150081, China*

Correspondence should be addressed to Yadong Wang; ydwang@hit.edu.cn, Zhenzhen Wang; wangzz@ems.hrbmu.edu.cn, and Jun Zhang; zhangjun13902003@163.com

Received 3 June 2016; Revised 23 July 2016; Accepted 1 August 2016

Academic Editor: Qin Ma

The overall goal is to establish a reliable human protein-protein interaction network and develop computational tools to characterize a protein-protein interaction (PPI) network and the role of individual proteins in the context of the network topology and their expression status. A novel and unique feature of our approach is that we assigned confidence measure to each derived interacting pair and account for the confidence in our network analysis. We integrated experimental data to infer human PPI network. Our model treated the true interacting status (yes versus no) for any given pair of human proteins as a latent variable whose value was not observed. The experimental data were the manifestation of interacting status, which provided evidence as to the likelihood of the interaction. The confidence of interactions would depend on the strength and consistency of the evidence.

1. Introduction

Individual proteins cannot perform their biological functions by themselves, and actually they need to perform their functions in the biological process through interacting with other proteins [1]. Usually the interaction between two proteins means either they perform a biological function corporately or there is physical direct contact between them [2]. Most of the important molecular processes in cell, such as DNA replication, need to be performed by a large number of protein complexes. And these complexes are made up by the interactions between proteins. The study of PPIs is also considered to be a central problem in proteomics for living cells. Due to the dynamic interaction between proteins, the impact of surrounding environment should also be taken into account. The study of human PPI network can help to enhance the understanding of the disease but also provide a theoretical foundation for finding new treatment.

With the continuous progress and development of high-throughput experimental technology, more and more large quantities of interactions between human proteins had been confirmed by a variety of experimental methods. And many kinds of biological interaction networks have been investigated [3–7]. However, current high-throughput experimental techniques also indicated the shortcomings of high error; not only might the different experimental methods induce different experimental results, but also even different research groups using the same experimental method could not guarantee the exact same result. Therefore, it was urgent to integrate the data from different biological experiments, and even different species, to construct a highly credible network of PPIs. So in this paper, a Bayesian hierarchical model of human PPI network was constructed with a variety of sources of protein interaction data. Meanwhile, a Monte Carlo expectation maximization algorithm was used to estimate the parameters of the model. Then the confidence of protein interaction relationship was calculated based on Bayesian model, and human PPI network with high-confidence level could be obtained.

Thereafter, the role of intrinsic disordered proteins (IDPs) was investigated in the high-confidence PPI network. First of all, different functional modules were obtained through

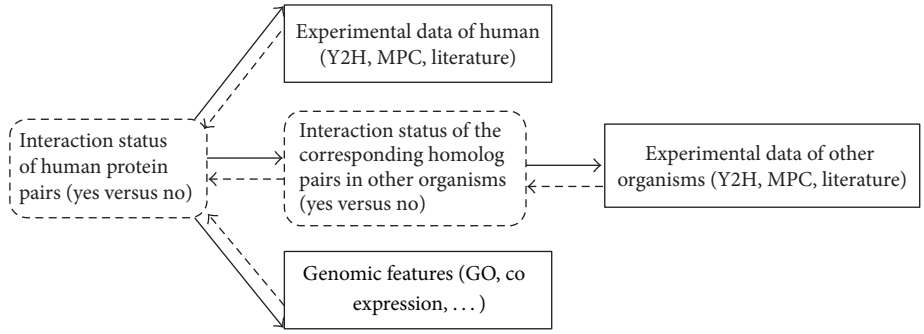


Figure 1: Overall scheme to construct the human protein-protein interaction network. The interaction status of a given pair of human proteins and their homolog in other organisms are unobserved (dashed box) and the experimental data and genomic features are observed evidence (solid boxes). Solid arrows represent model hierarchy and dashed arrows represent inference steps.

Table 1: Data sets or databases used to construct the human protein-protein interaction network.

Method	Organism	Reference
Y2H	Human	Stelzl et al. [8]
Y2H	Human	Rual et al. [9]
MPC	Human	Ewing et al. [10]
Literature	Human	HPRD [11], http://www.hprd.org/
Y2H	Yeast	Ito et al. [12]
Y2H	Yeast	Uetz et al. [13]
MPC	Yeast	Gavin et al. [14]
MPC	Yeast	Ho et al. [15]
MPC	Yeast	Gavin et al. [16]
MPC	Yeast	Krogan et al. [17]
Literature	Multiple	IntAct [18], http://www.ebi.ac.uk/intact/
Literature	Multiple	MIPS [19], http://mips.gsf.de/proj/ppi/
Multiple	Multiple	DIP [20], http://dip.doe-mbi.ucla.edu

clustering of high-confidence PPI network based on the network topology structure. Then we found the functional modules which were significantly correlated with intrinsically disordered proteins and analysed the effect of IDPs in these functional modules, while searching for the associations between these functional modules and diseases.

2. Materials and Methods

2.1. Data Collection. In Table 1, we show the experimental data that will be used for the construction of the human PPI network [8–20]. Note that the literature or text mining approach represents most of the low-throughput experimental studies of individual protein-protein interaction. It is possible that the result from the same experiment will be recorded in multiple databases. We will eliminate this type of redundancy. It should be emphasized that the MPC experiments provide result in the format of protein complexes instead of pair-wise protein-protein interactions. Since proteins located in the same complex might not interact with one another directly, we will account for this factor in our model.

2.2. Statistical Modeling of Various Data Sources. The overall scheme of our approach is illustrated in Figure 1. We consider an empirical Bayes approach to integrate various sources of evidence. Let Z_{ij} be the binary indicator such that $Z_{ij} = 1$ means that human proteins i and j have a direct physical interaction and it is 0 otherwise. Hence, Z_{ij} is the true interacting status that is not observed. To infer Z_{ij}, we consider individual model for each type of observed data and integrate the evidence to compute the probability of $Z_{ij} = 1$.

2.2.1. Human Y2H Data. It has been found that there are a number of mechanisms that can lead to the expression of the reporter gene in a Y2H experiment, which means that an observed interaction might not necessarily mean a true interaction. In our model, we consider the following mechanisms: (a) true interaction; (b) self-activation; and (c) unknown process. Let Y_{ij} be the binary indicator such that $Y_{ij} = 1$ if proteins i and j are observed to interact in a Y2H experiment and it is 0 otherwise. Then $Y_{ij} = 1$ only if at least one of the three above mechanisms is functional. Let $X_i = 1$ if protein i is a self-activation protein and let it be 0 otherwise. We define

$$\alpha_I = \Pr\left[a \text{ is functional} \mid Z_{ij} = 1\right], \tag{1}$$

$$\alpha_S = \Pr\left[b \text{ is functional} \mid X_i + X_j > 0\right], \tag{2}$$

$$\alpha_U = \Pr\left[c \text{ is functional}\right]. \tag{3}$$

Then we have

$$\begin{aligned}&\Pr\left[Y_{ij} = 1 \mid Z, X\right]\\ &\quad = 1 - \left(1 - \alpha_I\right)^{Z_{ij}} \left(1 - \alpha_S\right)^{X_i + X_j} \left(1 - \alpha_U\right).\end{aligned} \tag{4}$$

2.2.2. Human MPC Data. MPC experiment reveals protein complexes instead of individual pairwise PPI. We say protein B is an n-step neighbour of protein A if the shortest path between A and B in the PPI network is of length n. We conjecture that the bait will mostly fish out its 1-step neighbours, and 2-step neighbours and distant proteins (at least three

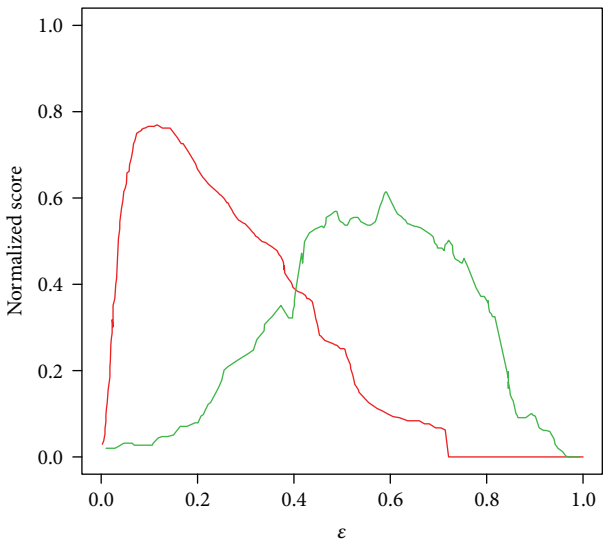


FIGURE 2: The optimization of Q_N and Q_S for different ε. Red line and green line correspond to Q_N and Q_S separately.

step-away) are occasionally observed. Hence, we define the following parameters for the bait proteins:

$$\begin{aligned} &\Pr\left[\text{1-step neighbour is observed}\right] = \psi_1, \\ &\Pr\left[\text{2-step neighbour is observed}\right] = \psi_2. \end{aligned} \tag{5}$$

Let C_k be the set of proteins in a complex corresponding to bait protein k. Denote by $n_k^{(1)}$, $n_k^{(2)}$ the set of 1-step and 2-step neighbours of the bait protein k under a given value of Z. Then the probability of observing C_k can be written as follows:

$$\begin{aligned} &\Pr\left[C_k \mid Z\right] \\ &= \psi_1^{\left|n_k^{(1)} \cap C_k\right|} \left(1-\psi_1\right)^{\left|n_k^{(1)} \setminus C_k\right|} \psi_2^{\left|n_k^{(2)} \cap C_k\right|} \left(1-\psi_2\right)^{\left|n_k^{(2)} \setminus C_k\right|}, \end{aligned} \tag{6}$$

where $|\cdot|$ is the function that maps a set to its size.

2.2.3. Literature Data on Human PPI. Let L_{ij} be the interaction status of proteins i and j reported. We will account for the false positive rate ($\gamma_{0,k}$) and false negative rate ($\gamma_{0,k}$):

$$\Pr\left[H_{ij} = 1 \mid Z_{ij}\right] = \gamma_1^{Z_{ij}} \gamma_0^{1-Z_{ij}}. \tag{7}$$

2.2.4. Data from Other Organisms. We will also collect (Y^*, C^*) from other organisms with corresponding unobserved variables denoted by (Z^*, X^*). Similar models can be used to model (Y, C, L) for inference of (Z^*, X^*). To connect (Z^*, X^*) to (Z, X), we consider the following models:

$$\begin{aligned} &\Pr\left[Z^*_{i'j'} = 1 \mid Z_{ij}\right] \\ &\quad = \left[\Delta_1\left(J_{ii',jj'}; \phi_1\right)\right]^{Z_{ij}} \left[\Delta_0\left(J_{ii',jj'}; \phi_0\right)\right]^{1-Z_{ij}}, \\ &\Pr\left[X^*_{i'} = 1 \mid X_i\right] = \left[\Omega_1\left(I_{ii'}; \lambda_1\right)\right]^{X_i} \left[\Omega_0\left(I_{ii'}; \lambda_0\right)\right]^{1-X_i}, \end{aligned} \tag{8}$$

where $J_{ii',jj'}$ is the joint sequence identity between i and i' and between j and j' and $I_{ii'}$ is sequence identity between i and i'; Δ_1, Δ_0, Ω_1, and Ω_0 are functions of the joint or individual sequence identities with parameters ϕ_1, ϕ_0, λ_1, and λ_0, which can be modeled by parametric structure.

2.3. Construction of Hierarchical Bayesian Model. So far we have introduced the distribution models for the experimental data and genomic features that are conditional on the values of Z and X. To finish the model, we also need to specify the distributions of Z and X, which can be modeled with independent Bernoulli distributions:

$$\begin{aligned} &\Pr\left(Z_{ij} = 1\right) = \rho, \\ &\Pr\left(X_i = 1\right) = r. \end{aligned} \tag{9}$$

With the observed data and the unobserved variables, we can infer the posterior probability of Z using the EM algorithm. Note that there are multiple organisms and multiple data sets for some of the organisms. Different parameters will be used to account for difference in the data.

As illustrated in (10), the complete log likelihood function of our model can be expanded below, and the factor of (10) can be substituted by (3)~(9):

$$\begin{aligned} L_C(\Psi) &= f\left(H, Y, W, Z, X, L, Y^*, W^*, Z^*, X^* \mid \theta\right) = f\left(H \mid Z, \theta\right) f\left(Y \mid Z, X, \theta\right) f\left(W \mid Z, \theta\right) f\left(L \mid Z, \theta\right) \\ &\cdot f\left(Y^* \mid Z^*, X^*, \theta\right) f\left(W^* \mid Z^*, \theta\right) f\left(Z^*, X^* \mid Z, X, \theta\right) f\left(Z, X \mid \theta\right) = \prod_{(i,j)\in S_H} f\left(H_{ij} \mid Z_{ij}, \theta\right) \\ &\cdot \prod_{(i,j)\in S_Y} \left[\prod_{t=1}^{r_{ij}} f\left(Y_{ij}^{(t)} \mid Z_{ij}, X_i, X_j, \theta\right)\right] \prod_{(i,j)\in S_M} \left[\prod_{t=1}^{e_{ij}} f\left(W_{ij}^{i(t)} \mid Z_{ij}, \theta\right) \sum_{t=1}^{e_{ji}} f\left(W_{ij}^{j(t)} \mid Z_{ij}, \theta\right)\right] \\ &\cdot \prod_{(i,j)\in S_{Y^*}} \left[\prod_{t=1}^{r^*_{ij}} f\left(Y_{ij}^{(t)*} \mid Z^*_{ij}, X^*_i, X^*_j, \theta\right)\right] \prod_{(i,j)\in S^*_M} \left[\prod_{t=1}^{e^*_{ij}} f\left(W_{ij}^{i(t)*} \mid Z^*_{ij}, \theta\right) \sum_{t=1}^{e^*_{ji}} f\left(W_{ij}^{j(t)*} \mid Z_{ij}, \theta\right)\right] \prod_{(i,j)\in S_L} f\left(L_{ij} \mid Z_{ij}, \theta\right) \\ &\cdot \prod_{(i,j)\in S^*} f\left(Z^*_{ij} \mid Z_{ij}, \theta\right) \prod_{i\in S^*_Y} f\left(X^*_i \mid X_i, \theta\right) \prod_{(i,j)\in S} f\left(Z_{ij} \mid \theta\right) \prod_{i\in S_Y} f\left(X_i \mid \theta\right) \end{aligned}$$

$$
\begin{aligned}
&= \prod_{(i,j)\in S_Y} \left[1-(1-\alpha_I)^{Z_{ij}}(1-\alpha_S)^{X_i+X_j}(1-\alpha_U)\right]^{Y_{ij}^{+}} \left[(1-\alpha_I)^{Z_{ij}}(1-\alpha_S)^{X_i+X_j}(1-\alpha_U)\right]^{Y_{ij}^{\#}} \\
&\cdot \prod_{(i,j)\in S_H} \left(\gamma_1^{Z_{ij}}\gamma_0^{1-Z_{ij}}\right)^{H_{ij}} \left(1-\gamma_1^{Z_{ij}}\gamma_0^{1-Z_{ij}}\right)^{1-H_{ij}} \prod_{(i,j)\in S_M} \left[\psi_1^{Z_{ij}W_{ij}^{+}}(1-\psi_1)^{Z_{ij}W_{ij}^{\#}} \times \psi_2^{(1-Z_{ij})W_{ij}^{+}}(1-\psi_2)^{(1-Z_{ij})W_{ij}^{\#}}\right] \\
&\cdot \prod_{(i,j)\in S_{Y^*}} \left[1-(1-\alpha_I)^{Z_{ij}^{*}}(1-\alpha_S)^{X_i^{*}+X_j^{*}}(1-\alpha_U)\right]^{Y_{ij}^{+*}} \left[(1-\alpha_I)^{Z_{ij}^{*}}(1-\alpha_S)^{X_i^{*}+X_j^{*}}(1-\alpha_U)\right]^{Y_{ij}^{\#*}} \\
&\cdot \prod_{(i,j)\in S_M^{*}} \left[\psi_1^{Z_{ij}^{*}W_{ij}^{+*}}(1-\psi_1)^{Z_{ij}^{*}W_{ij}^{\#*}} \times \psi_2^{(1-Z_{ij}^{*})W_{ij}^{+*}}(1-\psi_2)^{(1-Z_{ij}^{*})W_{ij}^{\#*}}\right] \prod_{(i,j)\in S_H} \left(\gamma_1^{Z_{ij}}\gamma_0^{1-Z_{ij}}\right)^{H_{ij}} \left(1-\gamma_1^{Z_{ij}}\gamma_0^{1-Z_{ij}}\right)^{1-H_{ij}} \\
&\cdot \prod_{(i,j)\in S_L} \left(\beta_1^{Z_{ij}}\beta_0^{1-Z_{ij}}\right)^{L_{ij}} \left(1-\beta_1^{Z_{ij}}\beta_0^{1-Z_{ij}}\right)^{1-L_{ij}} \prod_{(i,j)\in S} \rho^{Z_{ij}}(1-\rho)^{1-Z_{ij}} \prod_{i\in S_Y} r^{X_i}(1-r)^{1-X_i} \\
&\cdot \prod_{(i,j)\in S^{*}} \left(\phi_1^{Z_{ij}}\phi_0^{1-Z_{ij}}\right)^{Z_{ij}^{*}} \left(1-\phi_1^{Z_{ij}}\phi_0^{1-Z_{ij}}\right)^{1-Z_{ij}^{*}} \prod_{i\in S_Y^{*}} \left(\lambda_1^{X_i}\lambda_0^{1-X_i}\right)^{X_i^{*}} \left(1-\lambda_1^{X_i}\lambda_0^{1-X_i}\right)^{1-X_i^{*}},
\end{aligned}
\tag{10}
$$

where the parameter vector $\theta = \{\rho, r, \alpha_I, \alpha_S, \alpha_U, \psi_1, \psi_2, \gamma_1, \gamma_0, \beta_1, \beta_0, \phi_1, \phi_0, \lambda_1, \lambda_0\}$.

2.4. Monte Carlo Expectation Maximization for Parameter Estimation. In the model, it was not possible to estimate the true value of potential variables and model parameters directly. In order to effectively estimate the potential variables and model parameters, this paper used the Monte Carlo expectation maximization algorithm based on incomplete parameter estimation, as illustrated in Algorithm 1.

In the E-step of Algorithm 1, we use Gibbs sampling to sample (Z, X, Z^*, X^*) from $f(Z, X, Z^*, X^* \mid H, Y, W, L, Y^*, W^*, \widehat{\theta}_0)$ in turn. Repeat the sampling process until the estimations of missing data are obtained. Then in the M-step of Algorithm 1, the parameter vector $\theta = \{\gamma_1, \gamma_0, \alpha_I, \alpha_S, \alpha_U, \beta_1, \beta_0, \phi_1, \phi_0, \lambda_1, \lambda_0\}$ is estimated by Greedy Hill Climbing. Finally the iteration is stopped when diff $>$ 0.01.

3. Results

All the protein names were mapped to the Entrez IDs. Finally we got 32540 proteins, and there were 144603 interactions between these proteins.

3.1. Construction of the Human PPI Network with Reliable Confidence Measure. Four models were established separately using high-throughput Y2H experimental data, high-throughput MPC experimental data, human PPI data, and all the PPI data. The comparisons among these four models were listed in Table 2.

After the estimation of parameter vector θ by Monte Carlo EM, we recalculated the posterior probability of Z, which is $\Pr[Z \mid H, Y, W, L, Y^*, W^*]$, with θ and the observed values H, Y, W, L, Y^*, W^*. And for each pair of PPI, we considered them as reliable confidence interaction if $\Pr[Z_{ij} = 1 \mid H, Y, W, L, Y^*, W^*] > 0.8$. Then we got 48361 PPIs with reliable confidence measure among 23286 proteins.

3.2. Characterization of Network and Roles of IDPs Based on Network Topology. We analysed the role of IDPs in the human PPI networks with reliable confidence measure. A IDP was defined as a protein with continuous intrinsically disorder region whose length was larger than 40 amino acids. And 8735 IDPs were identified from 23286 proteins after predictions.

Firstly, the human PPI network was cut into subnetworks or modules by SCAN. SCAN obtained modules based on the similarity between common neighbors. Then we used modularity and similarity-based modularity as metrics. Modularity is a statistical measure of the quality of network clustering, which is defined as follows:

$$
Q_N = \sum_{s=1}^{N_C} \left[\frac{l_s}{L} - \left(\frac{d_s}{2L}\right)^2\right], \tag{11}
$$

where N_C is the number of clusterings, L is the number of edges, l_s is the number of edges for s_{th} module, and d_s is the degree of all the nodes in s_{th} module. We could obtain the best clustering by optimizing Q_N. And similarity-based modularity is the supplementary for the modularity, which is defined as follows:

$$
Q_S = \sum_{s=1}^{NC} \left[\frac{IS_i}{TS} - \left(\frac{DS_i}{2TS}\right)^2\right]. \tag{12}
$$

As shown in Figure 2, on one hand, the modularity monotonically decreased from the position nearby zero, and it could not be maximized. On the other hand, the similarity-based modularity could be maximized while the threshold ε equals 0.61. Conditional on the $\varepsilon = 0.61$, the reliable human PPI

(1) $i = 0$, initialize the parameters
(2) while (diff > 0.01) {
(3) $j = 0$ // E-Step
(4) while ($j \le T$) {
(5) Sample $X^{(j+1)}$ from $f(Z^{(j)}, X, Z^{*(j)}, X^{*(j)} \mid H, Y, W, L, Y^*, W^*, \widehat{\theta}_i)$
(6) Sample $Z^{(j+1)}$ from $f(Z, X^{(j+1)}, Z^{*(j)}, X^{*(j)} \mid H, Y, W, L, Y^*, W^*, \widehat{\theta}_i)$
(7) Sample $Z^{(j+1)}$ from $f(Z^{(j+1)}, X^{(j+1)}, Z^{*(j)}, X^* \mid H, Y, W, L, Y^*, W^*, \widehat{\theta}_i)$
(8) Sample $Z^{(j+1)}$ from $f(Z^{(j+1)}, X^{(j+1)}, Z^*, X^{*(j+1)} \mid H, Y, W, L, Y^*, W^*, \widehat{\theta}_i)$
(9) $j = j + 1$
(10) }
(11) calculate Q function

$$\widehat{Q}\left(\theta \mid \theta^{(i)}, Y, W, L, Y^*, W^*\right) = \frac{1}{T}\sum_{m=1}^{T} \log L^c\left(\theta \mid Y, W, L, Y^*, W^*, Z^{(m)}, X^{(m)}, Z^{(m)*}, X^{(m)*}\right)$$

(12) // M-Step
(13)

$$\widehat{\rho}^{(i+1)} = \frac{1}{T}\sum_{m=1}^{T}\left(\frac{\sum_{(i,j)\in S} Z_{ij}^{(m)}}{|S|}\right)$$

$$\widehat{r}^{(i+1)} = \frac{1}{T}\sum_{m=1}^{T}\left(\frac{\sum_{i\in S_Y} X_i^{(m)}}{\left\|S_Y\right\|}\right)$$

$$\widehat{\psi}_1^{(i+1)} = \frac{1}{T}\sum_{m=1}^{T}\left(\frac{\sum_{(i,j)\in S_M} Z_{ij}^{(m)}\left(\sum_{k=1}^{e_{ij}} W_{ij}^{ik} + \sum_{k=1}^{e_{ji}} W_{ij}^{jk}\right) + \sum_{(i,j)\in S_M^*} Z_{ij}^{*(m)}\left(\sum_{k=1}^{e_{ij}^*} W_{ij}^{ik*} + \sum_{k=1}^{e_{ji}^*} W_{ij}^{jk*}\right)}{\sum_{(i,j)\in S_M} Z_{ij}^{(m)}\left(e_{ij} + e_{ji}\right) + \sum_{(i,j)\in S_M^*} Z_{ij}^{*(m)}\left(e_{ij}^* + e_{ji}^*\right)}\right)$$

$$\widehat{\psi}_2^{(i+1)} = \frac{1}{T}\sum_{m=1}^{T}\left(\frac{\sum_{(i,j)\in S_M}\left(1 - Z_{ij}^{(m)}\right)\left(\sum_{k=1}^{e_{ij}} W_{ij}^{ik} + \sum_{k=1}^{e_{ji}} W_{ij}^{jk}\right) + \sum_{(i,j)\in S_M^*}\left(1 - Z_{ij}^{*(m)}\right)\left(\sum_{k=1}^{e_{ij}^*} W_{ij}^{ik*} + \sum_{k=1}^{e_{ji}^*} W_{ij}^{jk*}\right)}{\sum_{(i,j)\in S_M}\left(1 - Z_{ij}^{(m)}\right)\left(e_{ij} + e_{ji}\right) + \sum_{(i,j)\in S_M^*}\left(1 - Z_{ij}^{*(m)}\right)\left(e_{ij}^* + e_{ji}^*\right)}\right)$$

(14) $k = 0$;
(15) $\text{change}^0 = 0.01$
(16) $\theta^k = \theta^{(i)} = \left\{\alpha_I^{(i)}, \alpha_S^{(i)}, \alpha_U^{(i)}, \gamma_1^{(i)}, \gamma_0^{(i)}, \beta_1^{(i)}, \beta_0^{(i)}, \phi_1^{(i)}, \phi_0^{(i)}, \lambda_1^{(i)}, \lambda_0^{(i)}\right\}$
(17) while (1) {
(18) $\alpha_I^{k+1} = \arg\max Q\left(\alpha_I, \alpha_S^k, \alpha_U^k\right)$
(19) $\alpha_S^{k+1} = \arg\max Q\left(\alpha_I^{k+1}, \alpha_S, \alpha_U^k\right)$
$\alpha_U^{k+1} = \arg\max Q\left(\alpha_I^{k+1}, \alpha_S^{k+1}, \alpha_U\right)$
(20) $\text{change}^{k+1} = \widehat{Q}\left(\theta^{k+1}\right) - \widehat{Q}\left(\theta^k\right)$
(21) if (abs(change^{k+1}) < abs(change^k/20))
(22) break
(23) $k = k + 1$
(24) }
(25)

$$\text{diff} = \frac{\left|\widehat{Q}\left(\theta^{(i+1)}\right) - \widehat{Q}\left(\theta^{(i)}\right)\right|}{\widehat{Q}\left(\theta^{(i)}\right)}$$

(26) $i = i + 1$
(27) $T = T * 1.1$
(28) }

Algorithm 1: Monte Carlo expectation maximization for parameter estimation.

network was cut into 241 modules. Under the significant level $\alpha = 0.05$, the p value of each module was calculated by the formula below:

$$p\text{-value} = \sum_{i=m}^{n} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}, \tag{13}$$

where N is the number of all the proteins and M is the number of all the IDPs. 33 modules among 241 modules were significantly associated with IDPs.

However, due to the fact that acquisition of functional modules is only dependent on the network topology, we analysed the modules with known diseases. And the overlap of PPI in hela cell and a functional module which was highly related with IDPs was shown in Figure 3. The weight of each side is the posterior probability of the real value Z. If a node with more than 5 neighbours was defined as a hub node in this subnetwork, a total of 69% of the hub nodes were IDPs. It is verified that IDPs were easy to become hub nodes of the protein interaction network due to the flexibility

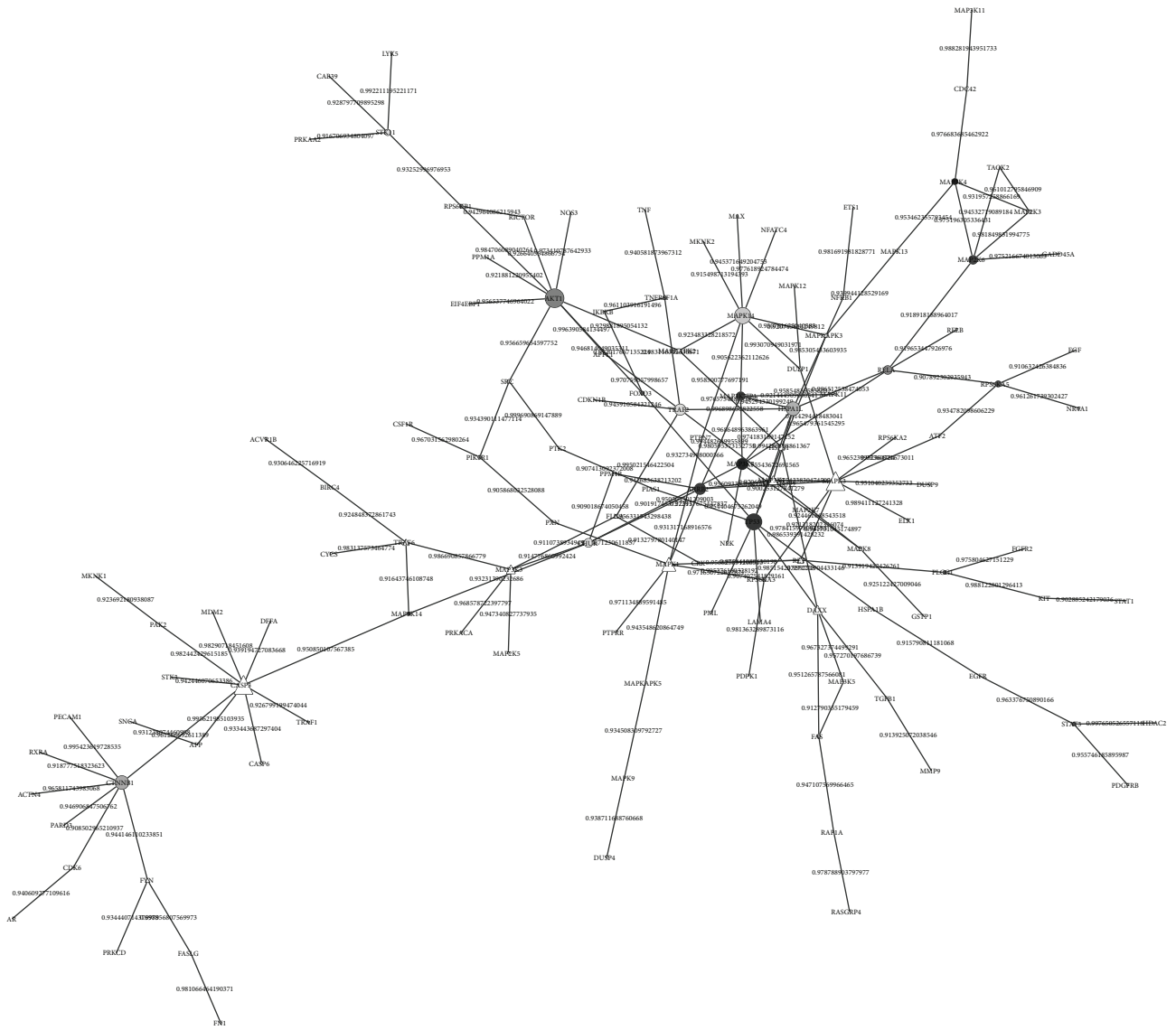


Figure 3: A reliable subnetwork for hela cell. Circles correspond to IDPs. And the degree of grey corresponds to the length of intrinsically disordered region for IDP.

Table 2: Comparison of parameters based on different data.

Parameters	High-throughput Y2H	High-throughput MPC	Human PPI data	All PPI data
ρ	6.8×10^{-3}	1.9×10^{-3}	6.1×10^{-3}	1.4×10^{-2}
r	7.7×10^{-5}	—	5.3×10^{-5}	8.9×10^{-5}
α_I	0.658	—	0.543	0.933
α_S	0.426	—	0.496	0.852
α_U	4.5×10^{-3}	—	9.7×10^{-4}	0.007
ψ_1	—	0.738	0.755	0.809
ψ_2	—	0.623	0.764	0.788

of the structure, revealing an important role of IDPs in the regulation of cervical cancer hela cell.

4. Discussion

Our model is unique and novel in the following perspectives. First, it integrates Y2H and MPC data in a cohesive and unified model that connect the two types of data through the unobserved true status of direct physical interaction Z. Second, the model allows a natural calculation of the confidence of each interacting pair via the posterior probability. This is a critical measurement in downstream analysis and will be accounted for. To our knowledge, no previous study has considered uncertainty in the PPI network analysis.

The inference of the interacting probability involves a large number of latent variables. The combinatorial effects make it impractical to compute the expectation of the missing variables analytically during the E-step. It is likely that various data sets carry different amount of information regarding the true interaction status. Hence, the inference can be

made by appropriately weighing data of various types instead of treating them equally. This can be achieved by setting parameter constrain.

Competing Interests

The authors confirm that there is no conflict of interests related to the content of this article.

Authors' Contributions

Yang Hu and Ying Zhang contributed equally to this work.

References

[1] C. Wu, F. Zhang, X. Li et al., "Composite functional module inference: detecting cooperation between transcriptional regulation and protein interaction by mantel test," *BMC Systems Biology*, vol. 4, article 82, 2010.

[2] Y.-A. Huang, Z.-H. You, X. Chen, K. Chan, and X. Luo, "Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding," *BMC Bioinformatics*, vol. 17, article 184, 2016.

[3] X. Chen, "Predicting lncRNA-disease associations and constructing lncRNA functional similarity network based on the information of miRNA," *Scientific Reports*, vol. 5, Article ID 13186, 2015.

[4] X. Zeng, X. Zhang, and Q. Zou, "Integrative approaches for predicting microRNA function and prioritizing disease-related microRNA using biological interaction networks," *Briefings in Bioinformatics*, vol. 17, no. 2, pp. 193–203, 2016.

[5] Q. Zou, J. Li, Q. Hong et al., "Prediction of microRNA-disease associations based on social network analysis methods," *BioMed Research International*, vol. 2015, Article ID 810514, 9 pages, 2015.

[6] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: a survey," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 55–64, 2016.

[7] F. Zhang, B. Gao, L. Xu et al., "Allele-specific behavior of molecular networks: understanding small-molecule drug response in yeast," *PLoS ONE*, vol. 8, no. 1, Article ID e53581, 2013.

[8] U. Stelzl, U. Worm, M. Lalowski et al., "A human protein-protein interaction network: a resource for annotating the proteome," *Cell*, vol. 122, no. 6, pp. 957–968, 2005.

[9] J.-F. Rual, K. Venkatesan, T. Hao et al., "Towards a proteome-scale map of the human protein-protein interaction network," *Nature*, vol. 437, no. 7062, pp. 1173–1178, 2005.

[10] R. M. Ewing, P. Chu, F. Elisma et al., "Large-scale mapping of human protein-protein interactions by mass spectrometry," *Molecular Systems Biology*, vol. 3, article 89, 2007.

[11] T. S. Keshava Prasad, R. Goel, K. Kandasamy et al., "Human Protein Reference Database—2009 update," *Nucleic Acids Research*, vol. 37, supplement 1, pp. D767–D772, 2009.

[12] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 8, pp. 4569–4574, 2001.

[13] P. Uetz, L. Glot, G. Cagney et al., "A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[14] A.-C. Gavin, P. Aloy, P. Grandi et al., "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[15] Y. Ho, A. Gruhler, A. Heilbut et al., "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[16] A.-C. Gavin, M. Bösche, R. Krause et al., "Functional organization of the yeast proteome by systematic analysis of protein complexes," *Nature*, vol. 415, no. 6868, pp. 141–147, 2002.

[17] N. J. Krogan, G. Cagney, H. Yu et al., "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, 2006.

[18] S. Kerrien, Y. Alam-Faruque, B. Aranda et al., "IntAct—open source resource for molecular interaction data," *Nucleic Acids Research*, vol. 35, no. 1, pp. D561–D565, 2007.

[19] H. W. Mewes, D. Frishman, U. Güldener et al., "MIPS: a database for genomes and protein sequences," *Nucleic Acids Research*, vol. 30, no. 1, pp. 31–34, 2002.

[20] I. Xenarios, Ł. Salwínski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg, "DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions," *Nucleic Acids Research*, vol. 30, no. 1, pp. 303–305, 2002.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 8182416, 15 pages
http://dx.doi.org/10.1155/2016/8182416

Research Article

Robust Individual-Cell/Object Tracking via PCANet Deep Network in Biomedicine and Computer Vision

Bineng Zhong,[1] Shengnan Pan,[1] Cheng Wang,[1] Tian Wang,[1] Jixiang Du,[1] Duansheng Chen,[1] and Liujuan Cao[2]

[1]*Department of Computer Science and Engineering, Huaqiao University, Xiamen, Fujian Province 361021, China*
[2]*School of Information Science and Technology, Xiamen University, China*

Correspondence should be addressed to Bineng Zhong; bnzhong@hqu.edu.cn

Received 1 June 2016; Accepted 17 July 2016

Academic Editor: Qin Ma

Tracking individual-cell/object over time is important in understanding drug treatment effects on cancer cells and video surveillance. A fundamental problem of individual-cell/object tracking is to simultaneously address the cell/object appearance variations caused by intrinsic and extrinsic factors. In this paper, inspired by the architecture of deep learning, we propose a robust feature learning method for constructing discriminative appearance models without large-scale pretraining. Specifically, in the initial frames, an unsupervised method is firstly used to learn the abstract feature of a target by exploiting both classic principal component analysis (PCA) algorithms with recent deep learning representation architectures. We use learned PCA eigenvectors as filters and develop a novel algorithm to represent a target by composing of a PCA-based filter bank layer, a nonlinear layer, and a patch-based pooling layer, respectively. Then, based on the feature representation, a neural network with one hidden layer is trained in a supervised mode to construct a discriminative appearance model. Finally, to alleviate the tracker drifting problem, a sample update scheme is carefully designed to keep track of the most representative and diverse samples during tracking. We test the proposed tracking method on two standard individual cell/object tracking benchmarks to show our tracker's state-of-the-art performance.

1. Introduction

Individual-cell/object tracking is a fundamental problem in computational biology [1–4], drug treatment effects on cancer cells [2], high-content screening [5], and computer vision [6–8]. Therefore, it has attracted much attention due to the potential value for its theoretical challenges and practical applications. Although it has been investigated in the past decades, designing a robust cell/object tracker to cope with appearance changes of a cell/object is still a great challenging task. The appearance changes of a cell/object include intrinsic (e.g., pose changes, motion blur, scale variations, and nonrigid deformation) and extrinsic (e.g., illumination variations, cluttered scenes, and occlusions) factors. Such appearance changes may make a tracker drift away from the cell/object. Moreover, because a large number of manual operations are required in existing cell tracking [9], how to design an accurate and automatic cell tracker with limited manual operations [10] is another challenge.

To capture appearance variations, most state-of-the-art trackers rely on handcrafted features to adaptively construct and update the generative or discriminative models of object appearances (e.g., principal component analysis [1, 2, 11, 12], Hough forest [13], support vector machine [14], and ensemble learning [15, 16]). By using various handcrafted features [16–26], these handcrafted feature-based tracking methods are developed for certain scenarios. Consequently, they are unable to capture the rich sematic information of a target as their generalization is not well. Therefore, they are prone to tracking failure in some challenging conditions.

Recently, deep learning [27–32] has attracted much attention in computational biology, cell biology, and computer

vision. Instead of using handcrafted features, deep learning aims to automatically learn hierarchical feature representation from raw data. With the impressive performance achieved by deep learning on speech recognition [28] and image recognition [29, 31, 32], a few of early researchers [33–39] have applied it to object tracking and achieved competitive performance. However, as only the annotation of a target object in the initial frames is available, the deep learning-based trackers usually use large-scale training data to prelearn deep structure and transfer the pretrained feature representation to the tracking tasks. Consequently, the large-scale pretraining is time-consuming and the pretrained feature representation may be less discriminative for tracking a specific cell/object. Moreover, they may be sensitive to partial occlusion and pose changes due to using a single global bounding box to delineate the entire cell/object.

In this paper, we propose a robust discriminative tracking method which automatically learns feature representation without large-scale pretraining and explicitly handles partial occlusion by fusing a global structure and local details in a cell/object. Specifically, in the initial frames, an unsupervised method is firstly used to learn the abstract feature of a cell/object by exploiting both classic principal component analysis (PCA) algorithms with recent deep learning representation architectures. We use learned PCA eigenvectors as filters and develop a novel algorithm to represent a target by composing of a PCA-based filter bank layer, a nonlinear layer, and a patch-based pooling layer, respectively. Then, based on the feature learned from the above unsupervised method, a neural network with one hidden layer is trained in a supervised mode to construct a discriminative target appearance model. By exploiting the advantage of deep learning architecture, our method is able to learn a generic and hierarchical feature representation while performing more efficiently without large-scale pretraining. Compared with holistic-based models, our method simultaneously maintains holistic and local appearance information and therefore provides a compact representation of the target object. Finally, to alleviate the tracker drifting problem, a simple yet effective sample update scheme is adopted to keep track of the most representative and diverse samples while tracking. The experiments on two standard individual-cell/object tracking benchmarks (i.e., the Mitocheck cell dataset [40] and the online tracking benchmark (OTB) [41]) show that our tracker achieves a promising performance.

The rest of the paper is organized as follows. Section 2 discusses the most related work to ours. The detailed overall framework of our tracking method is described in Section 3. The performance of our tracking method is demonstrated in Section 4. Finally, Section 5 summarizes our findings.

2. Related Work

Much work has been done in the area of cell/object tracking and the comprehensive review is beyond the scope of this paper. Please refer to [3, 4, 6–8] for more complete reviews on cell/object tracking and recent tracking benchmarks. In this section we briefly review some representative works on visual tracking and put our work in a proper context.

2.1. Individual-Cell and Object Tracking with Handcrafted Features. For decades, many tracking methods with handcrafted features have been proposed, which focus on constructing robust cell/object appearance models to handle the inevitable appearance changes of a cell/object. In [17], a mean shift-based tracking method using color histograms is proposed. Li et al. [18] propose a multiple nuclei tracking method with the intensity features for quantitative cancer cell cycle analysis. In [19], Danelljan et al. propose an adaptive color attribute-based tracking method under a coloration filtering framework. In [26], Lou et al. propose an active structured learning-based cell tracking method by combining multiple complementary features, such as position, intensity, and shape. In [12], an incremental principal component analysis-based tracking method is proposed for robust visual tracking. Recently, a variety of low-rank subspaces and sparse representations based tracking methods have been proposed [42–47] for cell/object tracking due to their robustness to occlusion and image noises. Zhong et al. [48] propose a weakly supervised learning-based tracking method, in which multiple complementary trackers are effectively fused to achieve robust tracking results. Zhou et al. [49] propose a similarity fusion-based tracking method, in which multiple features and context structure of unlabeled data are effectively utilized.

Coupled with designing handcrafted features, numerous advanced machine learning methods have been developed to further improve the tracking performances. The typical learning methods include support vector machine (SVM) classifiers [14], structured output SVM [21], online boosting [15, 20], P-N learning [50], multiple instance learning [51], and correlation filters [52–54]. In [55], for improving the tracking performance, Lou et al. incorporate a shape prior into a learning method to segment dense cell nuclei. Dzyubachyk et al. [56] utilize a level set-based method for cell tracking in time-lapse fluorescence microscopy.

Moreover, to explicitly deal with the occlusion problem, several part-based models have been proposed. In [13], Gall et al. propose a part-based voting schema via Hough forests for robust tracking. In [17], online latent structural learning is employed for a part-based object tracking method. However, the part-based tracking methods still rely on low-level features. Although tracking methods with handcrafted features usually produce more accurate results under less complex environments, they may be limited by using handcrafted features which cannot be simply adapted according to the new observed data obtained while tracking.

2.2. Single-Cell and Object Tracking with Deep Learning. Inspired by the success of deep learning in speech and visual recognition tasks [27–32], a few of deep learning-based tracking methods have been recently proposed [33–39] for robust cell/object tracking. In [35], based on a pretrained convolutional neural network, Fan et al. propose a tracking method for human. One of the limitations is that the pretrained convolutional neural network is fixed during the online tracking process. Wang and Yeung [34] propose an autoencoder based tracking method. Instead of using unrelated images

for pretraining, Wang et al. [57] propose a tracking method which prelearns features robust to diverse motion patterns from auxiliary video sequences. However, they only evaluate the method on 10 video sequences. In [58], Li et al. effectively combine multiple convolutional neural networks for robust tracking. Within a particle filtering framework, Carneiro and Nascimento [33] use deep learning architectures to cope with the left ventricle endocardium in ultrasound data. In [36], based on the deep network of VGG, a fully convolutional neural network is proposed for robust tracking. In [37], Hong et al. propose a tracking method by learning discriminative saliency map with convolutional neural network. In [38], Ma et al. fuse the correlation filters and pretrained VGG network for robust tracking. In [39], Nam and Han propose a multidomain convolutional neural network-based tracking method.

However, these tracking methods are time-consuming due to the large-scale pretraining. Moreover, the pretrained feature representation may be less discriminative for tracking specific target objects.

3. The Proposed Individual-Cell and Object Tracking Algorithm

In this section, we develop our discriminative tracking algorithm via a PCANet deep network [32]. Based on a particle filtering framework, the proposed PCANet-based tracking method for individual-cell/object is schematically shown in Algorithm 1.

Specifically, the proposed tracking algorithm works as follows: the target object is manually selected in the first frame by a bounding box. Then, an unsupervised method is used to learn the abstract feature of the target object by exploiting both classic principal component analysis (PCA) algorithms with recent deep learning representation architectures. Furthermore, based on the feature learned from the above unsupervised method, a neural network with one hidden layer is trained in a supervised mode to construct a discriminative object appearance model. Meanwhile, a set of particles with associated weights is initialized within a particle filtering framework. For one incoming video frame t, we first predict each particle using the dynamic model. Then, we compute weights for each particle using the observation model (i.e., the discriminative appearance model). According to the obtained weights, we determine the optimal object state as the particle with the maximum weight and resample particles. Finally, the pretrained feature is updated according to the new observed data. Meanwhile, the discriminative appearance model is also incrementally updated via a simple yet effective sample update scheme which keeps track of the most representative and diverse samples while tracking. The tracking procedure continues in this iterative fashion until the end of video.

Below we give a detailed description about each component of our method.

Algorithm 1. Overview of the proposed PCANet-based tracking method for individual-cell/object is shown below.

Input is as follows:

(1) Get one initialized video frame with ground-truth bounding box on a cell/object.

(2) Pretrain an abstract feature of a cell/object via an unsupervised method.

(3) Build a neural network-based discriminative appearance model for the cell/object based on the feature learned from the above unsupervised method.

(4) Initialize a set of particles with associated weights within a particle filtering framework.

Output is as follows:

(1) Predict each particle using a Gaussian function-based motion model.

(2) Compute weights for each particle using a PCANet-based discriminative appearance model.

(3) Determine the optimal cell/object state as the particle with the maximum weight.

(4) Resample particles based on their corresponding weights.

(5) Update the pretrained feature and the PCANet-based discriminative appearance model according to the newly observed data.

3.1. Particle Filtering. The proposed tracking algorithm is carried out using the particle filtering framework which is a Markov model with hidden state variables. Supposing that we have observations of the target object $Z_t = [z_1, \ldots, z_t]$ up to the tth frame, the hidden state variable x_t is estimated by the well-known two-step iteration (i.e., the prediction and the update steps):

$$p(x_t \mid Z_t) \propto p(z_t \mid x_t) \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid Z_{t-1})\, dx_{t-1}, \tag{1}$$

where $p(x_t \mid x_{t-1})$ is the dynamic (motion) model between two consecutive states and $p(z_t \mid x_t)$ is the observation model which estimates the likelihood of observing z_t at state x_t. The optimal object state x_t^* at time t can be determined by the maximum a posteriori estimation over N samples (particles) at the tth frame by

$$x_t^* = \arg\max_{x_t^i}{}^{i=1,\ldots,N} \left\{ p\left(z_t^i \mid x_t^i\right) p\left(x_t^i \mid x_{t-1}\right) \right\}, \tag{2}$$

where x_t^i is the ith sample of the state x_t and z_t^i is the image observation predicted by x_t^i.

Motion Estimation. In this paper, for simplicity and computational efficiency reasons, we choose to track only the location and size. Let $x_t = (l_t^x, l_t^y, w_t, h_t)$ denote the object state parameters including the horizontal coordinate, vertical coordinate, width, and height, respectively. We use a Gaussian

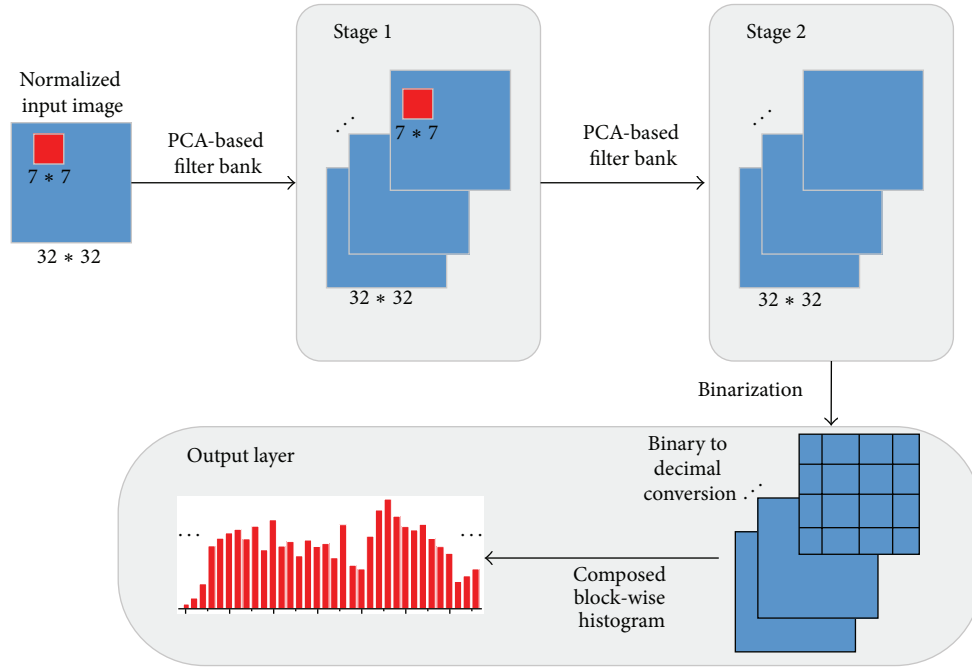


Figure 1: Illustration of the structure of the used PCANet deep network [32].

distribution to model the dynamic model between two consecutive frames.

Likelihood Evaluation. For each state x_t, there is a corresponding image patch that is normalized to $32 * 32$ pixels by image scaling. The likelihood function is calculated based on the proposed discriminative appearance model; that is, $p(z_t \mid x_t) = d_t$, where d_t is an output score from the proposed discriminative appearance model.

3.2. The Proposed Discriminative Appearance Model from PCANet. In this section, we address the problem of how to learn a data-driven and discriminative appearance model without large-scale pretraining. In the first frame, an unsupervised method is firstly used to learn the abstract feature of a target object by exploiting both classic principal component analysis (PCA) algorithms with recent deep learning representation architectures. Then, based on the feature learned from the above unsupervised method, a neural network with one hidden layer is trained in a supervised mode to construct a discriminative target appearance model.

More specifically, we use the newly proposed PCANet deep network [32] to prelearn the abstract feature of a target object. The PCANet is a simple convolutional deep learning network composed of cascaded PCA, binary hashing, and block histograms. The work on PCANet shows that applying arbitrary nonlinearities on top of PCA projections of image patches can be surprisingly effective for image classification. Inspired by their work, we propose a PCANet-based unsupervised method to effectively learn the abstract feature of a target object and the discriminative structure between the target and background.

The PCANet model is illustrated in Figure 1, and only the PCA filters need to be learned from the training images. Following the notations of Han Chan et al. [32], we will briefly review the PCANet model.

The Cascaded PCA. Denote $\{I_i \in \mathbb{R}^{m \times n}\}_{i=1}^{N}$ as N input training images and $k_1 \times k_2$ as the 2D convolutional filter size. Around each pixel, PCANet takes $k_1 \times k_2$ patch and collects all (overlapping) patches of the ith image as the training data. Then, PCANet computes projection vectors in such a way that most variations in the training data can be retained. The PCA filters in the PCANet are expressed as the leading principal eigenvectors. Similar to deep neural network, PCANet can stack multiple stages of PCA filters to extract higher level features.

Binary Hashing and Block Histograms. Let L_1 and L_2 denote the number of PCA filters in the first and second stage of PCANet, respectively. For each of the L_1 input images I_i^l for the second state, each input image has L_2 real-valued outputs $\{I_i^l * W_\ell^2\}_{\ell=1}^{L_2}$ from the second stage. These outputs are binarized via a hashing function, in which an output value is one for positive entries and zero otherwise.

Around each pixel, the vector of L_2 binary bits is viewed as a decimal number. This converts L_2 outputs of the ith input image I_i^l back into a single integer-valued image T_i^l, where $i = 1, \ldots, L_1$. Then, each of the L_1 images T_i^l is divided into m overlapping or nonoverlapping blocks. PCANet compute the histogram of the decimal values in each block and concatenate all m histograms into one vector and denote them as $H(T_i^l)$.

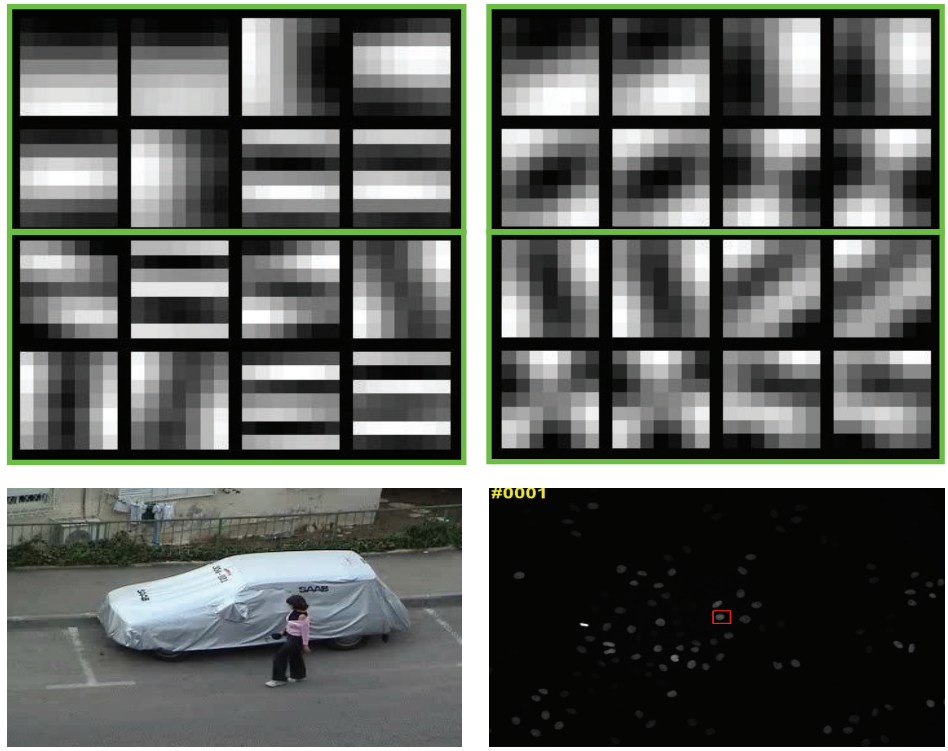


Figure 2: The PCA-based filters learned on the training data from the first frame of woman sequence from the online tracking benchmark (OTB) [41] and the Mitocheck cell dataset, respectively [40]. The top two rows show the eight PCA-based $7 * 7$ filters learned in first layer. The bottom two rows show the eight PCA-based $7 * 7$ filters in second layer.

After this encoding process, the feature of the input image I_i is then defined to be the set of block-wise histograms; that is, $f_i = [H(T_i^L), \ldots, H(T_i^{L_1})]$.

To empirically illustrate the efficacy of the learned PCANet features, we check the fine-tuned filters trained on the training data from a specific tracking task. In Figure 2, we show the PCA-based filters learned on the training data from the first frame of woman sequence from the online tracking benchmark (OTB) [41] and the Mitocheck cell dataset, respectively [40]. The top two rows show the eight PCA-based $7 * 7$ filters learned in first layer. The bottom two rows show the eight PCA-based $7 * 7$ filters in second layer. It is obvious that the proposed PCANet-based model can effectively learn the useful information from the data, such as edge and corner and junction detectors.

4. Experiments Evaluation

This section presents our implemental details, experimental configurations, dataset, and evaluation setting. The effectiveness of our tracking algorithm (named ours-1) is then demonstrated by quantitative and qualitative analysis on the online tracking benchmark (OTB) [41] and the Mitocheck cell dataset, respectively [40]. For the sake of computational robustness, we further consider the effect of the different PCA layers in PCANet (i.e., a variety of different numbers of the PCA layers) on the tracking performance.

4.1. Implementation Details and Experimental Configurations. To reduce computational cost, we simply consider the object state information in 2D translation and scaling in a particle filtering framework, where the corresponding variance parameters are set to 15, 15, 0.1, and 0.1, respectively. The proposed tracking method (i.e., ours-1) is implemented in Matlab without code optimization and runs on a PC with a 2.40 GHz processor and 12 G RAM. 1,000 samples are empirically drawn for particle filtering. For each particle, there is a corresponding image region normalized to a $32 * 32$ patch. The buffer size of a temporal sliding window is set as 25. The typical training time of PCANet-based deep network is about 10 seconds in Matlab without using GPUs. Our PCANet-based tracker takes about one second to process each video frame.

4.2. Datasets and Evaluation Settings

4.2.1. Datasets. To evaluate the performance of the proposed tracking method (i.e., ours-1) for tracking individual-cell/object, we use not only the Mitocheck cell dataset [40] but also the online tracking benchmark (OTB) [41]. The Mitocheck dataset is a time-lapse microscopic image sequence which contains higher cell density, larger intensity variability, and illumination variations. The online tracking benchmark (OTB) [41] is a collection of 50 video sequences tagged with 11 attributes which covers various challenging factors in visual tracking, such as deformation, fast motion,

background clutter, and occlusion. The 50 video sequences are defined with bounding box annotations.

4.2.2. Evaluation Settings. The OTB benchmark uses two different evaluation metrics: the precision plot and success plot. For the precision plot, a target object is considered to be successfully tracked on a video frame if the distance between the centers of the estimated box and the ground-truth bounding box is below a threshold. Thus, numerous precision plots can be obtained by varying the threshold values. Typically, the trackers are ranked based on the precision at threshold of 20 pixels for the precision plot. On the other hand, for the success plot, a target object is considered to be successfully located on a video frame if the predicted bounding box and the ground-truth bounding box have an intersection-over-union (IoU) overlap higher than a threshold. The success plot illustrates the percentage of frames considered to be successful. The area under curve (AUC) score is used to rank the tracking algorithms. Three different experiments are performed, that is, one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE). For TRE, the starting frame of the evaluation is randomized. For SRE, the initial bounding boxes are randomly perturbed. Please see the original paper [41] for more details. For the evaluation on the Mitocheck cell dataset [40], we just use the qualitative results to show the tracking performance due to the unavailability of ground-truth labeling.

4.3. Evaluation Results on the Online Tracking Benchmark (OTB)

Overall Performance. We quantitatively analysed the overall tracking performance, and Figure 3 shows the precision and success plots on all the 50 sequences of the top 10 tracking methods. In terms of both evaluation metrics, the proposed tracking method (i.e., ours-1) is able to obtain better results than any of the comparison methods due to the robust feature learning via online PCANet deep network. In the precision plot of OPE, the precision score of the proposed tracking method (i.e., ours-1) is 0.707, which is ranked the first place. Meanwhile, the other top four tracking methods are Struck (0.656), SCM (0.649), TLD (0.608), and VTD (0.576), respectively. In the success plot of OPE, the AUC score of the proposed tracking method (i.e., ours-1) is 0.566, which is also ranked the first place. Meanwhile, the other top four tracking methods are SCM (0.499), Struck (0.474), TLD (0.437), and ASLA (0.434), respectively. According to the precision and AUC scores, the proposed tracking method (i.e., ours-1) is comparable to the state-of-the-art tracking methods in both the precision and success plots.

Performance Analysis on 11 Different Attributes. To further analyse the proposed tracking method, we validate the performance of the proposed tracker on each attribute provided in the online tracking benchmark (OTB) [41]. In the OTB, there are 11 different attributes which describe a variety of tracking challenges. Each video sequence is annotated by some attributes. We report the precision and success plots of one-pass evaluation (OPE) for trackers on the 11 attributes in Figures 4 and 5, respectively. According to Figures 4 and 5, it is easy to observe that the proposed tracking method (i.e., ours-1) provides sufficient robustness to the 11 attributes, and our tracker consistently outperforms the other trackers in most of the challenges.

Qualitative Results. In Figure 6, we illustrate the qualitative results of four typical image sequences. To facilitate more detailed analysis, we further report the curves of center distance error per frame in Figure 7. As our tracker can better capture major variations in the data, we can observe that the proposed tracking method demonstrates superior performance over other tracking methods.

4.4. Effect of Different PCA Layers in PCANet. In this subsection, we investigate how the number of PCA layers in PCANet affects the tracking performance of the proposed method. Specifically, we compare our tracker (i.e., ours-1) with one different structure. The new variation of ours-1 is denoted as ours-2. Different to ours-1 tracker which contains two PCA filtering layers, ours-2 tracker contains three PCA filtering layers. Figure 8 demonstrates the performance comparison of the proposed tracking method with different PCA layers in PCANet in terms of the success and precision plots of TRE on the online tracking benchmark (OTB) [41]. We observe that ours-2 tracker with three PCA filtering layers obtains a better result than that of ours-1 tracker with two PCA filtering layers. This indicates that the performance of the proposed tracking method can be further improved when the number of PCA layers in PCANet is increased. However, the improvement is not significant and is computationally inefficient.

4.5. Qualitative Results on the Mitocheck Cell Dataset. To evaluate the performance of the proposed tracking method on individual-cell tracking, we test the proposed tracking method on the Mitocheck cell dataset [40]. In Figure 9, we report the qualitative tracking results of four individual-cells from the Mitocheck dataset. We can observe that the proposed tracking method simultaneously maintains holistic and local appearance information and therefore provides a compact representation of the cells. Consequently, the proposed tracking method can achieve a good performance on individual-cell tracking.

4.6. Discussion. In this paper, we focus on learning a robust PCANet-based appearance model for individual-cell/object tracking. According to the above experimental results on challenging dataset, the proposed tracking method has achieved promising results. However, the performance of the proposed tracker may be deteriorated when a target object is occluded over a long period of time. The reason is that the PCANet-based appearance model is updated via a simple yet concrete schema which does not explicitly detect occlusion. To address the problem, more complicated occlusion detection and forgetting schemas should be incorporated into the proposed tracker to achieve effective model updating.

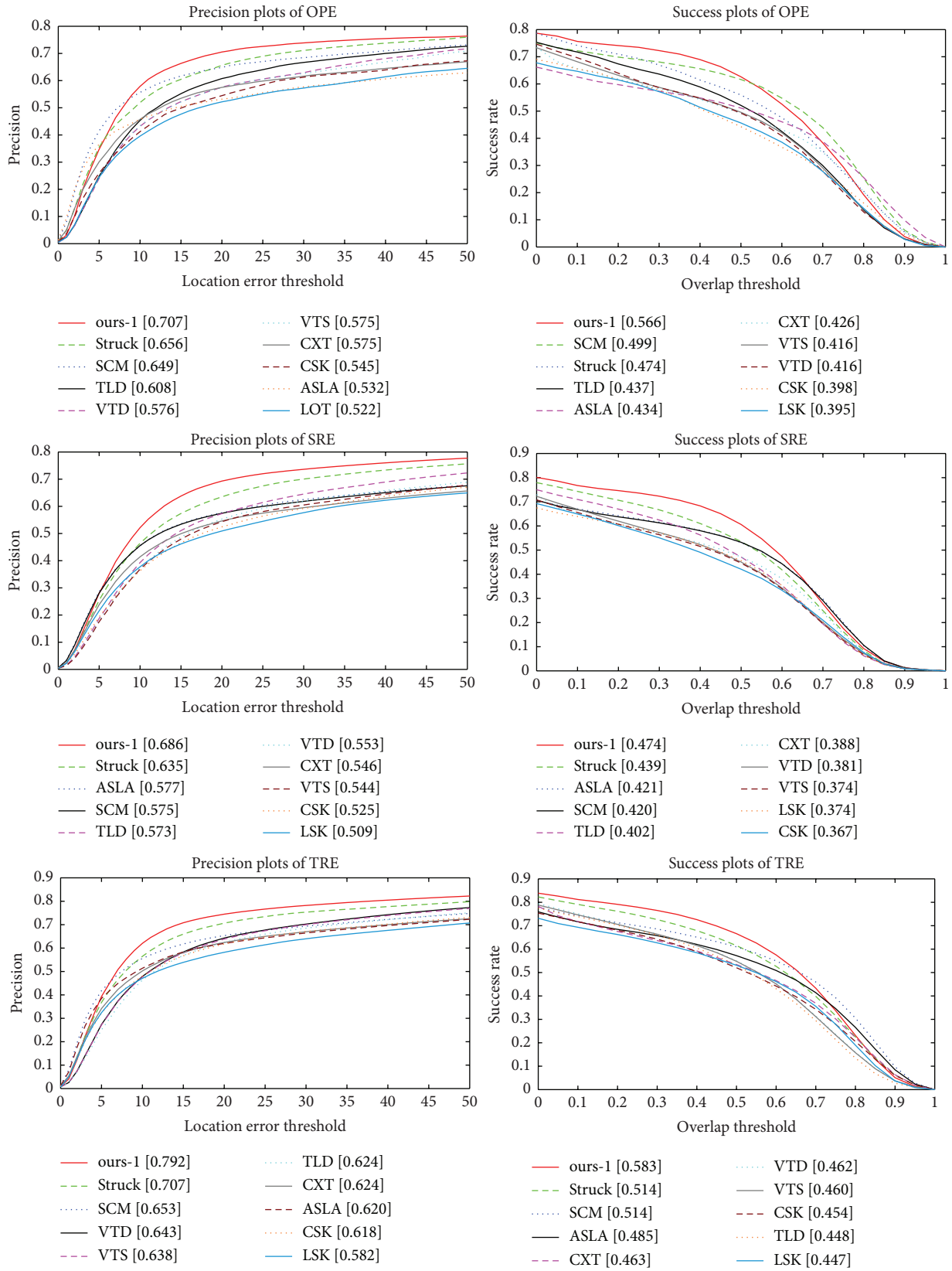


Figure 3: The precision and success plots of one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE) for the 50 sequences in the online tracking benchmark (OTB) [41], respectively. The legend lists the corresponding evaluation score for each tracking method. The proposed tracking method (i.e., ours-1 in red) is ranked first among the state-of-the-art trackers in both the precision and success plots.

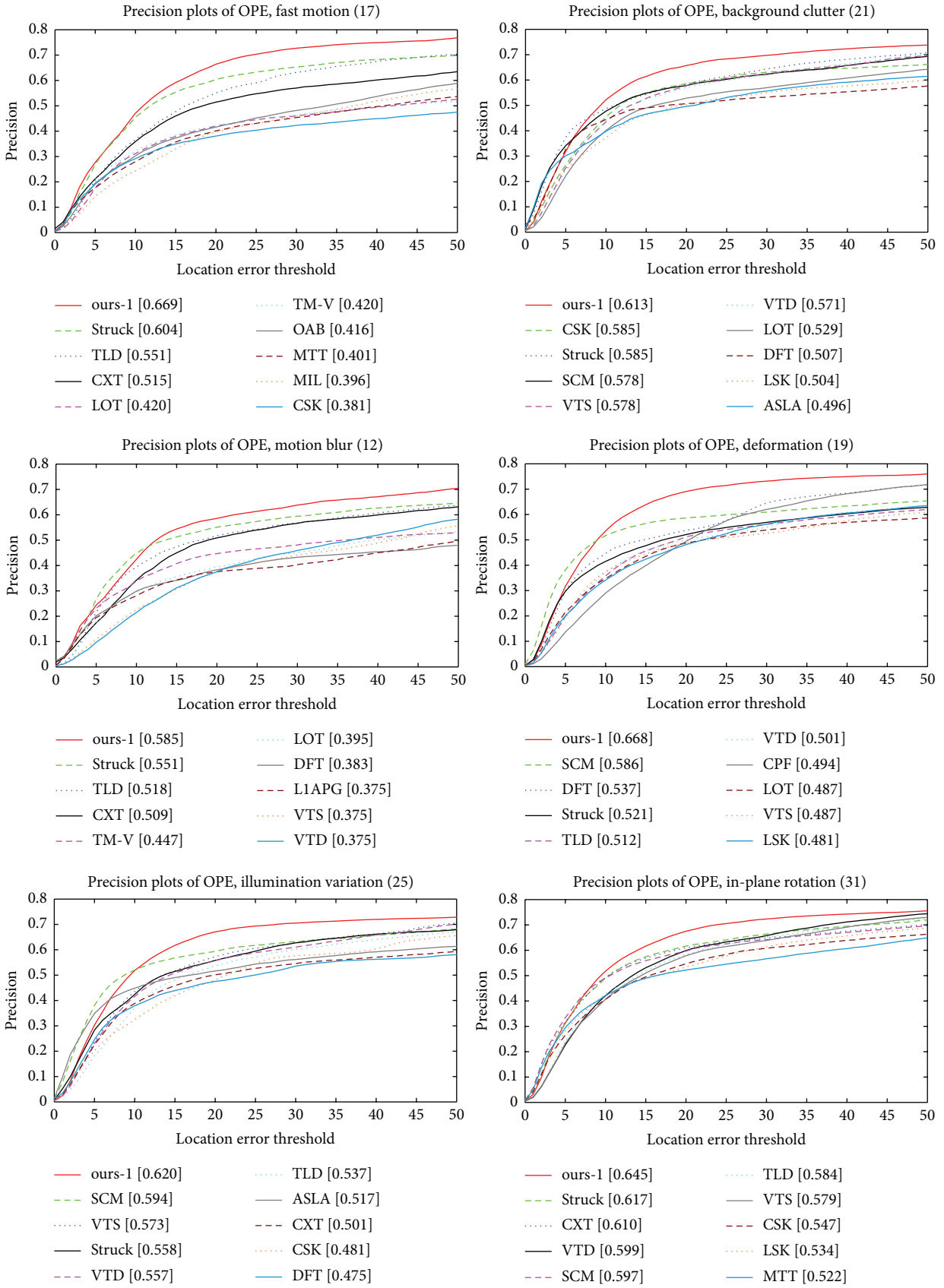


Figure 4: Continued.

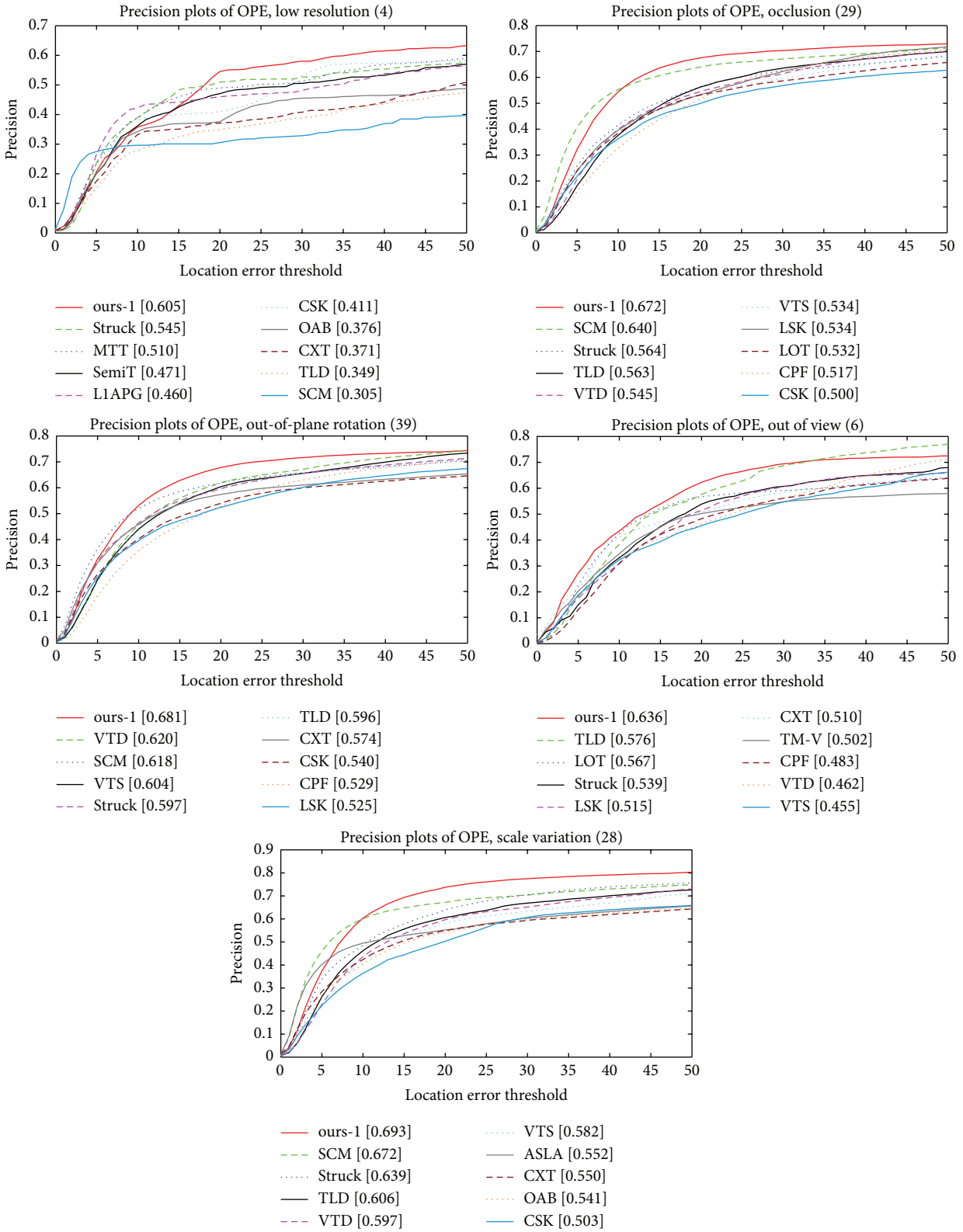


Figure 4: The precision plots of one-pass evaluation (OPE) for trackers on the 11 attributes. The values next to the attributes denote the number of video sequences involving the corresponding attribute.

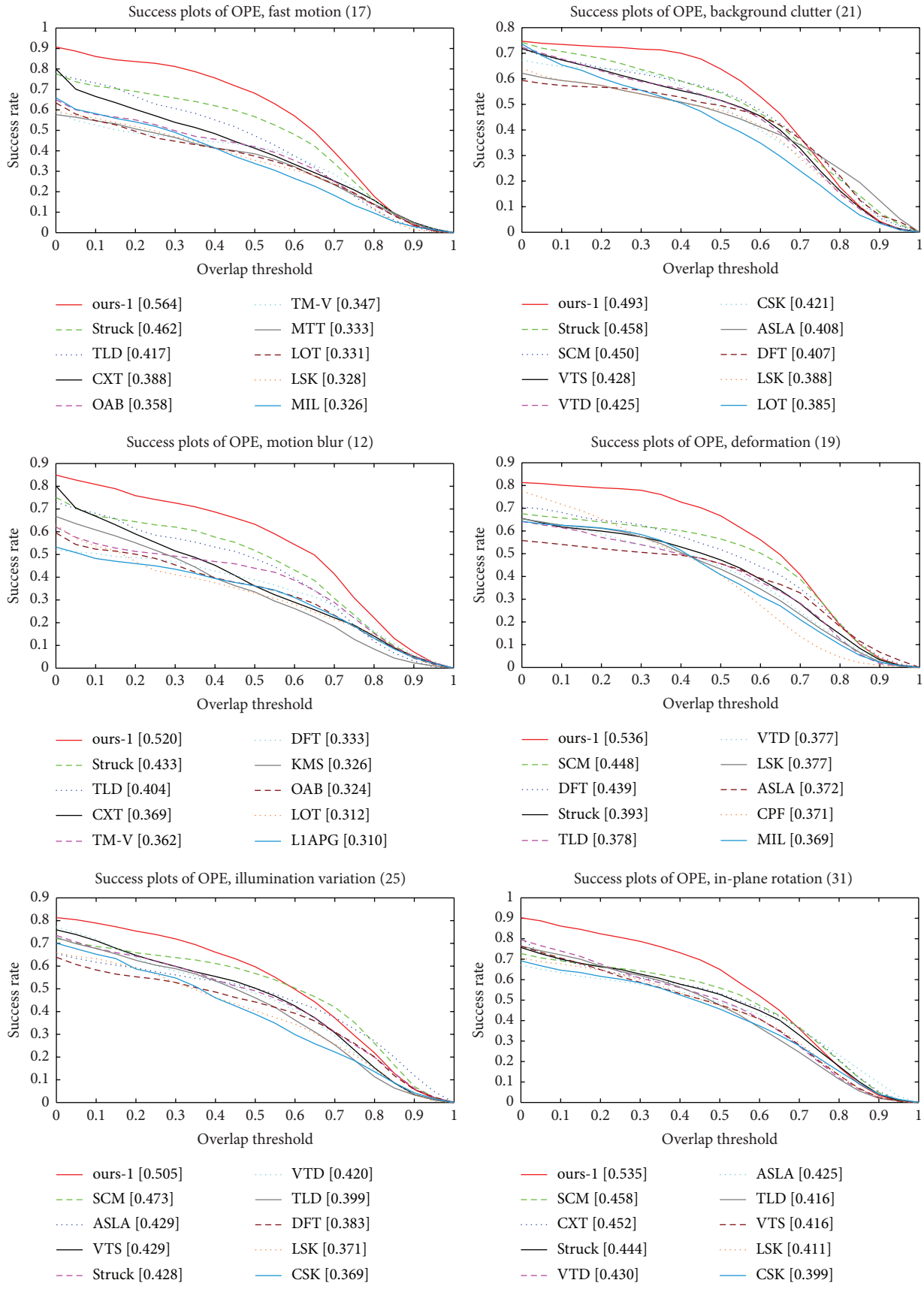


Figure 5: Continued.

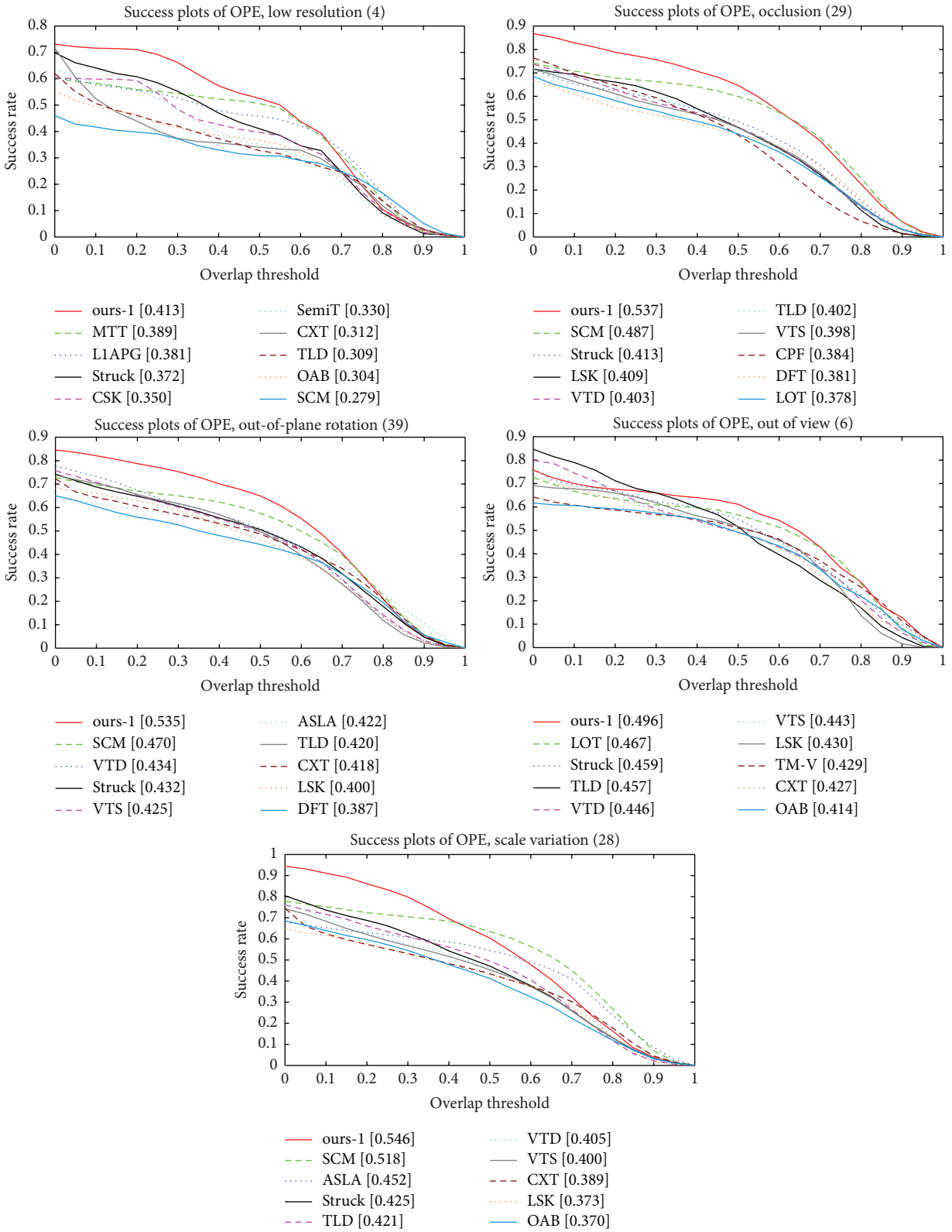


Figure 5: The success plots of one-pass evaluation (OPE) for trackers on the 11 attributes. The values next to the attributes denote the number of video sequences involving the corresponding attribute.

Figure 6: Qualitative results of the proposed tracking method (i.e., ours-1) on several challenging sequences from [41].

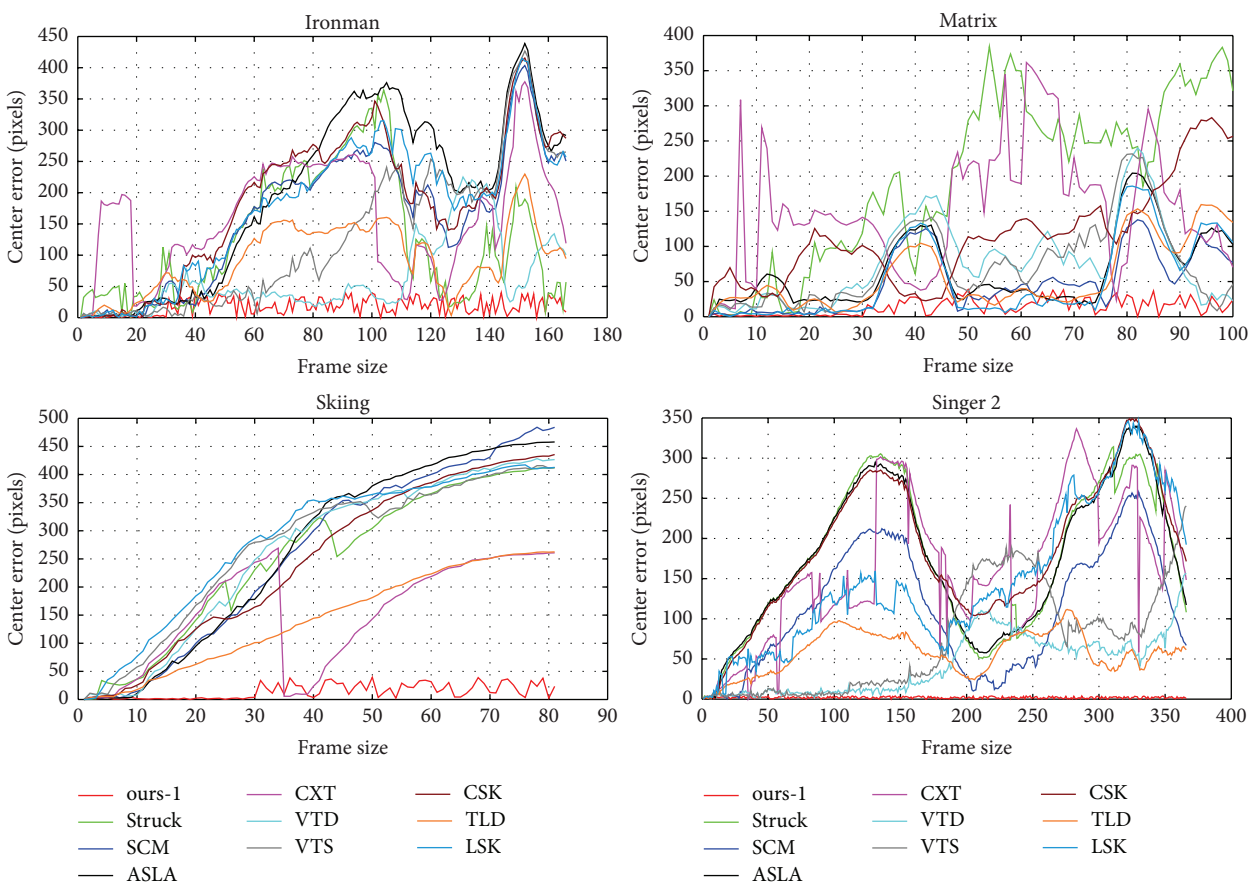


Figure 7: Quantitative results on the center distance error per frame for several challenging sequences from [41].

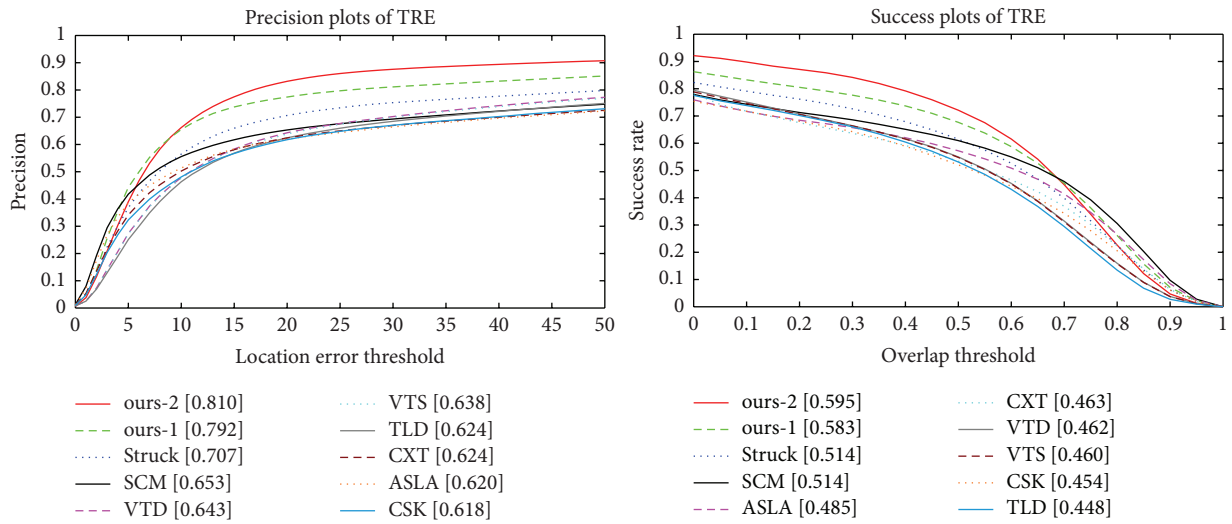


FIGURE 8: The precision and success plots of TRE for the proposed tracking method (e.g., ours-1 and ours-2) as the number of PCA filtering layers in PCANet grows. Please see the text for more details.

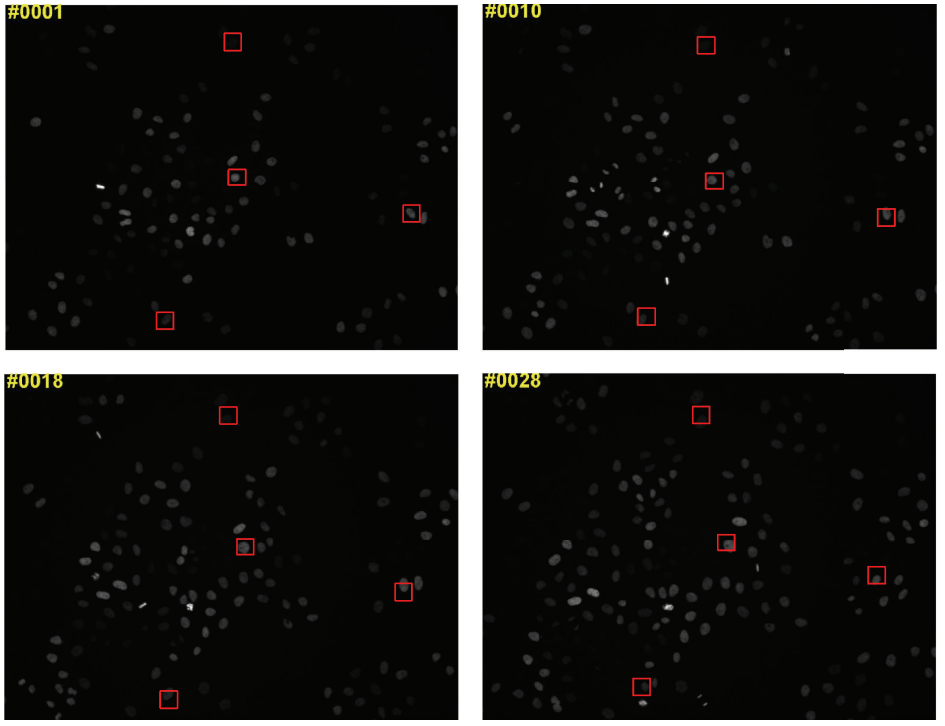


FIGURE 9: Qualitative results on individual-cell from the Mitocheck dataset [40].

5. Conclusion

We have proposed a robust feature learning method via PCANet deep network for robust individual-cell/object tracking in the time-lapse and 2D color imaging sequences. A cell/object is firstly effectively represented by composing of a PCA-based filter bank layer, a nonlinear layer, and a patch-based pooling layer, respectively. Then, a discriminative target appearance model is constructed by training a neural network with one hidden layer. Finally, to alleviate the tracker drifting problem, a sample update scheme is carefully designed to keep track of the most representative and diverse samples while tracking. Extensive experiments on challenging image sequences from the Mitocheck cell dataset

and the online tracking benchmark (OTB) [41] validate the robustness and effectiveness of the proposed individual-cell/object tracking method.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by Natural Science Foundation of China (nos. 61572205, 61572206, 51305142, and 61175121), Natural Science Foundation of Fujian Province (nos. 2015J01257 and 2013J06014), Promotion Program for Young and Middle-Aged Teacher in Science and Technology Research of Huaqiao University (nos. ZQN-PY210 and ZQN-YX108), and 2015 Program for New Century Excellent Talents in Fujian Province University.

References

[1] X. Chen, X. Zhou, and S. T. C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 762–766, 2006.

[2] E. Meijering, O. Dzyubachyk, and I. Smal, "Methods for cell and particle tracking," in *Imaging and Spectroscopic Analysis of Living Cells*, vol. 504, pp. 183–200, Elsevier, 2012.

[3] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen, "Tracking in cell and developmental biology," *Seminars in Cell and Developmental Biology*, vol. 20, no. 8, pp. 894–902, 2009.

[4] T. Kanade, Z. Yin, R. Bise et al., "Cell image analysis: Algorithms, system and applications," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '11)*, pp. 374–381, Kona, Hawaii, USA, January 2011.

[5] V. C. Abraham, D. L. Taylor, and J. R. Haskins, "High content screening applied to large-scale cell biology," *Trends in Biotechnology*, vol. 22, no. 1, pp. 15–22, 2004.

[6] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[7] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[8] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. van den Hengel, "A survey of appearance models in visual object tracking," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 1–58, 2013.

[9] T. Schroeder, "Long-term single-cell imaging of mammalian stem cells," *Nature Methods*, vol. 8, no. 4, pp. S30–S35, 2011.

[10] J. W. Young, J. C. W. Locke, A. Altinok et al., "Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy," *Nature Protocols*, vol. 7, no. 1, pp. 80–88, 2012.

[11] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[12] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008.

[13] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[14] S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[15] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proceedings of the 10th European Conference on Computer Vision*, Marseille, France, October 2008.

[16] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, 2014.

[17] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[18] F. Li, X. Zhou, J. Ma, and S. T. C. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 96–105, 2010.

[19] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1090–1097, Columbus, Ohio, USA, June 2014.

[20] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 260–267, IEEE, New York, NY, USA, June 2006.

[21] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 263–270, Barcelona, Spain, November 2011.

[22] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. Van Den Hengel, "Part-based visual tracking with online latent structural learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2363–2370, Portland, Ore, USA, June 2013.

[23] V. Takala and M. Pietikäinen, "Multi-object tracking using color, texture and motion," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–7, Minneapolis, Minn, USA, June 2007.

[24] Y. Lu, T. F. Wu, and S.-C. Zhu, "Online object tracking, learning, and parsing with and-or graphs," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 3462–3469, Columbus, Ohio, USA, June 2014.

[25] J. Fan, X. Shen, and Y. Wu, "Scribble tracker: a matting-based approach for robust tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1633–1644, 2012.

[26] X. Lou, M. Schiegg, and F. A. Hamprecht, "Active structured learning for cell tracking: algorithm, framework, and usability," *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, pp. 849–860, 2014.

[27] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[28] G. E. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, December 2012.

[30] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[31] J. Donahue, Y. Jia, O. Vinyals et al., "DeCAF: a deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 988–996, June 2014.

[32] T. Han Chan, K. Jia, S. H. Gao, J. W. Lu, Z. N. Zeng, and Y. Ma, "PCANet: a simple deep learning baseline for image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[33] G. Carneiro and J. C. Nascimento, "Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2592–2607, 2013.

[34] N. Y. Wang and D. Y. Yeung, *Learning a Deep Compact Image Representation for Visual Tracking*, Advances in Neural Information Processing Systems, 2013.

[35] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[36] L. J. Wang, W. L. Ouyang, X. G. Wang, and H. C. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3119–3127, Santiago, Chile, December 2015.

[37] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the 32nd International Conference on Machine Learning (ICML '15)*, pp. 597–606, 2015.

[38] C. Ma, J. B. Huang, X. K. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '15)*, pp. 3074–3082, Santiago, Chile, December 2015.

[39] H. S. Nam and B. Y. Han, "Learning multi-domain convolutional neural networks for visual tracking," http://arxiv.org/abs/1510.07945.

[40] http://www.mitocheck.org/cgi-bin/mtc?action=show_movie;query=243867.

[41] Y. Wu, J. W. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2411–2418, IEEE, Portland, Ore, USA, June 2013.

[42] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, September 2009.

[43] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1830–1837, Providence, RI, USA, June 2012.

[44] K. H. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proceedings of the European Conference on Computer Vision (ECCV '12)*, Florence, Italy, October 2012.

[45] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 171–190, 2014.

[46] X. Jia, H. C. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1822–1829, IEEE, Providence, RI, USA, June 2012.

[47] Z. Zhang and K. H. Wong, "Pyramid-based visual tracking using sparsity represented mean transform," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1226–1233, Columbus, Ohio, USA, June 2014.

[48] B. N. Zhong, H. X. Yao, S. Chen, R. R. Ji, T.-J. Chin, and H. Z. Wang, "Visual tracking via weakly supervised learning from multiple imperfect oracles," *Pattern Recognition*, vol. 47, no. 3, pp. 1395–1410, 2014.

[49] Y. Zhou, X. Bai, W. Y. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *International Journal of Computer Vision*, vol. 118, no. 3, pp. 337–363, 2016.

[50] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[51] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[52] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[53] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking," http://arxiv.org/abs/1509.05520.

[54] W. M. Zuo, X. H. Wu, L. Lin, L. Zhang, and M.-H. Yang, "Learning support correlation filters for visual tracking," http://arxiv.org/abs/1601.06032.

[55] X. Lou, U. Koethe, J. Wittbrodt, and F. A. Hamprecht, "Learning to segment dense cell nuclei with shape prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1012–1018, IEEE, Providence, RI, USA, June 2012.

[56] O. Dzyubachyk, W. A. Van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 852–867, 2010.

[57] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1424–1435, 2015.

[58] H. X. Li, Y. Li, and F. Porikli, "Deeptrack: learning discriminative feature representations by convolutional neural networks for visual tracking," in *Proceedings of the British Machine Vision Conference (BMVC '14)*, BMVA Press, Nottingham, UK, September 2014.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 2090286, 10 pages
http://dx.doi.org/10.1155/2016/2090286

Research Article

Uncovering Driver DNA Methylation Events in Nonsmoking Early Stage Lung Adenocarcinoma

Xindong Zhang,[1] Lin Gao,[1] Zhi-Ping Liu,[2] Songwei Jia,[1] and Luonan Chen[3,4,5]

[1]*School of Computer Science and Technology, Xidian University, Xi'an 710000, China*
[2]*Department of Biomedical Engineering, School of Control Science and Engineering, Shandong University, Shandong 250061, China*
[3]*Key Laboratory of Systems Biology, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China*
[4]*Institute of Industrial Science, University of Tokyo, Tokyo 153-8505, Japan*
[5]*School of Life Science and Technology, ShanghaiTech University, Shanghai 201210, China*

Correspondence should be addressed to Lin Gao; lgao@mail.xidian.edu.cn

Received 28 May 2016; Revised 28 June 2016; Accepted 5 July 2016

Academic Editor: Quan Zou

As smoking rates decrease, proportionally more cases with lung adenocarcinoma occur in never-smokers, while aberrant DNA methylation has been suggested to contribute to the tumorigenesis of lung adenocarcinoma. It is extremely difficult to distinguish which genes play key roles in tumorigenic processes via DNA methylation-mediated gene silencing from a large number of differentially methylated genes. By integrating gene expression and DNA methylation data, a pipeline combined with the differential network analysis is designed to uncover driver methylation genes and responsive modules, which demonstrate distinctive expressions and network topology in tumors with aberrant DNA methylation. Totally, 135 genes are recognized as candidate driver genes in early stage lung adenocarcinoma and top ranked 30 genes are recognized as driver methylation genes. Functional annotation and the differential network analysis indicate the roles of identified driver genes in tumorigenesis, while literature study reveals significant correlations of the top 30 genes with early stage lung adenocarcinoma in never-smokers. The analysis pipeline can also be employed in identification of driver epigenetic events for other cancers characterized by matched gene expression data and DNA methylation data.

1. Introduction

As a leading cause of death worldwide, lung cancer is mainly attributed to smoking in both men and women [1, 2], of which the most common histological subtype is adenocarcinoma. However, as smoking rates decrease, proportionally more cases occur in never-smokers [3]. Lung adenocarcinoma in never-smokers shows obvious distinctions in clinical and molecular mechanism to those cigarette smoking [4]. Both genetics and epigenetics in cancer genomes have been suggested to account for the development of lung adenocarcinoma.

As one of the vital epigenetic mechanisms, DNA methylation regulates gene expression without alterations in DNA sequence [5, 6] and plays key roles in X chromosome inactivation, genome stability, chromatin structure, embryonic development, differentiation, and maintenance of pluripotency in normal somatic cells [7, 8]. Genome-scale methylation-profiling techniques have confirmed the existence of widespread aberrations of DNA methylation patterns in human cancer genome [9–12]. Studies of DNA methylation have suggested that both global DNA hypomethylation and gene-specific hypermethylation may contribute to the initiation and progression of tumorigenesis, as well as gene body methylation [13–15]. It is challenging but of great significance to distinguish genes whose methylation changes are crucial in cancer occurrence, progression, or metastasis from genes whose methylation changes merely have effects on the process

of tumorigenesis in cancer research and therapy [13]. Unlike somatic mutations in the genome, DNA methylation is inherently reversible and serves as potential drug targets in cancer intervention [16, 17].

Numerous studies have focused on discovering genes whose DNA methylation potentially plays key roles in tumorigenesis of lung adenocarcinoma, including integration of genome-scale DNA methylation and gene expression [18–21]. The main idea of these works is to search genes whose gene expression fluctuations are highly correlated to DNA methylation changes. However, there is a deficiency derived on the complexity of the gene expression regulation. Both genetic and epigenetic alterations can contribute to gene expression as well as other transcriptional factors in sophisticated manners in complex diseases [22, 23]. In tumors, a differential gene expression may be induced by an aberrant DNA methylation in the promoter of the gene but also may be a consequence regulated by its upstream genes in regulatory mechanisms. These appeal to a great attention in uncovering driver DNA methylations, which play major roles in methylation-associated gene silencing and drive malignant transformation [5, 13]. In this work, we refine the generalized description of driver methylation as two properties. (1) Driver DNA methylation should induce distinctive expressions in tumors with differential DNA methylation (T-DM) when compared to expressions in matched adjacent nontumor (normal) and tumors with nondifferential DNA methylation (T-NDM), and (2) driver methylation should induce a distinct regulation module in the network perspective. The first property guarantees the major role of DNA methylation in the regulation of gene expression, while the second property guarantees the functional effects of driver genes on tumorigenesis.

Focusing on genes differentially expressed among matched adjacent nontumors (normal), tumors with aberrant DNA methylation (T-DM), and tumors without aberrant DNA methylation (T-NDM), we integrate genome-wide DNA methylation data and gene expression data to uncover driver methylation events in never-smokers in early stage lung adenocarcinoma. Differential network analyses show significant changes of DNA methylation-responsive modules in network topology across normal, T-DM, and T-NDM, which imply potential mechanisms of identified driver genes underlying the tumorigenesis.

2. Materials and Methods

2.1. Data Sets. Both the DNA methylation data and gene expression data are downloaded from NCBI Gene Expression Omnibus (GEO) with accession number GSE32867 [18]. The series contains 59 samples with paired genome-scale DNA methylation profiling and gene expression. Stage I and stage II are merged as early stage and stages III-IV are labeled with late stage [18]. After removing noisy data [18], 22 samples are labeled with "never smoking" and "early stage" simultaneously. Paired DNA methylation data and gene expression data of these 22 samples are collected to further analysis. Probes in gene expression data are firstly mapped to Entrez gene ID and expression values sharing same Entrez gene IDs are averaged among samples.

2.2. Schematic Overview of the Analysis Pipeline. The schematic overview of the analysis pipeline is shown in Figure 1, and detailed procedures are described in the following sections.

2.2.1. Candidate Driver Gene Selection. Figure 1(a) shows a brief schematic overview of this procedure. The difference matrix is firstly created to measure differences of beta values of DNA methylation between tumor and normal. The kernel probability distribution with normal smoothing function is used to estimate the probability density distribution for each probe in the difference matrix (Figure 1(a)). The hypothesis is that the differences of beta values for given probes come from distributions with the mean 0 and unknown variances. The cumulative density function (CDF) is used to estimate the probability of a beta value falling within given interval. Hypermethylation and hypomethylation are determined by the upper bound CDF > 0.95 and the lower bound CDF < 0.05, respectively. For each probe, tumors are partitioned into two groups, tumors with differential methylation group (T-DM) and tumors without differential methylation group (T-NDM).

Then, the two-sample t-test is used to evaluate differential expression under conditions [24], and p values are adjusted by the procedure introduced by Storey [25]. The mapping from DNA methylation to gene expression is performed by shared Entrez gene ID. Probes remain if the mapped genes are differentially expressed in T-DM when compared to normal and T-NDM (adjusted p value < 0.05), which implies that the differential methylation of given probes in T-DM is more likely to induce significant expression changes. Probes mapping to same genes are removed if hypermethylation and hypomethylation coexist in more than 5 samples. Then samples in T-DMs and T-NDMs merge, respectively, by shared Entrez gene ID and serve as T-DM and T-NDM of the gene.

We then search for genes whose expressions are highly discriminative and consistent in T-DM when compared to normal and T-NDM. Many types of statistics, such as Wilcoxon score, Pearson correlation coefficient (PCC), or mutual information (MI), could be used to score the relationship between gene expression and class labels, and a T-score method is used in this work [26]. For a given gene, let a be the gene expression levels across samples with class c and the discriminative score $s(a, c)$ is defined as the t-test statistic. To determine whether the discriminative level of the gene among groups is consistent, we permute the class c by 1000 times and obtain a background distribution of the discriminative scores $S'(a, c)$ derived on the gene expression levels a and permuted class c'. Genes with significant values (p value < 0.05) among groups (normal versus T-DM and T-DM versus T-NDM) are considered differentially methylated and served as candidates for further analysis.

2.2.2. Detection of DNA Methylation-Responsive Module. To construct the DNA methylation-responsive module for a

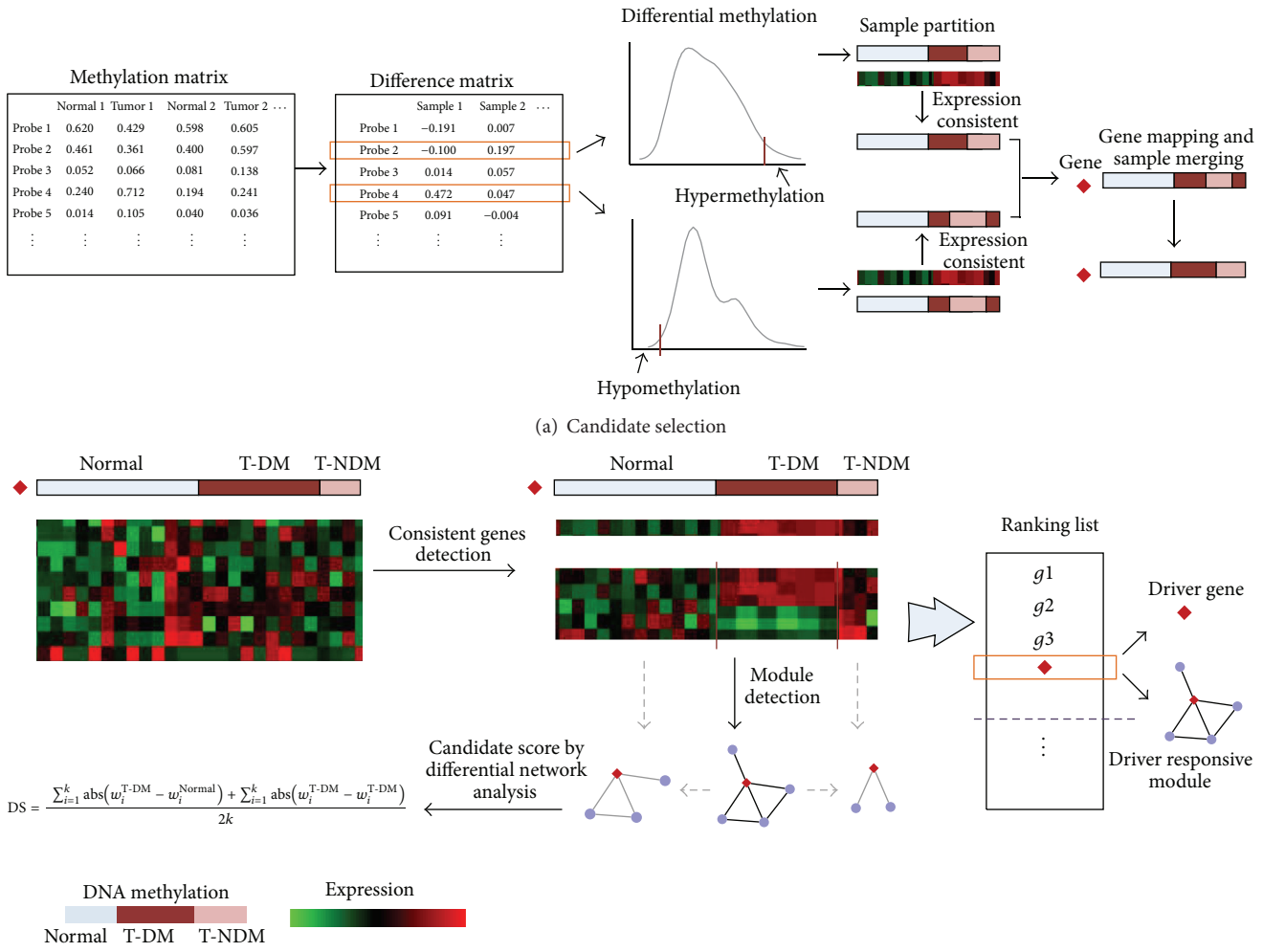


Figure 1: Schematic overview of the pipeline proposed in this work. (a) Candidate gene selection. Methylation matrix of continuous beta values is converted into difference matrix and discretized by kernel distribution function, which partition samples into normal, T-DM, and T-NDM. Probes are mapped to genes after noise filtering and genes passing the consistent test are collected as candidate driver genes. (b) For each candidate gene, a subset of DM responsive genes is collected and DM responsive modules are constructed by the CLR method. Candidate driver genes are ranked by differential scores derived on the differential network analysis.

candidate gene g, we firstly recognize a set of genes whose expressions are highly discriminative among groups defined by DNA methylation profiles of g. These genes are potentially responsive to aberrant DNA methylation of g.

The Context Likelihood of Relatedness (CLR) method [27] is used to assess regulatory relationships among these genes. CLR estimates MI for each pair of variables and corrects the MI via a background-corrected procedure. In particular, for mutual information $I(X_i; X_j)$, CLR scores the relatedness between a pair of variables X_i and X_j by the joint likelihood measurement:

$$z_{ij} = \sqrt{z_i^2 + z_j^2}, \tag{1}$$

where

$$z_i = \max\left(0, \frac{I\left(X_i; X_j\right) - \mu_i}{\sigma_i}\right), \tag{2}$$

where μ_i and σ_i are the mean and standard deviation derived on the empirical distribution of MI between X_i and arbitrary variables X_k $(k = 1, 2, \ldots, n)$ and $I(X_i; X_j)$ is the mutual information of X_i and X_j.

CLR employs B-spline smoothing and discretization method [28] to estimate the MI for a pair of variables. However, it is time-consuming in this work under diversiform conditions and permutations. Thus, we use the following estimation method to calculate MI for pair of variables X_i and X_j [29]; that is,

$$I\left(X_i; X_j\right) = -\frac{1}{2}\log\left(1 - \rho^2\right), \tag{3}$$

where ρ is the PCC of X_i and X_j.

An experienced threshold δ is necessary when CLR is employed. A larger threshold results in a higher precision but a smaller size of responsive modules. The size of more

than 70% modules is less than three when $\delta = 4.46$, while the size of 80% modules is larger than 3 when $\delta = 4.46$ and approximate ranking lists of top 30 genes are obtained when δ falls in the interval between 3.96 and 5.46. Thus, we set $\delta = 4.46$ in this work.

2.2.3. Scoring Candidate Driver Genes by Differential Network Analysis. Differential network analysis reveals dynamic changes of pathways and potential mechanisms in complex diseases including cancers [30]. For each candidate gene, we calculate CLR scores for edges in responsive modules under normal and T-NDM. Differential scores are calculated to estimate network differences among groups. The differential score (DS) is yielded by the following equation:

$$\mathrm{DS} = \frac{\sum_{i=1}^{k} \mathrm{abs}\left(w_i^{\text{T-DM}} - w_i^{\text{Normal}}\right) + \sum_{i=1}^{k} \mathrm{abs}\left(w_i^{\text{T-DM}} - w_i^{\text{T-TDM}}\right)}{2k}, \quad (4)$$

where w_i is the CLR score of the ith edge and k is the number of edges in driver methylation-responsive module. Then candidate genes are prioritized by DS scores in descending order.

3. Results

We focus on the detection of differentially methylated genes which play key roles in tumorigenesis ("driver methylation gene") and modules responsive to aberrant methylation of these genes. Rather than genes with consistent expressions to DNA methylation levels in whole tumors, we detect genes differentially expressed and consistent with DNA methylation in T-DM when compared to normal and T-NDM.

3.1. Identification of Candidate Driver Genes in Tumorigenesis. By integrating DNA methylation and corresponding gene expression data, the samples are partitioned into three groups (normal, T-DM, and T-NDM) for each gene (Figure 1(a)). Firstly, we remove genes that are not differentially expressed in T-DM when compared to normal and T-NDM. Then a permutation test is performed to determine the significance of the consistency of gene expression changes in T-DM when compared to T-NDM. To obtain a significant level of differences, we randomly permute T-DM and T-NDM and calculate differences. After 1000 times permutation, a background distribution of differences is constructed. After removing genes with the absolute mean beta value less than 0.1, 135 genes remain in the candidate list (see Supplementary File in Supplementary Material available online at http://dx.doi.org/10.1155/2016/2090286). We perform a functional enrichment analysis using DAVID [31, 32]. Of these 135 genes, 115 are annotated to GO terms including cancer-related functions such as response to stimulus, development process, cell differentiation, cell adhesion, cell growth and cell death, DNA repair, and apoptosis, which imply potential relationships between cancers and these 135 genes.

3.2. Detection Responsive Modules of Candidate Driver Genes. Biological network reveals cell's functional organization [33]. To characterize the functional implications of candidate driver genes in tumorigenesis, we detect modules responsive to differential methylation of candidate driver genes (Section 2). Totally, 130 of 135 modules have at least one edge when the threshold of CLR is set to 4.46, and the mean size of 130 modules is 15.

3.3. Prioritization of Candidate Driver Genes by Differential Network Analysis. We argue that a driver DNA methylation can induce not only a distinctive gene expression in T-DM, but also a distinctive module responsive to the alteration. We score each candidate driver gene by analysis of the differential level of the responsive module. Candidate driver genes are ranked by differential scores in descending order.

We testify the significance of the differential score to a background distribution derived from random permutations. For a given candidate driver gene, genes are randomly selected from its possible responsive genes with module size maintained, and a new module is constructed by CLR with $\delta = 4.46$ as well as a differential score. A sequence of DS$'$ consisting of random differential scores is obtained after 1000 times random permutation. Of 135 candidate driver genes, 130 genes pass the test with p value < 0.01.

We also perform a differential network analysis of responsive modules under different CLR thresholds from 1.96 to 6.96 with step 0.5. Almost all modules obtain significant differential scores under CLR cutoffs (Supplementary File). Table 1 lists details of top 30 genes.

4. Discussion

We build two lists as background to testify the accuracy of the ranked list. The first consists of genes that show absolute mean fold change larger than 0.2 in T-DM and literature annotated in lung cancer. Totally, 29 genes are contained in the first list and denoted as Standard_Lit. The other one comes from Selamat et al. of 76 genes [18]. In fact, this list is not very suitable because genes in Selamat et al. are confused with differentially methylated genes under smoking and late stage. Thus, we select genes covered by list from Selamat et al. and our list. Totally 19 genes are in the list and denote as Standard_Sel. Genes in these two lists are listed in Supplementary File.

We test the accuracy of our list to Standard_Lit and Standard_Sel; Figure 2(a) shows the ROC curves with AUC = 0.686 and AUC = 0.628, respectively, which means that over half of genes in two standard lists are high-ranked in our list. Figure 2(b) shows the overlaps of the top 30 genes in our list to Standard-Lit and Standard-Sel. For Standard-Lit, 12 of 29 genes are overlapped (Fisher exact test p value = 0.0018), while for Standard-Lit, 10 of 29 genes are overlapped (Fisher exact test p value = $2.67E-04$).

The ranked list is also validated by literature annotation. Of the top 30 genes, 27 genes are previously reported to be cancer-relevant, while 17 of them are lung cancer or non-small-cell lung cancer-related (Table 1).

Table 1: Top 30 genes ranked by differential score in lung adenocarcinoma.

Gene symbol[a]	Differential score	Number of samples in T-DM group[b]	p value
FAM107A [34]	16.301	20	$7.80E-06$
SPARCL1 [35, 36]	14.920	20	$1.40E-07$
TRPC6 [37]	14.649	11	$<1.0E-10$
CRYAB [38]	14.508	12	$3.84E-10$
WFDC3	14.483	−14	$<1.0E-10$
EFEMP2 [39]	13.958	20	$<1.0E-10$
MX2 [40, 41]	13.895	−18	$2.12E-05$
PLA2G4C [42]	13.870	−8	$<1.0E-10$
ST6GALNAC5 [43]	13.848	9	$<1.0E-10$
PLAT [44]	13.690	8	$2.45E-04$
TCF21 [45]	13.664	22	$<1.0E-10$
SOX17 [46]	13.368	22	$<1.0E-10$
SH3GL2 [47]	13.300	5	$<1.0E-10$
MAMDC2 [18]	13.274	19	$4.54E-07$
GCNT3 [48]	13.238	−14	$<1.0E-10$
MSR1 [49]	13.144	−16	$<1.0E-10$
PPP1R14D [50]	13.057	−12	$<1.0E-10$
COL5A2 [51]	13.045	19	$6.67E-04$
PTPRH [52]	12.967	−16	$8.98E-13$
HKDC1 [53]	12.961	−20	$<1.0E-10$
CDH13 [54]	12.932	−20	$3.34E-04$
CFI [55]	12.932	5	$1.20E-04$
ARL14	12.880	−12	$2.06E-04$
MMP9 [56]	12.866	7	$<1.0E-10$
CELSR3	12.856	16	$4.65E-10$
CDO1 [57]	12.846	22	$<1.0E-10$
AGR2 [58]	12.836	−22	$<1.0E-10$
S100P [59, 60]	12.828	−10	$2.29E-04$
DOCK2 [61]	12.777	20	$2.54E-03$
TNFRSF1B [62]	12.736	13	$<1.0E-10$

[a]Bold: gene literature annotated to lung cancer.
[b]−: Gene hypomethylated in samples.

Table 2: Functional annotation of driver-responsive network to KEGG signaling pathways (p value < 0.01).

Gene symbol	Enriched KEGG signaling pathway	p value
SPARCL1	CYTOSOLIC_DNA_SENSING	$3.22E-03$
TRPC6	PPAR_SIGNALING	$9.50E-03$
	P53_SIGNALING	$9.50E-03$
	MTOR_SIGNALING	$7.16E-03$
	NOTCH_SIGNALING	$6.47E-03$
EFEMP2	NOTCH_SIGNALING	$9.70E-03$
MX2	RIG_I_LIKE_RECEPTOR_SIGNALING	$9.33E-04$
PLA2G4C	PPAR_SIGNALING	$9.50E-03$
	P53_SIGNALING	$9.50E-03$
	MTOR_SIGNALING	$7.16E-03$
	NOTCH_SIGNALING	$6.47E-03$
ST6GALNAC5	PPAR_SIGNALING	$9.50E-03$
	P53_SIGNALING	$9.50E-03$
	MTOR_SIGNALING	$7.16E-03$
	NOTCH_SIGNALING	$6.47E-03$
PLAT	TOLL_LIKE_RECEPTOR_SIGNALING	$3.09E-03$
	NOD_LIKE_RECEPTOR_SIGNALING	$1.16E-03$
	CYTOSOLIC_DNA_SENSING	$9.44E-04$
	JAK_STAT_SIGNALING	$6.99E-03$
TCF21	FC_EPSILON_RI_SIGNALING	$4.89E-03$
GCNT3	NOTCH_SIGNALING	$9.70E-03$
MSR1	NOTCH_SIGNALING	$9.70E-03$
PTPRH	B_CELL_RECEPTOR_SIGNALING	$9.80E-03$
HKDC1	PPAR_SIGNALING	$9.50E-03$
	P53_SIGNALING	$9.50E-03$
	MTOR_SIGNALING	$7.16E-03$
	NOTCH_SIGNALING	$6.47E-03$
CDH13	ERBB_SIGNALING	$1.55E-03$
	T_CELL_RECEPTOR_SIGNALING	$2.38E-03$
CFI	PPAR_SIGNALING	$2.90E-03$
	MAPK_SIGNALING	$3.47E-03$
ARL14	VEGF_SIGNALING	$3.93E-03$
S100P	HEDGEHOG_SIGNALING	$6.47E-04$
	TGF_BETA_SIGNALING	$1.51E-03$
DOCK2	CHEMOKINE_SIGNALING	$3.49E-05$
	TOLL_LIKE_RECEPTOR_SIGNALING	$4.85E-03$
	NOD_LIKE_RECEPTOR_SIGNALING	$3.27E-05$
	T_CELL_RECEPTOR_SIGNALING	$5.42E-03$
	B_CELL_RECEPTOR_SIGNALING	$2.66E-03$
TNFRSF1B	NOTCH_SIGNALING	$7.06E-03$
	FC_EPSILON_RI_SIGNALING	$1.24E-03$

We also annotate responsive modules of top 30 ranked genes to KEGG signaling pathways. Among them, responsive modules for 18 genes are enriched with KEGG signaling pathways with significance level p value < 0.01, which imply significant relations of these responsive modules to cancer processes (Table 2) and indicate potential mechanism changes induced by aberrant DNA methylation. The KEGG signaling pathways are collected from MsigDB [63, 64].

Of 30 top ranked genes, *FAM107A*, *MAMDC2*, *SOX17*, *TCF21*, *PTPRH*, and *CDO1* have been previously reported with aberrant DNA methylation in lung cancer [18, 34, 45, 46, 52, 57]. All these genes obtain higher occurrences ($n > 19$) in lung adenocarcinoma. *AGR2*, *CDH13*, *CRYAB*, *MX2*, *SH100P*, and *SH3GL2* are reported with aberrant gene expression [38, 40, 47, 54, 58, 59], while *AGR2*, *CDH13*, and *MX2* are of high occurrences in aberrant DNA methylation ($n \geq 18$). Differential expression of these genes has been reported playing crucial roles in key pathways in tumorigenesis or serving as potential prognostic targets. With higher occurrences, the

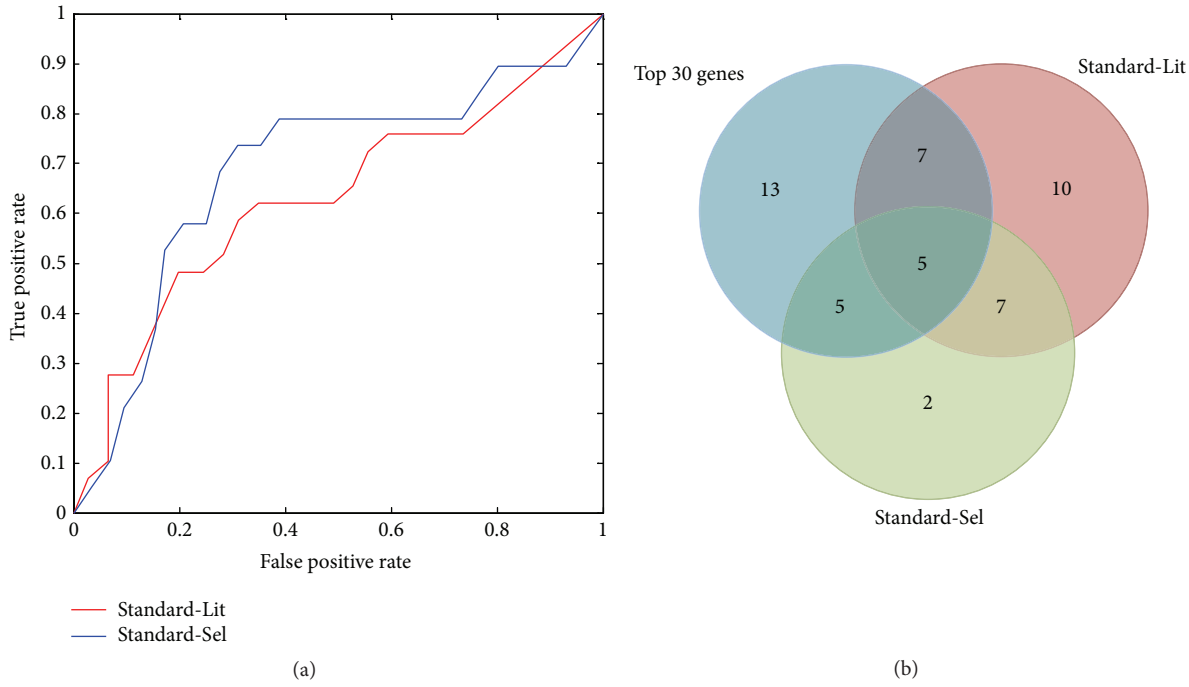


FIGURE 2: Comparison of the ranked list to two standard sets denoted by Standard-Lit and Standard-Sel. (a) ROC curves of our ranked list compared to Standard-Lit and Standard-Sel with AUC equal to 0.686 and 0.628, respectively. (b) Venn diagram showing the overlap of top 30 ranked genes in our list to Standard-Lit and Standard-Sel.

correlation of differential gene expression and aberrant DNA methylation of *AGR2*, *CDH13*, and *MX2* have been reported relevant to lung adenocarcinoma [18].

Alpha B-crystallin (*CRYAB*) is one of the important members of the small heat-shock protein family with aberrant DNA methylation occurring in 12 of 22 samples. The upregulated expression of *CRYAB* is reported relevant to the poor survival of patients with non-small-cell lung cancer (NSCLC) [38]. Interestingly, we find a contrary expression pattern in early stage lung adenocarcinoma in nonsmoking patients (Figure 3). A decreased expression is observed in both T-DM (p value = $8.20E-11$) and T-NDM (p value = $7.72E-8$) when compared to normal, while a relatively weak difference is also observed between T-DM group and T-NDM group (mean fold change difference = 0.07, p value = 0.15), which implies multiple mechanisms in regulation of *CRYAB*, as well as DNA hypermethylation. The responsive module of *CRYAB* is highly changed in normal and T-NDM (DS = 14.508, p value = $3.84E-10$). The similar case is *SH3GL2*, deletion of which downregulates tumor growth by modulating *EGFR* signaling [47].

Another interesting case is *S100P*, which has been reported as a key gene in tumor progression in both initial stage and advanced stage in lung adenocarcinoma [60]. The gene shows distinctive expressions among normal, T-DM, and T-NDM. There are nearly no changes existent in gene expression between normal and T-NDM, while in T-DM, upregulation is observed, which implies that the upregulation of *S100P* may be an important step in the early stage of lung adenocarcinomas.

Also some genes are relevant to cancers but lung cancer from literature study (*COL5A2* [51], *SPARCL1* [35], *EFEMP2* [39], *MSR1* [49], and *DOCK2* [61]). *APARCL1* and *DOCK2* have shown downregulation in types of cancer [36, 61], while both of them show downregulated gene expressions in T-DM with high occurrences of DNA hypermethylation. Similar to *CRYAB*, *EFEMP2* shows contrary expression patterns in our observation compared to which in gliomas [39]. *EFEMP2* has high occurrences of DNA hypermethylation and downregulated gene expression in totally 20 samples, while 2 samples in T-NDM show little differences when compared to matched normal. *COL5A2* also shows T-DM specific upregulation of gene expression and DNA hypermethylation with high occurrences.

We show the responsive module of *MSR1* in Figure 4(a) as a representation of responsive modules of cancer-related genes. All these genes exhibit significant changes in responsive modules in T-DM when compared to normal and T-NDM.

Besides cancer-related genes, three genes *ARL14*, *CELSR3*, and *WFDC3* are also observed in our list. These three genes show T-DM specific expression changes (Figure 3), and regulatory correlations in responsive modules show significant differences in T-DM when compared to normal and T-NDM (Figures 4(b)–4(d)) which also imply potential roles of the three genes in the tumorigenesis of lung adenocarcinoma.

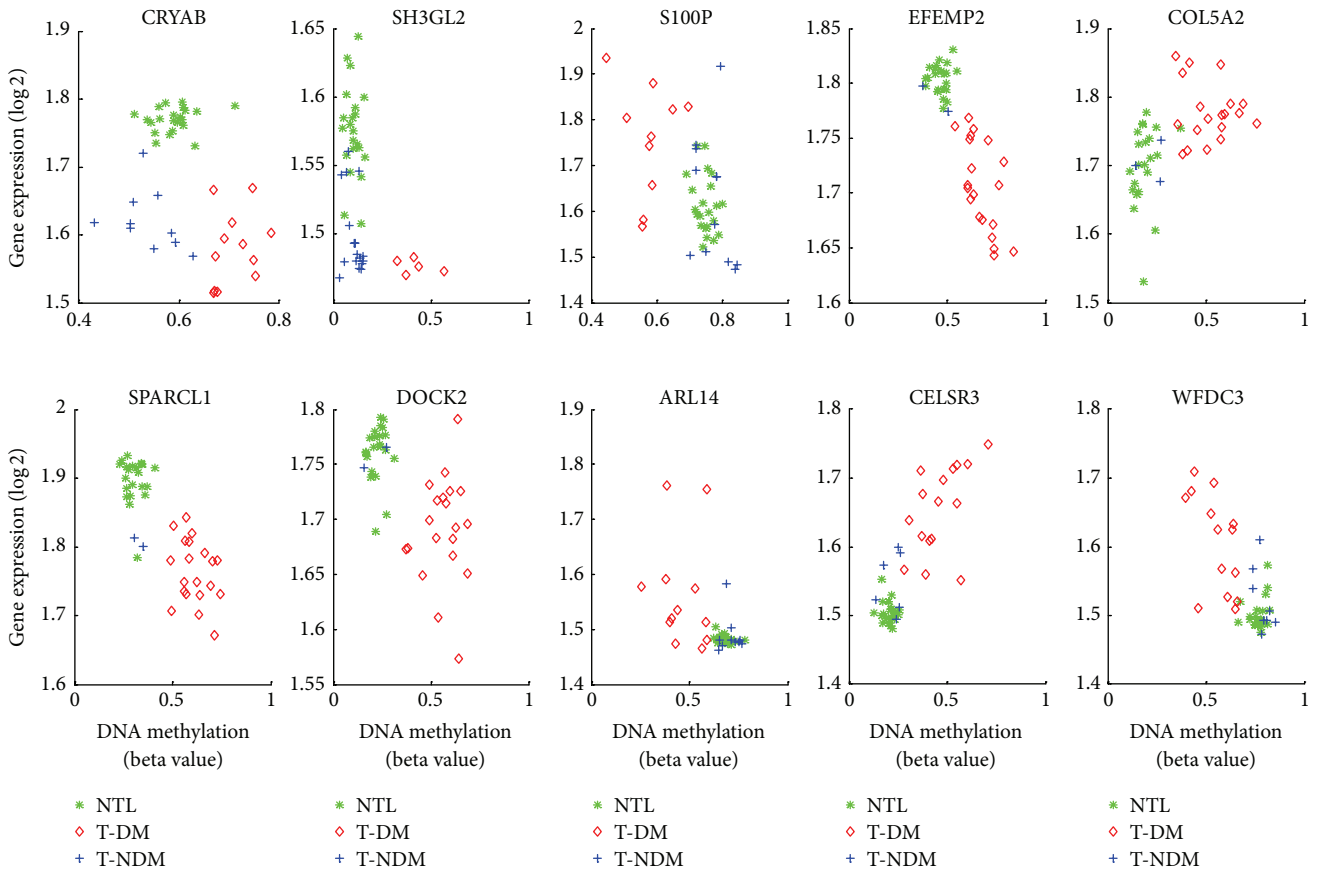


Figure 3: Genes show consistently significant changes in gene expression and DNA methylation in T-DM (red diamond) when compared to normal (green star) and T-NDM (blue plus). Results indicate different distributions of gene expression with altered DNA methylation in three groups of top ranked genes.

All top 30 genes show significant changes in responsive modules in T-DM, while detailed information of the top 30 genes and responsive modules are listed in Supplementary File.

5. Conclusions

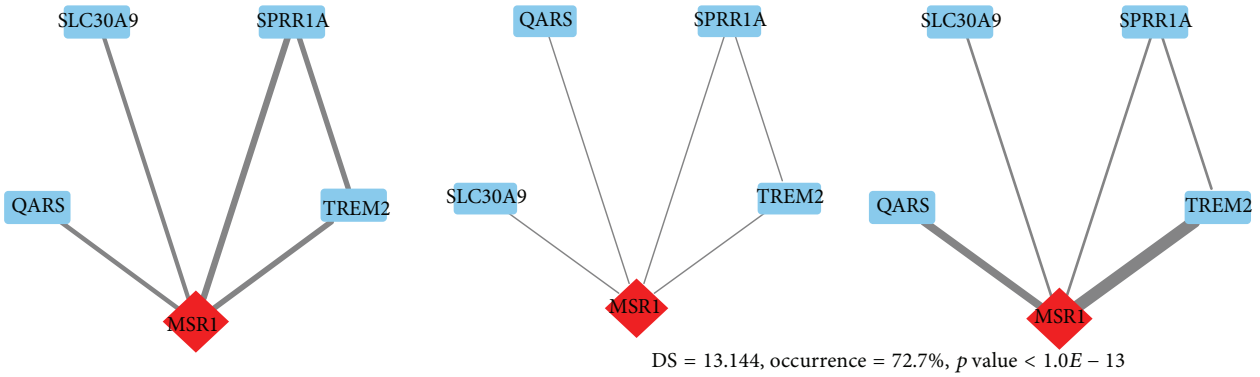
By integration of gene expression and DNA methylation data, we analyzed 22 matched lung adenocarcinoma/nontumor lung pairs for nonsmokers in early stage lung adenocarcinoma. By focusing on differences in gene expression patterns and responsive modules derived from T-DM compared to those in normal and T-NDM, we proposed a pipeline by employing a differential network analysis strategy. Totally, 135 candidate genes are analyzed, and top 30 genes are well studied in this work. All 135 genes are differentially expressed in T-DM when compared to matched normal and T-NDM, while 130 of them show significant changes in regulatory correlations of responsive modules. Literature mining of top 30 genes indicates a high proportion of lung cancer-relevant genes, which implies potential risks of these genes to disturb functions and pathways via differential methylation mechanisms, and further drives the tumorigenesis of lung adenocarcinoma in early stage. In conclusion, we provide a bioinformatics pipeline to identify driver genes with aberrant DNA methylation by fully considering differential expression and network changes in T-DM, normal, and T-NDM. The analysis pipeline can also be employed in identification of driver genes with aberrant DNA methylation of other cancers characterized by paired gene expression and DNA methylation.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

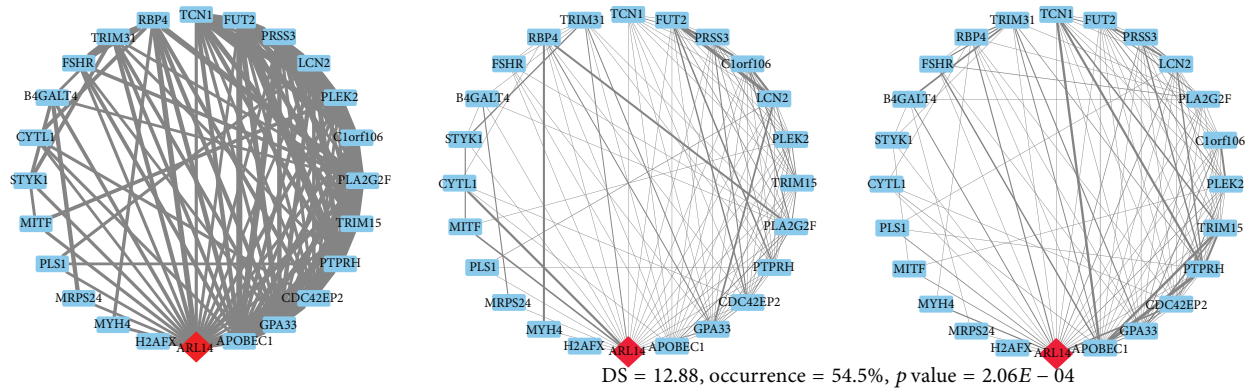
Acknowledgments

This work was supported by National Natural Science Foundation of China (Grants nos. 61532014, 61572287, 61432010, and 61402349), the Fundamental Research Funds for the



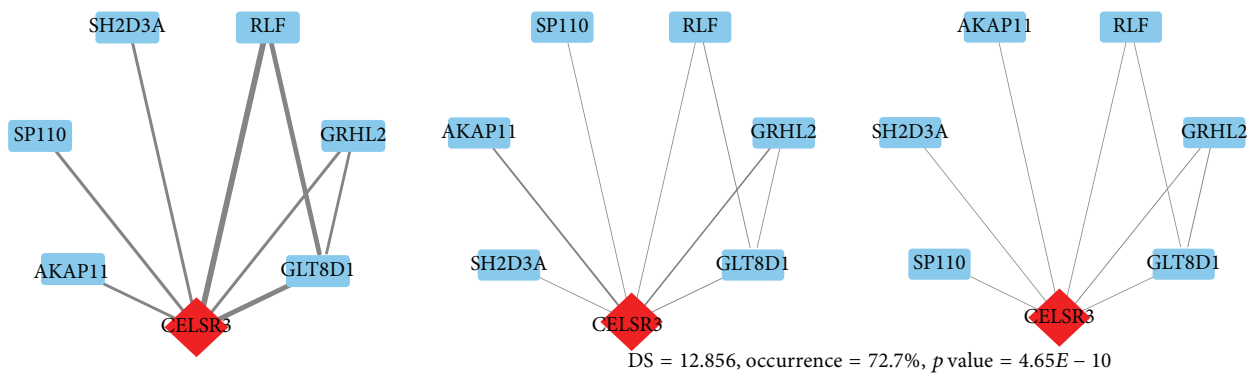
DS = 13.144, occurrence = 72.7%, p value < $1.0E-13$

(a)



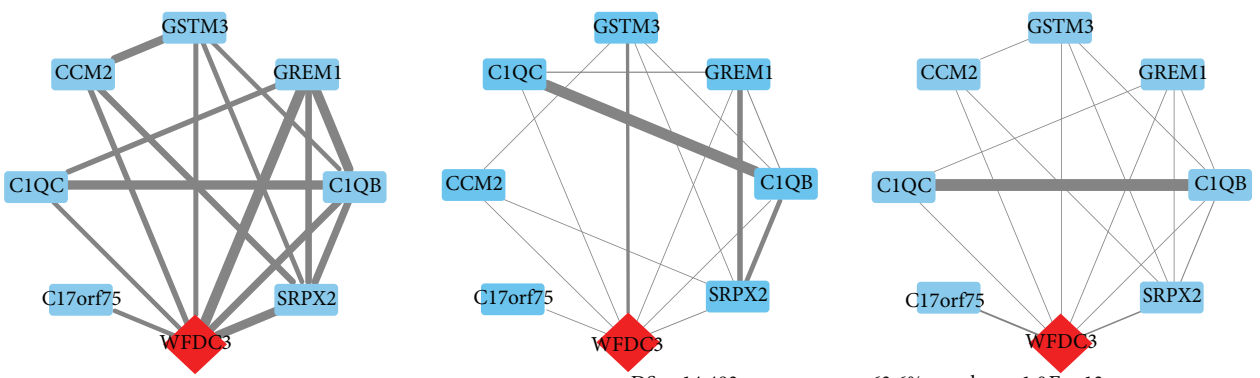
DS = 12.88, occurrence = 54.5%, p value = $2.06E-04$

(b)



DS = 12.856, occurrence = 72.7%, p value = $4.65E-10$

(c)



DS = 14.483, occurrence = 63.6%, p value < $1.0E-13$

(d)

FIGURE 4: Differential representation of responsive modules for *MSR1*, *ARL14*, *CELSR3*, and *WFDC3* in T-DM (left), normal (middle), and T-NDM (right). Significant changes of responsive modules for identified driver genes (red diamond) imply functional alterations of driver genes in tumorigenesis.

Central Universities (Grant no. BDZ021404), and Shandong Provincial Natural Science Foundation of China under Grant no. ZR2015FQ001.

References

[1] H. Zhang and B. Cai, "The impact of tobacco on lung health in China," *Respirology*, vol. 8, no. 1, pp. 17–21, 2003.

[2] J. Ferlay, P. Autier, M. Boniol, M. Heanue, M. Colombet, and P. Boyle, "Estimates of the cancer incidence and mortality in Europe in 2006," *Annals of Oncology*, vol. 18, no. 3, pp. 581–592, 2007.

[3] C.-K. Toh, F. Gao, W.-T. Lim et al., "Never-smokers with lung cancer: epidemiologic evidence of a distinct disease entity," *Journal of Clinical Oncology*, vol. 24, no. 15, pp. 2245–2251, 2006.

[4] Y. Hu and G. Chen, "Pathogenic mechanisms of lung adenocarcinoma in smokers and non-smokers determined by gene expression interrogation," *Oncology Letters*, vol. 10, no. 3, pp. 1350–1370, 2015.

[5] D. De Carvalho, S. Sharma, J. S. You et al., "DNA methylation screening identifies driver epigenetic events of cancer cell survival," *Cancer Cell*, vol. 21, no. 5, pp. 655–667, 2012.

[6] Y. Delpu, P. Cordelier, W. C. Cho, and J. Torrisani, "DNA methylation and cancer diagnosis," *International Journal of Molecular Sciences*, vol. 14, no. 7, pp. 15029–15058, 2013.

[7] D. D. De Carvalho, J. S. You, and P. A. Jones, "DNA methylation and cellular reprogramming," *Trends in Cell Biology*, vol. 20, no. 10, pp. 609–617, 2010.

[8] A. Meissner, "Epigenetic modifications in pluripotent and differentiated cells," *Nature Biotechnology*, vol. 28, no. 10, pp. 1079–1088, 2010.

[9] R. L. Momparler and V. Bovenzi, "DNA methylation and cancer," *Journal of Cellular Physiology*, vol. 183, no. 2, pp. 145–154, 2000.

[10] M. Kulis and M. Esteller, "DNA methylation and cancer," in *Advances in Genetics*, H. Zdenko and U. Toshikazu, Eds., pp. 27–56, Academic Press, 2010.

[11] N. The Cancer Genome Atlas Research, "Comprehensive genomic characterization of squamous cell lung cancers," *Nature*, vol. 489, no. 7417, pp. 519–525, 2012.

[12] I. Balgkouranidou, T. Liloglou, and E. S. Lianidou, "Lung cancer epigenetics: emerging biomarkers," *Biomarkers in Medicine*, vol. 7, no. 1, pp. 49–58, 2013.

[13] S. Kalari and G. P. Pfeifer, "Identification of driver and passenger DNA methylation in cancer by epigenomic analysis," *Advances in Genetics*, vol. 70, pp. 277–308, 2010.

[14] D. Jjingo, A. B. Conley, S. V. Yi, V. V. Lunyak, and I. King Jordan, "On the presence and role of human gene-body DNA methylation," *Oncotarget*, vol. 3, no. 4, pp. 462–474, 2012.

[15] X. Yang, H. Han, D. D. DeCarvalho, F. D. Lay, P. A. Jones, and G. Liang, "Gene body methylation can alter gene expression and is a therapeutic target in cancer," *Cancer Cell*, vol. 26, no. 4, pp. 577–590, 2014.

[16] J. M. Teodoridis, G. Strathdee, and R. Brown, "Epigenetic silencing mediated by CpG island methylation: potential as a therapeutic target and as a biomarker," *Drug Resistance Updates*, vol. 7, no. 4-5, pp. 267–278, 2004.

[17] L. Sigalotti, E. Fratta, S. Coral et al., "Epigenetic drugs as pleiotropic agents in cancer treatment: biomolecular aspects and clinical applications," *Journal of Cellular Physiology*, vol. 212, no. 2, pp. 330–344, 2007.

[18] S. A. Selamat, B. S. Chung, L. Girard et al., "Genome-scale analysis of DNA methylation in lung adenocarcinoma and integration with mRNA expression," *Genome Research*, vol. 22, no. 7, pp. 1197–1211, 2012.

[19] M. Tessema, C. M. Yingling, Y. Liu et al., "Genome-wide unmasking of epigenetically silenced genes in lung adenocarcinoma from smokers and never smokers," *Carcinogenesis*, vol. 35, no. 6, pp. 1248–1257, 2014.

[20] A. Karlsson, M. Jönsson, M. Lauss et al., "Genome-wide DNA methylation analysis of lung carcinoma reveals one neuroendocrine and four adenocarcinoma epitypes associated with patient outcome," *Clinical Cancer Research*, vol. 20, no. 23, pp. 6127–6140, 2014.

[21] T. Sato, E. Arai, T. Kohno et al., "Epigenetic clustering of lung adenocarcinomas based on DNA methylation profiles in adjacent lung tissue: its correlation with smoking history and chronic obstructive pulmonary disease," *International Journal of Cancer*, vol. 135, no. 2, pp. 319–334, 2014.

[22] X. Zeng, X. Zhang, and Q. Zou, "Integrative approaches for predicting microRNA function and prioritizing disease-related microRNA using biological interaction networks," *Briefings in Bioinformatics*, vol. 17, no. 2, pp. 193–203, 2016.

[23] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: a survey," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 55–64, 2016.

[24] W. Huber, A. Von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron, "Variance stabilization applied to microarray data calibration and to the quantification of differential expression," *Bioinformatics*, vol. 18, no. 1, pp. S96–S104, 2002.

[25] J. D. Storey, "A direct approach to false discovery rates," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, vol. 64, no. 3, pp. 479–498, 2002.

[26] E. Lee, H.-Y. Chuang, J.-W. Kim, T. Ideker, and D. Lee, "Inferring pathway activity toward precise disease classification," *PLoS Computational Biology*, vol. 4, no. 11, Article ID e1000217, 2008.

[27] J. J. Faith, B. Hayete, J. T. Thaden et al., "Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles," *PLoS Biology*, vol. 5, no. 1, article e8, 2007.

[28] C. O. Daub, R. Steuer, J. Selbig, and S. Kloska, "Estimating mutual information using B-spline functions—an improved similarity measure for analysing gene expression data," *BMC Bioinformatics*, vol. 5, no. 1, article 118, pp. 1–12, 2004.

[29] C. Olsen, P. E. Meyer, and G. Bontempi, "On the impact of entropy estimation on transcriptional regulatory network inference based on mutual information," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 308959, 2009.

[30] T. Ideker and N. J. Krogan, "Differential network biology," *Molecular Systems Biology*, vol. 8, article 565, 2012.

[31] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, vol. 37, no. 1, pp. 1–13, 2009.

[32] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature Protocols*, vol. 4, no. 1, pp. 44–57, 2009.

[33] A.-L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[34] D. Pastuszak-Lewandoska, K. H. Czarnecka, M. Migdalska-Sęk et al., "Decreased FAM107A expression in patients with non-small cell lung cancer," *Advances in Experimental Medicine and Biology*, vol. 852, pp. 39–48, 2015.

[35] Y. Xiang, Q. Qiu, M. Jiang et al., "SPARCL1 suppresses metastasis in prostate cancer," *Molecular Oncology*, vol. 7, no. 6, pp. 1019–1030, 2013.

[36] P. Li, J. Qian, G. Yu et al., "Down-regulated SPARCL1 is associated with clinical significance in human gastric cancer," *Journal of Surgical Oncology*, vol. 105, no. 1, pp. 31–37, 2012.

[37] A. F. Pla and D. Gkika, "Emerging role of TRP channels in cell migration: from tumor vascularization to metastasis," *Frontiers in Physiology*, vol. 4, article 311, 2013.

[38] H. Qin, Y. Ni, J. Tong et al., "Elevated expression of CRYAB predicts unfavorable prognosis in non-small cell lung cancer," *Medical Oncology*, vol. 31, no. 8, article 142, 2014.

[39] L. Wang, Q. Chen, Z. Chen et al., "EFEMP2 is upregulated in gliomas and promotes glioma cell proliferation and invasion," *International Journal of Clinical and Experimental Pathology*, vol. 8, no. 9, pp. 10385–10393, 2015.

[40] M. Watanabe, N. Komeshima, S. Nakajima, and T. Tsuruo, "MX2, a morpholino anthracycline, as a new antitumor agent against drug-sensitive and multidrug-resistant human and murine tumor cells," *Cancer Research*, vol. 48, no. 23, pp. 6653–6657, 1988.

[41] K. Kobayashi, M. Nishioka, T. Kohno et al., "Identification of genes whose expression is upregulated in lung adenocarcinoma cells in comparison with type II alveolar cells and bronchiolar epithelial cells in vivo," *Oncogene*, vol. 23, no. 17, pp. 3089–3096, 2004.

[42] C. Hartmann, L. Johnk, H. Sasaki, R. B. Jenkins, and D. N. Louis, "Novel PLA2G4C polymorphism as a molecular diagnostic assay for 19q loss in human gliomas," *Brain Pathology*, vol. 12, no. 2, pp. 178–182, 2002.

[43] P. D. Bos, X. H.-F. Zhang, C. Nadal et al., "Genes that mediate breast cancer metastasis to the brain," *Nature*, vol. 459, no. 7249, pp. 1005–1009, 2009.

[44] G. Buccheri and D. Ferrigno, "Lung tumour markers in oncology practice: a study of TPA and CA125," *British Journal of Cancer*, vol. 87, no. 10, pp. 1112–1118, 2002.

[45] L. T. Smith, M. Lin, R. M. Brena et al., "Epigenetic regulation of the tumor suppressor gene TCF21 on 6q23-q24 in lung and head and neck cancer," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 4, pp. 982–987, 2006.

[46] D. Yin, Y. Jia, Y. Yu et al., "SOX17 methylation inhibits its antagonism of Wnt signaling pathway in lung cancer," *Discovery Medicine*, vol. 14, no. 74, pp. 33–40, 2012.

[47] S. Dasgupta, J. S. Jang, C. Shao et al., "SH3GL2 is frequently deleted in non-small cell lung cancer and downregulates tumor growth by modulating EGFR signaling," *Journal of Molecular Medicine (Berlin, Germany)*, vol. 91, no. 3, pp. 381–393, 2013.

[48] N. E. Reticker-Flynn and S. N. Bhatia, "Aberrant glycosylation promotes lung cancer metastasis through adhesion to galectins in the metastatic niche," *Cancer Discovery*, vol. 5, no. 2, pp. 168–181, 2015.

[49] Y. Chen, C. Sullivan, C. Peng et al., "A tumor suppressor function of the *Msr1* gene in leukemia stem cells of chronic myeloid leukemia," *Blood*, vol. 118, no. 2, pp. 390–400, 2011.

[50] K. Lokk, T. Vooder, R. Kolde et al., "Methylation markers of early-stage non-small cell lung cancer," *PLoS ONE*, vol. 7, no. 6, article e39813, 2012.

[51] H. Fischer, R. Stenling, C. Rubio, and A. Lindblom, "Colorectal carcinogenesis is associated with stromal expression of COL11A1 and COL5A2," *Carcinogenesis*, vol. 22, no. 6, pp. 875–878, 2001.

[52] T. Sato, K. Soejima, E. R. I. Arai et al., "Prognostic implication of PTPRH hypomethylation in non-small cell lung cancer," *Oncology Reports*, vol. 34, no. 3, pp. 1137–1145, 2015.

[53] G.-H. Li and J.-F. Huang, "Inferring therapeutic targets from heterogeneous data: HKDC1 is a novel potential therapeutic target for cancer," *Bioinformatics*, vol. 30, no. 6, pp. 748–752, 2014.

[54] K. O. Toyooka, S. Toyooka, A. K. Virmani et al., "Loss of expression and aberrant methylation of the CDH13 (H-cadherin) gene in breast and lung carcinomas," *Cancer Research*, vol. 61, no. 11, pp. 4556–4560, 2001.

[55] M. Okroj, Y.-F. Hsu, D. Ajona, R. Pio, and A. M. Blom, "Non-small cell lung cancer cells produce a functional set of complement factor I and its soluble cofactors," *Molecular Immunology*, vol. 45, no. 1, pp. 169–179, 2008.

[56] D. Schveigert, S. Cicenas, S. Bruzas, N. Samalavicius, Z. Gudleviciene, and J. Didziapetriene, "The value of MMP-9 for breast and non-small cell lung cancer patients' survival," *Advances in Medical Sciences*, vol. 58, no. 1, pp. 73–82, 2013.

[57] J. Wrangle, E. O. Machida, L. Danilova et al., "Functional identification of cancer-specific methylation of CDO1, HOXA9, and TAC1 for the diagnosis of lung cancer," *Clinical Cancer Research*, vol. 20, no. 7, pp. 1856–1864, 2014.

[58] M. Alavi, V. Mah, E. L. Maresh et al., "High expression of AGR2 in lung cancer is predictive of poor survival," *BMC Cancer*, vol. 15, no. 1, article 655, 2015.

[59] B. Bartling, G. Rehbein, W. D. Schmitt, H.-S. Hofmann, R.-E. Silber, and A. Simm, "S100A2-S100P expression profile and diagnosis of non-small cell lung carcinoma: impairment by advanced tumour stages and neoadjuvant chemotherapy," *European Journal of Cancer*, vol. 43, no. 13, pp. 1935–1943, 2007.

[60] G. Rehbein, A. Simm, H.-S. Hofmann, R.-E. Silbar, and B. Bartling, "Molecular regulation of S100P in human lung adenocarcinomas," *International Journal of Molecular Medicine*, vol. 22, no. 1, pp. 69–77, 2008.

[61] H. Nishihara, M. Maeda, A. Oda et al., "DOCK2 associates with CrkL and regulates Rac1 in human leukemia cell lines," *Blood*, vol. 100, no. 12, pp. 3968–3974, 2002.

[62] X. Guan, Z. Liao, H. Ma et al., "TNFRSF1B +676 T>G polymorphism predicts survival of non-Small cell lung cancer patients treated with chemoradiotherapy," *BMC Cancer*, vol. 11, article 447, 2011.

[63] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," *Nucleic Acids Research*, vol. 44, no. 1, pp. D457–D462, 2016.

[64] A. Subramanian, P. Tamayo, V. K. Mootha et al., "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 43, pp. 15545–15550, 2005.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 1358142, 7 pages
http://dx.doi.org/10.1155/2016/1358142

Research Article

Optimization to the Culture Conditions for *Phellinus* Production with Regression Analysis and Gene-Set Based Genetic Algorithm

Zhongwei Li,[1] Yuezhen Xin,[1] Xun Wang,[1,2] Beibei Sun,[1] Shengyu Xia,[2] Hui Li,[2] and Hu Zhu[2]

[1]*College of Computer and Communication Engineering, China University of Petroleum, Qingdao, Shandong 266580, China*
[2]*Center for Bioengineering and Biotechnology, China University of Petroleum, Qingdao, Shandong 266580, China*

Correspondence should be addressed to Hu Zhu; zhuhu@upc.edu.cn

Received 28 May 2016; Revised 11 July 2016; Accepted 16 July 2016

Academic Editor: Quan Zou

Phellinus is a kind of fungus and is known as one of the elemental components in drugs to avoid cancers. With the purpose of finding optimized culture conditions for *Phellinus* production in the laboratory, plenty of experiments focusing on single factor were operated and large scale of experimental data were generated. In this work, we use the data collected from experiments for regression analysis, and then a mathematical model of predicting *Phellinus* production is achieved. Subsequently, a gene-set based genetic algorithm is developed to optimize the values of parameters involved in culture conditions, including inoculum size, PH value, initial liquid volume, temperature, seed age, fermentation time, and rotation speed. These optimized values of the parameters have accordance with biological experimental results, which indicate that our method has a good predictability for culture conditions optimization.

1. Introduction

Phellinus is a kind of fungus having great medicinal value, since it is known as one of the elemental components in drugs with functions of avoiding cancers [1, 2]. *Phellinus* flavonoids are one of the most popular parasitifers of *Phellinus* in nature [3], and the research on *Phellinus* focuses on polysaccharides, proteoglycans medicinal mechanism, composition, and so forth, which are mostly extracted from the fruiting bodies of *Phellinus* flavonoids [4]. *Phellinus* rarely exists in the wild environment [5], and it becomes a promising research branch to cultivate it in the laboratory. With mycelial growth by liquid fermentation, the fermentation broth flavonoids, polysaccharides, alkaloids, and other active substances can be produced, which have high level physical activity, short fermentation period, and mass productions, thus providing a possible way of producing *Phellinus* in the laboratory [6]. In recent years, updated machine learning approaches (see, e.g., [7, 8]) have been developed and applied in biological data processing.

From the understanding of the wild conditions of *Phellinus*, it is believed that PH value, temperature, and fermentation time have effect on the productions. Also, in general biochemical experiments, we need to consider the inoculum size, initial liquid volume, seed age, and rotation speed. In the laboratory, plenty of experiments have been designed and operated for maximizing the *Phellinus* production. The methods can be separated into two major groups.

(i) With biological technologies: it used optimum media on mycelial growth of *Phellinus* in [9] and liquid fermentation technology to cultivate *Phellinus* in [10]. Active ingredients in *Phellinus* and polysaccharide metabolism regulation are designed in [11].

(ii) With mathematical models: some researches focus on building mathematical models for the progress of producing *Phellinus* by differential equations [12], metabolic path and network [13], and complex network models [14].

Artificial algorithms and models have been used in the bioprocess, particularly for the optimization of culture conditions. In [15], artificial neural network (ANN) is used to optimize the extraction process of azalea flavonoids. Neural networks combined with evolutionary algorithms have been used to optimize the experimental environment, such that neural network and particle swarm optimization method were used for finding optimized culture conditions to maximize the production of Pleuromutilin from *Pleurotus mutilus* in [16]. Recently, with the increment of biological data, regression analysis becomes a useful tool for the data analysis. In [17] the method of fitting models to biological data using linear and nonlinear regression is proposed, where some multivariate statistical analysis strategies from [18, 19] are formulated to be helpful and useful for biologists. These results give us hints of using regression analysis and artificial algorithms to optimize the culture conditions for *Phellinus* production. And, to the best of our knowledge, few work focuses on the optimization of culture conditions to maximize the production of *Phellinus* in the laboratory.

In this work, we start from operating 45 experiments for producing *Phellinus* from *Phellinus* flavonoids with different culture conditions, involving parameters PH value, temperature and fermentation time, inoculum size, initial liquid volume, seed age, and rotation speed. With the data collected during the experiments, we use regression analysis method to create a mathematical model, which can forecast the flavonoid yield and the most important element to the production of *Phellinus*. After that, a gene-set based genetic algorithm (GA) is proposed to optimize the culture condition, where the obtained mathematic model is used as fitness function for the evolution of individuals. Data experimental results show that predicted optimal values of the parameters have accordance with biological experimental results, which indicate that our method has a good predictability for culture conditions optimization.

2. Data Collected from Experiments

In this section, biological experiments are performed for finding optimal value of certain single factor.

In Table 1, experiments are operated for collecting data. In rows 1–14, it is associated with experiments with PH values ranging from 1 to 14, where the temperature is fixed to 28°C, initial volume is set to be 100 mL, the rotation speed is 140 r/m, and seed age is 8 days. Rows 15 to 20 are 6 experiments with initial volume ranging from 40 mL to 140 mL, where PH value is set to be 6, the best one obtained from experiments with PH values ranging from 1 to 14.

In Table 2, experiments with including inoculum ranging from 2% to 16% and temperature ranging from 25°C to 40°C are performed. In Table 3 the situations on experiments with fermentation time ranging from 1 to 12 hours are shown. From the in total 45 experiments, we collect data of culture conditions for production of *Phellinus*. Different culture conditions have a fundamental influence on the production of *Phellinus*, but the optimized culture conditions remain unknown.

3. Methods

We consider here using regression analysis and gene-set based genetic algorithm to find the optimized culture conditions for maximizing the production of *Phellinus*. In general, we convert the data collected in Section 2 to construct a mathematical model by regression analysis. And then, the obtained model can be used as fitness function for optimizing the culture condition with gene-set based genetic algorithm.

3.1. Regression Analysis. In statistical modeling, regression analysis is a statistical process for estimating the relationships among variables [20]. Regression analysis is one of the extremely versatile data analysis methods, which is appropriated to establish dependencies between variables based on observational data and widely used to analyze the data inherent law and to predict the result. Regression analysis can be divided into linear regression and nonlinear regression analysis [21], according to the type of relationship between the independent variables and dependent variables. In general, the relationship between variables is determined by the independent variables and dependent selected variables, by which regression models can be made. After that, it is used to solve the various parameters of the model based on the measured data and then evaluate whether the regression model can fit the observed data. If the model can fit the data well, then the model can be used to further predict based arguments [22]. The regression analysis is composed of the following steps [23, 24].

Regression analysis is widely used in data mining, particularly for biological data analysis in recent years, with the purpose of finding a feasible statistical law by the large amount of data of experiments. The general process is given as follows.

Step 1. Determine the variables.

Step 2. Establish the prediction model.

Step 3. Relate analysis.

Step 4. Calculate the prediction error.

Step 5. Determine the predicted value.

From the data collected in Section 2, it consists of seven independent variables and one dependent variable. The seven independent variables are inoculum size, PH values, initial liquid volume, temperature, seed age, fermentation time, and rotation speed. And, the dependent variable is flavonoid yield. From the observation of the experiments, it is found that some culture conditions are not suitable for production of *Phellinus*. These data are taken as extreme data are removed from regression analysis. Extreme data refers to the data which were measured in extreme experimental environment. Also duplicate data were cancelled. Only the following data are selected in regression analysis.

(i) Inoculum size 0.5%~1.2%.

(ii) PH 5~7.

Table 1: Experiments with PH values ranging from 1 to 14 and initial volume ranges from 40 mL to 140 mL.

Phellinus production	PH	Temp.	Initial volume	Rotation speed	Including inoculum	Seed age	Fermentation time
45.929	1	28°C	100 mL	140	5%	8	8
35.077	2	28°C	100 mL	140	5%	8	8
45.654	3	28°C	100 mL	140	5%	8	8
534.39	4	28°C	100 mL	140	5%	8	8
702.81	5	28°C	100 mL	140	5%	8	8
1467.7	6	28°C	100 mL	140	5%	8	8
189.20	7	28°C	100 mL	140	5%	8	8
91.049	8	28°C	100 mL	140	5%	8	8
60.841	9	28°C	100 mL	140	5%	8	8
57.255	10	28°C	100 mL	140	5%	8	8
43.238	11	28°C	100 mL	140	5%	8	8
36.288	12	28°C	100 mL	140	5%	8	8
20.943	13	28°C	100 mL	140	5%	8	8
22.306	14	28°C	100 mL	140	5%	8	8
508.495	6	28°C	40 mL	140	10%	8	8
900.662	6	28°C	60 mL	140	10%	8	8
1273.594	6	28°C	80 mL	140	10%	8	8
1153.937	6	28°C	100 mL	140	10%	8	8
1123.330	6	28°C	120 mL	140	10%	8	8
1088.064	6	28°C	140 mL	140	10%	8	8

Table 2: Experiments with including inoculum ranging from 2% to 16% and temperature ranging from 25°C to 40°C.

Phellinus production	PH	Temp.	Initial volume	Rotation speed	Including inoculum	Seed age	Fermentation time
546.609	6	28°C	100 mL	140	2%	8	8
606.345	6	28°C	100 mL	140	4%	8	8
1320.794	6	28°C	100 mL	140	6%	8	8
1447.519	6	28°C	100 mL	140	8%	8	8
1841.729	6	28°C	100 mL	140	10%	8	8
1631.990	6	28°C	100 mL	140	12%	8	8
481.1172	6	28°C	100 mL	140	14%	8	8
449.5187	6	28°C	100 mL	140	16%	8	8
1145.669	6	25°C	40 mL	140	10%	8	8
1506.055	6	30°C	60 mL	140	10%	8	8
1374.982	6	35°C	80 mL	140	10%	8	8
875.341	6	40°C	100 mL	140	10%	8	8

Table 3: Experiments with fermentation time ranging from 1 to 12 hours.

Phellinus production	PH	Temp.	Initial volume	Rotation speed	Including inoculum	Seed age	Fermentation time
56.606	6	28°C	100 mL	150	2%	8	1
83.435	6	28°C	100 mL	150	4%	8	2
303.984	6	28°C	100 mL	150	6%	8	3
449.919	6	28°C	100 mL	150	8%	8	4
777.331	6	28°C	100 mL	150	10%	8	5
1103.987	6	28°C	100 mL	150	12%	8	6
1619.554	6	28°C	100 mL	150	14%	8	7
1597.995	6	28°C	100 mL	150	16%	8	8
1546.336	6	28°C	100 mL	150	10%	8	9
1502.487	6	28°C	100 mL	150	10%	8	10
1489.364	6	28°C	100 mL	150	10%	8	11
1465.664	6	28°C	100 mL	150	10%	8	12

Table 4: Regression analysis results.

	Sum of square	df	Mean square	F	R	R-squared	Standard error
Regression	3796787.42	14	249770.53	5.234	0.93	0.88	218.48
Residuals	47719.89	10	47719.54				
Sum	3973983.26	24					

(iii) Initial liquid volume 60~100 mL.

(iv) Temperature 25~30°C.

(v) Seed age 4~9 days.

(vi) Fermentation time 6~12 days.

(vii) Rotation speed 140~200 r/m.

After data filtering, a statistical model is made to represent these data. It is known that there is a correlation between these data relationships, so we applied linear regression analysis to fit them. At this stage, a lot of models were tested one by one with IBM SPSS software and response surface methodology. The statistical model is $Y = A1 * X^2(1) + \cdots + A7 * X^2(7) + B1 * X(1) * X(2) + B2 * X(1) * X(3) + B3 * X(1) * X(4) + B4 * X(1) * X(5) + B5 * X(1) * X(6) + B7 * X(1) * X(7) + B8 * X(2) * X(3) + B9 * X(2) * X(4) + B10 * X(2) * X(5) + B11 * X(2) * X(6) + B12 * X(2) * X(7) + B13 * X(3) * X(4) + B14 * X(3) * X(5) + B15 * X(3) * X(6) + B16 * X(3) * X(7) + B17 * X(4) * X(5) + B18 * X(4) * X(6) + B19 * X(4) * X(7) + B20 * X(5) * X(6) + B21 * X(5) * X(7) + B22 * X(6) * X(7) + C$, where Y is a dependent variable of the flavonoid yield, $X(1), X(2), \ldots, X(7)$ are the seven independent variables associated with inoculum size, PH value, initial liquid volume, temperature, seed age, fermentation time, and rotation speed, respectively, and A, B, and C are real numbers.

Although the relationship between the data may not be linear, we can put squared term for a type of data into these data. If this term is useful it will be retained after linear regression analysis; otherwise, the data will be deleted.

In the regression analysis, it needs to focus on the values of R-squared and the significance of correlation coefficients for regulating the model. We use the regression analysis tools in the IBM SPSS, setting regression coefficients as estimated (E) and selecting the display model fit (M). Set the stepping method criteria as use of probability F, entry (E) as 0.5, and removal (M) as 0.10. After regression analysis, we can get the results as shown in Table 4.

It is obtained that significance = 0.006 < 0.05; that is, the regression results are obvious. R-squared value is 0.88, which means that the model is valid for fitting the 88% data. We get the statistical model: $Y = 3662.278 * x(1) - 4263.361 * (x(1)^2) + 11737.986 * x(2) - 999.556 * x(2)^2 - 0.238 * x(3)^2 + 3353.461 * x(4) - 59.662 * x(4)^2 - 420.854 * x(5) + 42.495 * x(5)^2 + 966.796 * x(6) - 53.489 * x(6)^2 + 27.213 * x(7) - 0.234 * x(7)^2 + 0.434 * x(3) * x(7) - 86781.046$.

3.2. Gene-Set Based Genetic Algorithm. Genetic algorithm (GA) was first proposed by J. Holland in 1975 [25, 26], whose general process is shown in Figure 1. In the mutation operation, if a short segment is selected in a mutation possibility and replaced by another segment, then the gene-set based GA is achieved [27].

In gene-set based GA, a chromosome is treated as a set of gene-sets, instead of a set of genes as in classical GAs. It starts with gene-sets of the largest size equal to half the chromosome length. It is most appropriate to genetics model because each gene-set represents a set of adjacent parameters of certain factor of the culture conditions.

It is noted that, in the selection, only the winning individuals from the population can be selected. Select operators are also known as reclaimed operator (reproduction operator), whose purpose is to optimize the selection of individuals (or solutions) to the next generation. Population can be updated by fitness ratio method and random sampling method to traverse, local selection. Cross operator refers to the part of the structure of the two parent individuals to generate new recombinant replacing individual operation. Variation is to make GA have local random search capability. When the GA crossover neighborhood is close to the optimal solution, the use of such a mutation operator of local random search capability can accelerate the convergence to the optimal solution.

The statistical model obtained by regression analysis is used as the fitness function here, and gene-set based GA is used to optimize the culture condition for maximizing the production of *Phellinus*. The data simulation is achieved by gatool in MATLAB. In the data experiments, we use a binary string composed of 7 segments to represent an individual in GA population, where each segment is associated with the value of one of the 7 parameters for the culture condition. Initial population size is 50, and cross rate is set to 0.8. Mutation rate is set to be 0.01, and selection method is roulette wheel selection. If the time is long enough then the GA process will halt by meeting the stopping conditions, such as generations limit or fitness limit.

After 156 iterations the gene-set based GA process returns the best individual and shuts down the process in Figure 2.

After the regression analysis and GA process, an optimized culture condition is obtained, shown in Table 5.

The results obtained by our method have accordance with experimental experience in literature of *Phellinus* growth environmental studies. Specifically, the suitable environment is neutral acidic environment, about PH value 6. The appropriate temperature range is from 22°C to 28°C [10]. Seed age and fermentation time of species vary due to the strain [3, 28, 29]. These optimized values of the parameters have accordance with biological experimental results, which indicate that our method has a good predictability for culture conditions optimization.

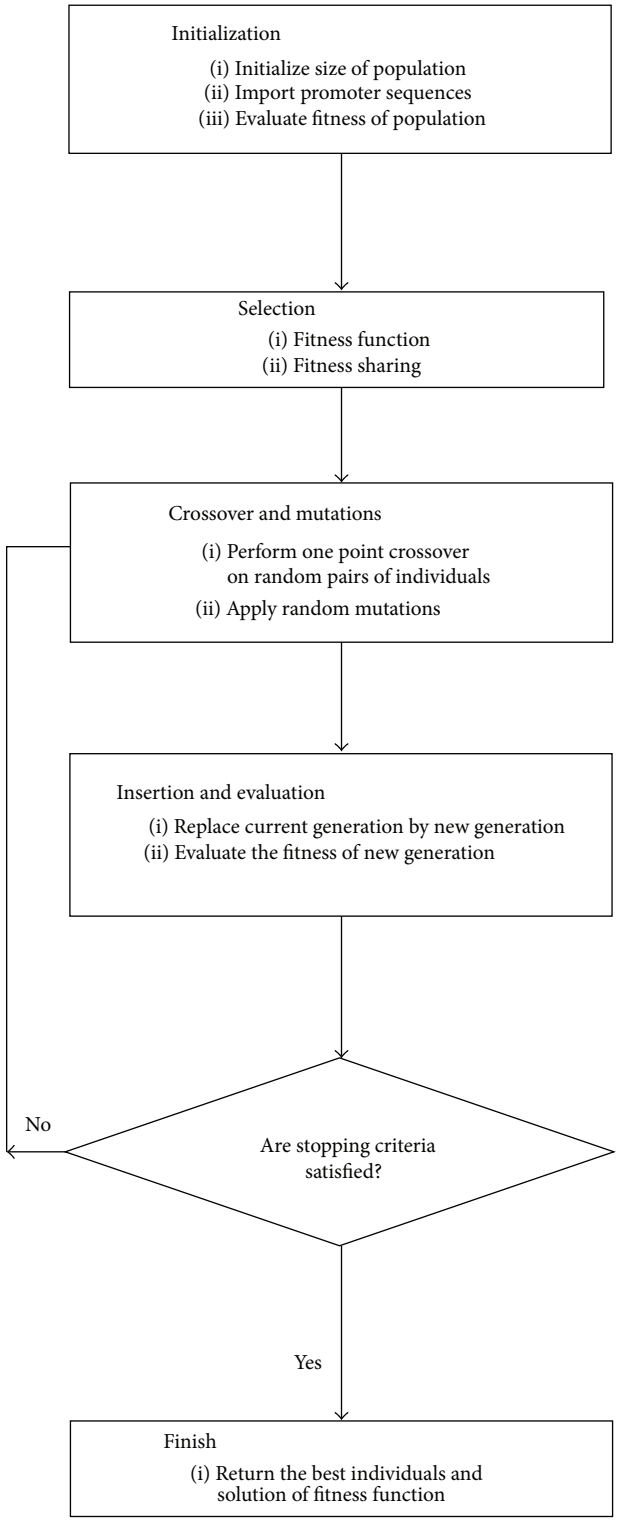


Figure 1: GA process.

4. Conclusion

In this work, 45 experiments are firstly operated for collecting data related to the production of *Phellinus* from *Phellinus* flavonoids. We use regression analysis method to create a mathematical model with the collected data, and then a gene-set based GA is proposed to optimize the culture condition, where the obtained mathematic model is used as fitness function for the evolution of individuals. In the comparison results, it is believed that PH value is credible and

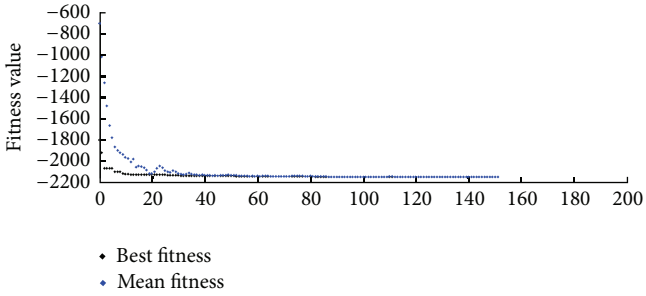


Figure 2: GA best fitness.

Table 5: Optimized culture conditions.

Type	Experiment data	Computer data
Inoculum size	10%	12%
PH	6	5.8
Initial liquid volume	100 mL	100 mL
Temperature	28°C	28°C
Age	8	9
Fermentation time	8	9
Rotation speed	150	150
Flavonoid yield	2164.512	2150.128

the temperature is also within the appropriate temperature range. Taking into account environmental factors in the laboratory, the temperature value we predicted is also reliable. The seed age and fermentation time predicted are 9, close to the original data 8. Data experimental results show that predicted optimal values of the parameters have accordance with biological experimental results, which indicate that our method has a good predictability for culture conditions optimization.

Neural-like computing models, such as artificial neural networks [30], spiking neural networks [31], and spiking neural P systems [32–34], have been successfully used in pattern recognition and engineering practice. It is of interest to use these neural-like computing models for optimizing culture conditions for *Phellinus* production. Our work would also guide for the "Precision Medicine" with personal SNP data [35] and other tasks in bioinformatics [21, 22].

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The research is under the auspices of National Natural Science Foundation of China (nos. 41276135, 31172010, 61272093, 61320106005, 61402187, 61502535, 61572522, and 61572523), Program for New Century Excellent Talents in University (NCET-13-1031), 863 Program (2015AA020925), Fundamental Research Funds for the Central Universities (R1607005A), and China Postdoctoral Science Foundation funded project (2016M592267).

References

[1] T. Zhu, J. Guo, L. Collins et al., "Phellinus linteus activates different pathways to induce apoptosis in prostate cancer cells," *British Journal of Cancer*, vol. 96, no. 4, pp. 583–590, 2007.

[2] D. Sliva, A. Jedinak, J. Kawasaki, K. Harvey, and V. Slivova, "*Phellinus linteus* suppresses growth, angiogenesis and invasive behaviour of breast cancer cells through the inhibition of AKT signalling," *British Journal of Cancer*, vol. 98, no. 8, pp. 1348–1356, 2008.

[3] Y. Wang, J.-X. Yu, C.-L. Zhang et al., "Influence of flavonoids from *Phellinus igniarius* on sturgeon caviar: antioxidant effects and sensory characteristics," *Food Chemistry*, vol. 131, no. 1, pp. 206–210, 2012.

[4] G. Xia, Y. Ge, and H. Fu, "Research on the extraction of total flavonoids from *Phellinus vaninii* with ultrasonic-assisted technique," *Journal of Jiangsu University*, vol. 20, no. 1, pp. 40–41, 2010.

[5] H. H. Doğan and M. Karadelev, "*Phellinus sulphurascens* (Hymenochaetaceae, Basidiomycota): a very rare wood-decay fungus in Europe collected in Turkey," *Turkish Journal of Botany*, vol. 33, no. 3, pp. 239–242, 2009.

[6] W. Liu, *Study on the metabolic regulation of flavones produced by medicinal fungus Phellinus igniarius [M.S. thesis]*, 2012.

[7] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.

[8] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[9] S. Zhong, Y. G. Li, and J. Q. Zhu, "Optimum media on mycelial growth of *Phellinus*," *Zhejiang Agricultural Sciences*, vol. 1, pp. 173–175, 2011.

[10] S. Li, Y. X. Ding, J. Xu, and M. W. Zhao, "Optimization for medium compositions for intracellular polysaccharide of *Phellinus baumii* in submerged culture," *Food Science*, vol. 11, pp. 236–240, 2006.

[11] X. Guo, X. Zou, and M. Sun, "Optimization of extraction process by response surface methodology and preliminary characterization of polysaccharides from *Phellinus igniarius*," *Carbohydrate Polymers*, vol. 80, no. 2, pp. 345–350, 2010.

[12] X.-K. Ma, L. Li, E. C. Peterson, T. Ruan, and X. Duan, "The influence of naphthaleneacetic acid (NAA) and coumarin on flavonoid production by fungus *Phellinus* sp.: modeling of production kinetic profiles," *Applied Microbiology and Biotechnology*, vol. 99, no. 22, pp. 9417–9426, 2015.

[13] N. W. Hanson, K. M. Konwar, A. K. Hawley, T. Altman, P. D. Karp, and S. J. Hallam, "Metabolic pathways for the whole community," *BMC Genomics*, vol. 15, no. 1, article 619, 2014.

[14] M. Kim, G. Kim, B. Nam et al., "Development of species-specific primers for rapid detection of *Phellinus linteus* and *P. baumii*," *Mycobiology*, vol. 33, no. 2, pp. 104–108, 2005.

[15] M. Zhang, D. R. Pan, and F. Zhou, "BP neural network extraction process by orthogonal beautiful azalea flavonoids," *Journal of Xinyang Normal University*, vol. 2, pp. 261–264, 2011.

[16] L. Khaouane, C. Si-Moussa, S. Hanini, and O. Benkortbi, "Optimization of culture conditions for the production of pleuromutilin from pleurotus mutilus using a hybrid method based on central composite design, neural network, and particle swarm optimization," *Biotechnology and Bioprocess Engineering*, vol. 17, no. 5, pp. 1048–1054, 2012.

[17] L. Harvey and C. Arthur, *Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting*, Oxford University Press, 2004.

[18] S. Hilary, *Multivariate Statistical Analysis for Biologists*, John Wiley & Sons, 1964.

[19] S. Robert and F. Rohlf, "The principles and practice of statistics in biological research," in *Multivariate Statistical Analysis for Biologists*, Methuen, London, UK, 1969.

[20] H. L. Seal, *Multivariate Statistical Analysis for Biologists*, Methuen, London, UK, 1964.

[21] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. 1, pp. W65–W71, 2015.

[22] R. Wang, Y. Xu, and B. Liu, "Recombination spot identification based on gapped k-mers," *Scientific Reports*, vol. 6, Article ID 23934, 2016.

[23] F. John, *Applied Regression Analysis, Linear Models, and Related Methods*, Sage, 1997.

[24] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*, vol. 936, John Wiley & Sons, 2012.

[25] D. Lawrence, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[26] D. Beasley, R. R. Martin, and D. R. Bull, "An overview of genetic algorithms: part 1. Fundamentals," *University Computing*, vol. 15, no. 2, pp. 58–69, 1993.

[27] T.-P. Hong, M.-T. Wu, Y.-F. Tung, and S.-L. Wang, "Using escape operations in gene-set genetic algorithms," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '07)*, pp. 3907–3911, Montreal, Canada, October 2007.

[28] J. Luo, J. Liu, C. Ke et al., "Optimization of medium composition for the production of exopolysaccharides from *Phellinus baumii* Pilát in submerged culture and the immuno-stimulating activity of exopolysaccharides," *Carbohydrate Polymers*, vol. 78, no. 3, pp. 409–415, 2009.

[29] D. B. Harper and J. T. Kennedy, "Effect of growth conditions on halomethane production by *Phellinus species*: biological and environmental implications," *Journal of General Microbiology*, vol. 132, no. 5, pp. 1231–1246, 1986.

[30] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[31] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[32] T. Song, J. Xu, and L. Pan, "On the universality and non-universality of spiking neural P systems with rules on synapses," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 960–966, 2015.

[33] T. Song, Q. Zou, X. Liu, and X. Zeng, "Asynchronous spiking neural P systems with rules on synapses," *Neurocomputing*, vol. 151, no. 3, pp. 1439–1445, 2015.

[34] X. Wang, T. Song, F. Gong, and P. Zheng, "On the computational power of spiking neural P systems with self-organization," *Scientific Reports*, vol. 6, Article ID 27624, 2016.

[35] P. Li, M. Guo, C. Wang, X. Liu, and Q. Zou, "An overview of SNP interactions in genome-wide association studies," *Briefings in Functional Genomics*, vol. 14, no. 2, pp. 143–155, 2015.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 4374603, 6 pages
http://dx.doi.org/10.1155/2016/4374603

Research Article

A Computational Method for Optimizing Experimental Environments for *Phellinus igniarius* via Genetic Algorithm and BP Neural Network

Zhongwei Li,[1] Beibei Sun,[1] Yuezhen Xin,[1] Xun Wang,[1,2] and Hu Zhu[2]

[1]*College of Computer and Communication Engineering, China University of Petroleum, Qingdao, Shandong 266580, China*
[2]*Center for Bioengineering and Biotechnology, China University of Petroleum, Qingdao, Shandong 266580, China*

Correspondence should be addressed to Hu Zhu; zhuhu@upc.edu.cn

Received 29 May 2016; Revised 11 July 2016; Accepted 13 July 2016

Academic Editor: Quan Zou

Flavones, the secondary metabolites of *Phellinus igniarius* fungus, have the properties of antioxidation and anticancer. Because of the great medicinal value, there are large demands on flavones for medical use and research. Flavones abstracted from natural *Phellinus* can not meet the medical and research need, since *Phellinus* in the natural environment is very rare and is hard to be cultivated artificially. The production of flavones is mainly related to the fermentation culture of *Phellinus*, which made the optimization of culture conditions an important problem. Some researches were made to optimize the fermentation culture conditions, such as the method of response surface methodology, which claimed the optimal flavones production was 1532.83 μg/mL. In order to further optimize the fermentation culture conditions for flavones, in this work a hybrid intelligent algorithm with genetic algorithm and BP neural network is proposed. Our method has the intelligent learning ability and can overcome the limitation of large-scale biotic experiments. Through simulations, the optimal culture conditions are obtained and the flavones production is increased to 2200 μg/mL.

1. Introduction

Phellinus is an ancient Chinese medicine, which has high medicinal value. Resent research confirmed that flavones, the secondary metabolites of *Phellinus*, can improve human immune system, reduce the side effects of anticancer agents, and relieve the reaction of patients to radiotherapy or chemotherapy [1]. In addition, flavones have positive effect on irregular menstruation and other gynecological diseases of female. There are large demands on the production of flavones, but the natural *Phellinus* is very rare. The artificial culture of *Phellinus* is hard to implement, because of the lack of culture technology and the long growth cycle of *Phellinus*. In 2008, Zeng et al. introduced a breeding method of *Phellinus* by protoplast fusion [2]. Considering the limited production of *Phellinus*, it is necessary to develop the extraction process of flavones from *Phellinus*. Fermentation is usually used to produce the secondary metabolites of fungus, such as ethanol extracts of *Phellinus baumii* [3]. It should be noticed that different fermentation methods can generate secondary metabolites with different biological activities. Currently, the production of flavones is mainly based on the fermentation culture of *Phellinus*.

In order to increase the production of *Phellinus*, the fermentation culture conditions are considered carefully, such as the fermentation temperature, the PH value, the rotation speed of the centrifuge, the inoculation volume, and the seed age. In the meantime, the composition of medium should also be taken into consideration. The multiple variables of fermentation conditions and culture mediums make the optimization by biotic experiments a hard problem. Zhu et al. took the fermentation temperature, the inoculum size, the rotation speed of the centrifuge, and bottling capacity as the independent variables and the fermentation yield as a dependent variable [4]. The quadratic regression orthogonal rotating combination design method was used to build the model for *Phellinus linteus* fermentation process, and the following optimal fermentation conditions of *Phellinus*

linteus were obtained: the bottling volume is 120 mL, the inoculum size is 17 mL, the temperature is 26°C, and the rotation speed of the centrifuge is 135 r/min. At this time, the theoretical extreme value of fermentation mycelium production was 24.51 mg/mL. Other researches were focused on the fermentation parameters, such as the dosage of carbon source and nitrogen source. In 2010, Zhu et al. used response surface methodology and gave out the optimum liquid fermentation conditions as follows: the concentration of corn starch is 0.5%, the concentration of yeast extract is 2%, the concentration of VB_1 is 0.1%, the period of fermentation is 6 days, and the production of mycelium (dry weight) of *P. linteus* is 18.43 g/L [5, 6]. It should be noticed that these researches were based on single-factor experiments, which made the outcomes rely on some presented parameters. Some machine learning strategies [7, 8] have been applied in solving problem with multiple variables and rely on fewer biotic experiments [9, 10].

In this work, we focus on the optimization of fermentation conditions, which include the concentrations of glucose, maltose, mannitol, corn powder, yeast extract, copper sulfate, sodium chloride, ferrous sulfate, and vitamin B_1. A hybrid algorithm combined by genetic algorithm (GA) and artificial neural network (ANN) is introduced in this paper. This new algorithm has the intelligent learning ability and can overcome the limitation of large-scale biotic experiments. Through simulations, the optimal culture conditions are obtained and the flavones production is increased to 2200 μg/mL.

2. Method

In this section, our method for optimizing the fermentation condition by BP neural network and genetic algorithm is introduced.

2.1. BP Neural Network. The back propagation (BP) neural network proposed in [11] is a kind of former multiway propagation network, with an input layer, an intermediate layer (hidden layer), and an output layer. The model is now known as one of the most widely applied neural network models in practice. Any neuron from the input layer has a connection to every neuron in the hidden layer, while any neuron in the hidden layer connects with every output neuron. There is no connection between each pair of neurons in the same layer [12]. BP network can be used to learn and store the relationship between the input and output. In the learning process, back propagation is used to update the weights and threshold values of the network to achieve the minimum error sum of square [13]. When a pair of the learning samples are input into the network, the neuron activation values are regulated from the output layer to the input layer, to obtain input response in the output layer neurons. With the spread of correcting such errors inversely ongoing, correct rate of the network is input to the model in response to the rise.

BP neural network is used here as a mathematic model on fermentation condition for *Phellinus igniarius*. The proposed BP neural network contains three layers of neurons:

(i) Input layer has 9 neurons to input the values of the 9 related factors of fermentation condition: glucose, maltose, mannitol, corn powder, yeast, cupric sulfate, ferrous sulfate, sodium chloride, and vitamin B_1.

(ii) Hidden layer with 11 neurons is used to generate the scaled estimated value of *Phellinus* yield. The hidden layer neurons should be selected as an integer between 3 and 13. Here the hidden layer neurons are decided to be 11, since the variance between the predicted value and the actual value is minimum when the number of the hidden layer neurons is 11.

(iii) Output layer with one neuron is used to calculate the production of *Phellinus igniarius*.

The input and output values are limited in the range $[-1, 1]$ by

$$y = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})}, \tag{1}$$

where $x_{\max}$ is the maximum value of the same class in the training set of data, $x_{\min}$ is the minimum value of the same class in the training set of data, x is the true value of the same class in the training set of data, and y is the input or output value of the network.

The topological structure of the proposed BP neural network model is shown in Figure 1.

The Levenberg-Marquardt algorithm and scaled conjugate gradient method are used for training the network. Such training strategy uses both the information of the first derivative of the objective function and the information with the second derivative of the objective function, which are described as follows:

$$X^{k+1} = X^k + \alpha^k S\left(X^k\right), \tag{2}$$

where X^k is a vector composed of all the weights and thresholds in the network, $S(X^k)$ is the search direction of the vector space composed of every component of X, and α^k is the smallest steps using $f(X^{k+1})$ on the directory of $S(X^k)$.

In order to establish a training set, more than 5000 experiments were completed as follows: inoculating the *Phellinus* strains on PDA slant medium and cultivating it for 7 days under the temperature of 28°C; loading 200 mL PDA liquid medium in 500 mL flask, keeping the temperature at 28°C, the speed of the centrifuge at 150 rpm, and the PH nature, and cultivating it for 7 days after inoculating; cultivating the 250 mL shake flasks for 7 days after inoculating, where the inoculation of seed liquid is 10%, the capacity of medium is 100 mL, the temperature is 28°C, and the speed is 150 rpm. Data of 25 experimental fermentation conditions with optimal production of *Phellinus igniarius* are selected from the above experiments as training set. The data of 25 experimental conditions is shown in Table 1, in which "Glu" stands for glucose, "Mal" for maltose, "Mann" for mannitol, "CP" for corn powder, "CS" for cupric sulfate, "SC" for sodium chloride, "FS" for ferrous sulfate, and "TF" for total flavonoids.

Three parameters are used to measure the accuracy and speed of the network. Specifically, obtained target error

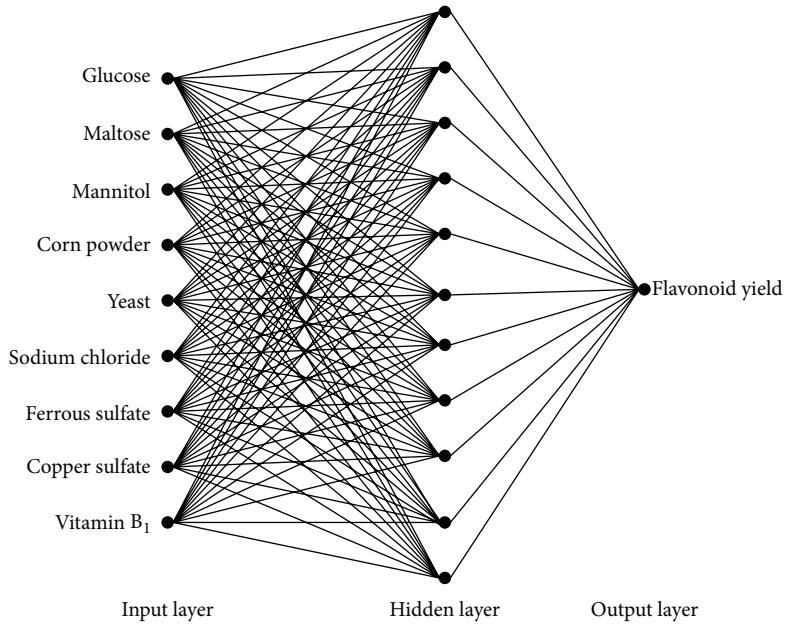


Figure 1: The BP neural network model.

Table 1: 25 optimal fermentation conditions of experiments for production of *Phellinus igniarius*.

Number	Glu	Mal	Mann	CP	Yeast	CS	SC	FS	VB_1	TF
1	25	25	20	12.5	12.5	5	0.5	12.5	1.5	$1.24E+03$
2	20	25	25	10	12.5	6.25	0.5	10	1.5	$0.88E+03$
3	25	20	25	12.5	10	6.25	0.625	12.5	1.5	$0.98E+03$
4	20	25	25	12.5	10	5	0.625	12.5	1.5	$0.95E+03$
5	25	25	20	10	10	5	0.625	10	1.875	$0.80E+03$
6	25	25	25	10	10	6.25	0.5	10	1.875	$0.76E+03$
7	20	20	20	10	10	5	0.5	10	1.5	$0.74E+03$
8	25	20	20	10	12.5	5	0.5	10	1.5	$0.74E+03$
9	20	20	20	12.5	10	6.25	0.5	12.5	1.875	$0.85E+03$
10	20	20	25	10	12.5	5	0.625	10	1.875	$0.85E+03$
11	20	25	20	12.5	12.5	6.25	0.625	12.5	1.875	$0.99E+03$
12	25	20	25	12.5	12.5	5	0.5	12.5	1.875	$1.03E+03$
13	20	20	20	10	10	5	0.5	12.5	1.5	$1.52E+03$
14	20	20	20	12.8	12	5	0.5	10	1.5	$0.99E+03$
15	20	20	20	10	10	5	0.5	12.5	1.5	$1.51E+03$
16	20	20	20	10	7.2	5	0.5	12.5	1.5	$1.36E+03$
17	20	20	20	10	12.8	5	0.5	10	1.5	$0.89E+03$
18	20	20	20	7.2	10	5	0.5	10	1.5	$1.32E+03$
19	20	20	20	12	12	5	0.5	10	1.5	$1.42E+03$
20	20	20	20	10	10	5	0.5	10	1.5	$1.56E+03$
21	20	20	20	12	8	5	0.5	12.5	1.5	$1.51E+03$
22	20	20	20	8	12	5	0.5	10	1.5	$1.08E+03$
23	20	20	20	10	10	5	0.5	12.5	1.5	$1.50E+03$
24	20	20	20	8	8	5	0.5	12.5	1.5	$1.61E+03$
25	20	20	20	10	10	5	0.5	25	1.5	$1.31E+03$

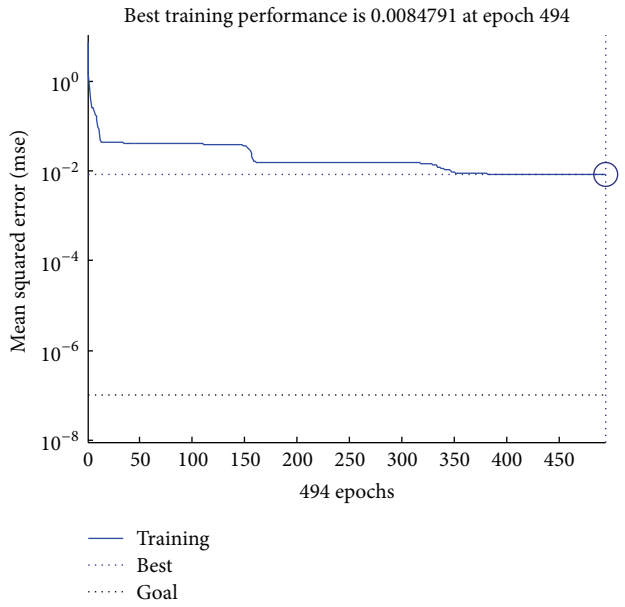


Figure 2: Convergence property of neural network.

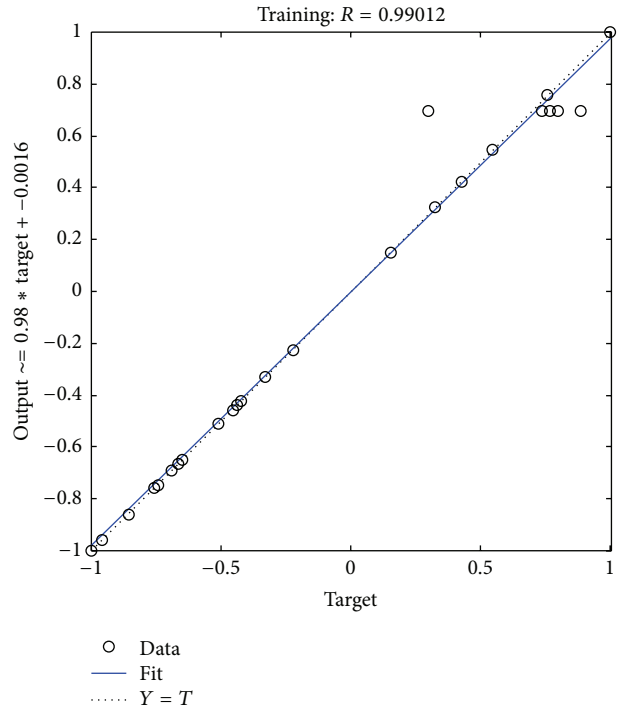


Figure 3: The coverage of model.

(MSE) is 0.018999, operation time is 3 seconds, and the number of iterations is 577 times. Convergence property of the neural network is shown in Figure 2.

As shown in Figure 2, it is easy to find that the convergence is rapid, and the neural network converges slowly when it is closed to the target solution and finally converges to a best value. The coverage of model is shown in Figure 3.

2.2. Genetic Algorithm for Optimization of Fermentation Condition. In this subsection, genetic algorithm (GA) is used to optimize the fermentation condition.

GA is known as an intelligent algorithm inspired by the natural selection and genetic mechanism [14]. Similar to the basic laws of nature evolution, "survival of the fittest" is the core mechanism of the genetic algorithm; meanwhile, "reproduce," "crossover," and "mutation" operators are used in GA. The process has the following steps [15].

Step 1. Encode the individuals; since there are 9 variables in consideration, an individual should contain 9 bases. For example, encode [glucose, maltose, mannitol, corn powder, yeast, copper sulfate, sodium chloride, ferrous sulfate, vitamin B_1] to $[g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9]$, where $g_1 \in [0, 40]$, $g_2 \in [0, 40]$, $g_3 \in [0, 40]$, $g_4 \in [0, 100]$, $g_5 \in [0, 100]$, $g_6 \in [0, 0.5]$, $g_7 \in [0, 10]$, $g_8 \in [0, 0.5]$, $g_9 \in [0, 0.1]$, and the unit in use is g/L.

Step 2. Select "good" individuals according to a fitness function.

Step 3. Remove individuals with low fitness.

Step 4. Perform crossover and mutation to generate new individuals.

Step 5. Generate a new generation for evaluation by fitness function, and go to Step 2.

The process can be repeated until the halting condition is matched.

In iterations analysis, the numbers of iterations are set to be 100, 150, 200, and 500, respectively. The range of crossover probability is from 0.6 to 1, and mutation probability is from 0.01 to 0.1. The size of the population is set to be 300 and chromosome size is 9 for factors glucose, maltose, mannitol, corn pulp powder, yeast, sodium chloride, ferrous sulfate, copper sulfate, and vitamin B_1. The encoding strategy of chromosome is floating-point (real) coding, where crossover and mutation are directly on the real operation. The concentration ranges of the factors are given as follows: glucose: 0–40 g/L, maltose: 0–40 g/L, mannitol: 0–40 g/L, corn pulp powder: 0–100 g/L, yeast: 0–100 g/L, copper sulfate: 0–0.5 g/L, sodium chloride: 0–10 g/L, ferrous sulfate: 0–0.5 g/L, and vitamin B_1: 0–0.1 g/L.

In our method, the BP neural network obtained in Section 2.1 is used as fitness function to select good individuals. Selection operator is roulette selection method, roulette wheel selection, also known as proportional selection operator. The basic idea is that the probability of each individual selected is proportional to its fitness value. Assuming that group size is N, x_i is an individual, the fitness of x_i is $f(x_i)$, and the selection probability of x_i is

$$P(x_i) = \frac{f(x_i)}{\sum_{j-1}^{N} f(x_j)}. \quad (3)$$

Table 2: 10 optimized fermentation conditions for *Phellinus igniarius*.

Number	Glu	Mal	Mann	CP	Yeast	CS	SC	FS	VB_1	TF
1	15.04	15.2	29.97	5.27	5.09	0.18	0.22	9.8	1	$2.22E+03$
2	15.13	15.27	29.85	5	5.16	0.2	0.23	9.82	1.08	$2.22E+03$
3	15.04	15.63	29.81	5.16	5.11	0.19	0.55	9.96	1.01	$2.22E+03$
4	15.03	15.64	29.98	6.35	5.39	0.16	0.4	9.81	1.11	$2.20E+03$
5	15.09	15.1	29.82	5.1	5.24	0.2	0.23	9.99	1.08	$2.23E+03$
6	15.13	15.04	29.6	5.1	5.15	0.19	0.22	9.74	1	$2.21E+03$
7	15.01	15.13	29.81	5.14	5.06	0.16	0.22	9.88	1.01	$2.23E+03$
8	15.01	15.2	29.46	5.06	5.09	0.2	0.27	9.94	1.1	$2.22E+03$
9	15.02	15.39	29.99	5.02	5.1	0.2	0.29	9.99	1	$2.23E+03$
10	15.52	15.04	29.98	5.21	5.09	0.18	0.32	9.98	1	$2.22E+03$

3. Results

Since GA starts with certain randomly selected individuals, we perform 100 data experiments. In Table 2, 10 optimized fermentation conditions for *Phellinus igniarius* are shown, where

(i) the average of glucose is 15.1 g/mL;

(ii) the average of maltose is 15.264 g/mL;

(iii) the average of mannitol is 29.83 g/mL;

(iv) the average of corn pulp powder is 5.23 g/mL;

(v) the average of yeast is 5.14 g/mL;

(vi) the average of copper sulfate is 0.18 g/mL;

(vii) the average of sodium chloride is 9.89 g/mL;

(viii) the average of ferrous sulfate is 0.3 g/mL;

(ix) the average of vitamin B_1 is 1.03 g/mL.

The average production of *Phellinus* is 2200 μg/mL.

Our method has the intelligent learning ability (by BP neural network) and can overcome the limitation of large-scale biotic experiments. Through simulations, the optimal culture conditions are obtained and the flavones production is increased to 2200 μg/mL from the known optimal result of 1532.83 μg/mL in [6].

4. Conclusion

In this work, we focused on the optimization of fermentation conditions for *Phellinus igniarius*, including the concentration of glucose, maltose, mannitol, corn powder, yeast extract, copper sulfate, sodium chloride, ferrous sulfate, and vitamin B_1. A hybrid algorithm of GA is proposed, where a BP neural network trained by 25 groups of data of experiments with optimal productions of *Phellinus igniarius* is used as the fitness function of GA. The simulation results show that our method has the ability to overcome the limitation of large-scale biotic experiments. The optimal culture conditions are obtained and the flavones production is increased to 2200 μg/mL. Our work would also be a guide for the "Precision Medicine" with personal SNP data [16] and other tasks in bioinformatics [17, 18].

In our study, BP neural network is used, which is known as some classical neural computing models. It is of interest to use spiking neural networks computing models to do the optimization [19–22]. In the framework of membrane computing, cell-like [23] and tissue-like [24] computing models have been proved to be powerful as bioinspired computing models. What will happen if these models are used in calculating the optimized conditions? Some web servers are useful for biological data processing; see, for example, [25]. It is worthy to develop some web servers for experimental conditions optimization.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The research is under the auspices of National Natural Science Foundation of China (nos. 41276135, 31172010, 61272093, 61320106005, 61402187, 61502535, 61572522, and 61572523), Program for New Century Excellent Talents in University (NCET-13-1031), 863 Program (2015AA020925), Fundamental Research Funds for the Central Universities (R1607005A), and China Postdoctoral Science Foundation funded project (2016M592267).

References

[1] Y. Yang, J. Hu, Y. Liu et al., "Antioxidant and cytotoxic activities of ethanolic extracts and isolated fractions of species of the genus *Phellinus* Quél. (Aphyllophoromycetideae)," *International Journal of Medicinal Mushrooms*, vol. 13, no. 2, pp. 145–152, 2011.

[2] N.-K. Zeng, Q.-Y. Wang, and M.-S. Su, "The breeding of *Phellinus baumii* by protoplast fusion," *Journal of Chinese Medicinal Materials*, vol. 31, no. 4, pp. 475–478, 2008.

[3] Q. Shao, Y. Yang, T. Li et al., "Biological activities of ethanol extracts of *Phellinus baumii* (higher basidiomycetes) obtained by different fermentation methods," *International Journal of Medicinal Mushrooms*, vol. 17, no. 4, pp. 361–369, 2015.

[4] Z. Zhu, N. Li, J. Wang, and X. Tang, "Establishment and analysis of the fermentation model of *Phellinus igniarius*," *AASRI Procedia*, vol. 1, pp. 2–7, 2012.

[5] B. Gu, X. Sun, and V. S. Sheng, "Structural minimax probability machine," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[6] M. Zhu, Y. Wang, G. Zhang et al., "Liquid fermentation conditions for production of mycelium of *Phellinus linteus*," *China Brewing*, vol. 29, no. 11, pp. 88–91, 2010.

[7] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403–1416, 2015.

[8] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for v-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[9] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.

[10] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[12] L.-Y. Zhang and B.-R. Tao, "Ontology mapping based on Bayesian network," *Journal of Donghua University (English Edition)*, vol. 32, no. 4, pp. 681–687, 2015.

[13] C. Kwak, J. A. Ventura, and K. Tofang-Sazi, "Neural network approach for defect identification and classification on leather fabric," *Journal of Intelligent Manufacturing*, vol. 11, no. 5, pp. 485–499, 2000.

[14] P. K. H. Phua and D. Ming, "Parallel nonlinear optimization techniques for training neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1460–1468, 2003.

[15] G. Mitsuo and C. Runwei, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, 2000.

[16] P. Li, M. Guo, C. Wang, X. Liu, and Q. Zou, "An overview of SNP interactions in genome-wide association studies," *Briefings in Functional Genomics*, vol. 14, no. 2, pp. 143–155, 2015.

[17] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. 1, pp. W65–W71, 2015.

[18] R. Wang, Y. Xu, and B. Liu, "Recombination spot identification based on gapped k-mers," *Scientific Reports*, vol. 6, Article ID 23934, 2016.

[19] X. Zhang, B. Wang, and L. Pan, "Spiking neural P systems with a generalized use of rules," *Neural Computation*, vol. 26, no. 12, pp. 2925–2943, 2014.

[20] T. Song and L. Pan, "On the universality and non-universality of spiking neural P systems with rules on synapses," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 960–966, 2015.

[21] T. Song, Q. Zou, X. Liu, and X. Zeng, "Asynchronous spiking neural P systems with rules on synapses," *Neurocomputing*, vol. 151, part 3, pp. 1439–1445, 2015.

[22] X. Wang, T. Song, F. Gong, and P. Zheng, "On the computational power of spiking neural P systems with self-organization," *Scientific Reports*, vol. 6, Article ID 27624, 2016.

[23] X. Zhang, L. Pan, and A. Păun, "On the universality of axon P systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2816–2829, 2015.

[24] X. Zhang, Y. Liu, B. Luo, and L. Pan, "Computational power of tissue P systems for generating control languages," *Information Sciences*, vol. 278, pp. 285–297, 2014.

[25] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K.-C. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, pp. W65–W71, 2015.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 2375268, 12 pages
http://dx.doi.org/10.1155/2016/2375268

Research Article

In Silico Prediction of Gamma-Aminobutyric Acid Type-A Receptors Using Novel Machine-Learning-Based SVM and GBDT Approaches

Zhijun Liao,[1] Yong Huang,[2] Xiaodong Yue,[3] Huijuan Lu,[4] Ping Xuan,[5] and Ying Ju[6]

[1]*Department of Biochemistry and Molecular Biology, School of Basic Medical Sciences, Fujian Medical University, Fuzhou, Fujian 350122, China*
[2]*College of Animal Science and Technology, Henan University of Science and Technology, Luoyang, Henan 471023, China*
[3]*School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*
[4]*College of Information Engineering, China Jiliang University, Hangzhou, Zhejiang 310018, China*
[5]*School of Computer Science and Technology, Heilongjiang University, Harbin, Heilongjiang 150080, China*
[6]*School of Information Science and Technology, Xiamen University, Xiamen, Fujian 361005, China*

Correspondence should be addressed to Ping Xuan; 2004058@hlju.edu.cn

Received 24 April 2016; Revised 8 June 2016; Accepted 19 June 2016

Academic Editor: Yungang Xu

Gamma-aminobutyric acid type-A receptors ($GABA_A$Rs) belong to multisubunit membrane spanning ligand-gated ion channels (LGICs) which act as the principal mediators of rapid inhibitory synaptic transmission in the human brain. Therefore, the category prediction of $GABA_A$Rs just from the protein amino acid sequence would be very helpful for the recognition and research of novel receptors. Based on the proteins' physicochemical properties, amino acids composition and position, a $GABA_A$R classifier was first constructed using a 188-dimensional (188D) algorithm at 90% cd-hit identity and compared with pseudo-amino acid composition (PseAAC) and ProtrWeb web-based algorithms for human $GABA_A$R proteins. Then, four classifiers including gradient boosting decision tree (GBDT), random forest (RF), a library for support vector machine (libSVM), and k-nearest neighbor (*k*-NN) were compared on the dataset at cd-hit 40% low identity. This work obtained the highest correctly classified rate at 96.8% and the highest specificity at 99.29%. But the values of sensitivity, accuracy, and Matthew's correlation coefficient were a little lower than those of PseAAC and ProtrWeb; GBDT and libSVM can make a little better performance than RF and *k*-NN at the second dataset. In conclusion, a $GABA_A$R classifier was successfully constructed using only the protein sequence information.

1. Introduction

Gamma-aminobutyric acid (GABA) is a major human brain inhibitory neurotransmitter and plays a principal role in the regulation of pituitary gland function. GABA is made up of a four-carbon chain flexible carbon skeleton (Figure 1), which can adopt a number of conformations when interacting with many macromolecular receptor targets. This characteristic of GABA can provide many selective ligands by producing conformationally restricted analogues [1]. GABA is mainly synthesized in the hypothalamus as well as within the pituitary gland and stored in the anterior lobe and intermediate lobe cells, the GABA-synthesizing enzyme is glutamic acid decarboxylase (GAD) which is relevant to TCA cycle [2], and the direct substrate is glutamate [3] (Figure 2). In addition to GAD, the GABA level is also related to glutamine-glutamate (Gln-Glu) cycling [4], in which glutaminase and glutamine synthetase play a key role in keeping the cycling balance. Gln is first converted to Glu and then to GABA in the cycle, or Glu solution is catalyzed to GABA; this process is known to play a significant role in the regulation of neurogenesis, and the release of GABA is mainly produced from Purkinje cells in the cerebellar cortex via special regulatory mechanism [5–7].

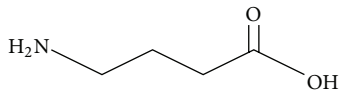


Figure 1: GABA conformation.

GABA can specifically interact with the postsynaptic GABA receptor in human central nervous system (CNS) [8]; the specific binding of GABA to synaptic membrane fractions is saturable. Three types of GABA receptors are expressed in human, namely, the ionotropic $GABA_A$ receptor ($GABA_AR$), the metabotropic $GABA_B$ receptor (such as G protein-coupled receptor) [9], and another ionotropic $GABA_C$ receptor, among them $GABA_AR$ is relevant to epilepsy [10]. These receptors belong to the Cys-loop superfamily of ligand-gated ion channels (LGICs) and exhibit a long (about 200 a.a.) extracellular amino terminus, which is thought to be responsible for ligand channel interactions. The amino terminus forms agonist or antagonist binding sites, four transmembrane (TM) domains, and a large intracellular domain between TM3 and TM4 for phosphorylating regulation and localization at synapses, and five TM2 domains in a cycle form the lining segment of the ion channel (Figure 3). The extracellular amino terminus contains a conserved motif, called the Cys-loop (13-amino acid disulfide loop), which is characterized by 2 cysteine residues spaced by 13 different amino acid residues [11]; the amino terminus incorporates neurotransmitters and some modulator binding sites. For example, the extracellular domain of $GABA_AR$ $\beta 2$ subunits contains the amino acid residue "CMMDLRRYPLDEQNC" (C stands for cysteine). For the structural details of Cys-loop receptors see review [12].

$GABA_A$Rs form pentameric chloride channels comprising various combinations from eight kinds of subunits (α, β, γ, δ, ε, θ, π, and ρ), each of which comprises several subtypes [13]. These receptors belong to a superfamily of pentameric ligand-gated ion channels (pLGICs) with five-membered ring structures; pLGICs are also known as Cys-loop receptors including two classes: the cation-selective (e.g., nicotinic acetylcholine receptors and serotonin type 3 receptors) and anion-selective (e.g., glycine receptors (GlyRs) and $GABA_A$Rs) [14]. According to their extracellular domain, pentameric receptors can be further divided into these containing only one conserved Cys-loop and those containing an additional disulfide bond that links the $\beta 9$-$\beta 10$ strands in Loop C. Human $GABA_AR$ subunits are encoded by 19 different genes, namely, $\alpha 1$–6, $\beta 1$–3, $\gamma 1$–3, δ, ε, θ, π, and $\rho 1$–3; among these subunits, the crystallization shows that human $GABA_AR$ $\beta 3$ subunit is unique to eukaryotic Cys-loop receptors [15]. The $\alpha 1$–$\alpha 6$ subunits are encoded by *GABRA1* to *GABRA6* genes; the $\alpha 1$ subtype is widely expressed in the whole brain, whereas $\alpha 2$, $\alpha 3$, $\alpha 4$, $\alpha 5$, and $\alpha 6$ subtypes are expressed in specific brain areas [16]. Most of the pentameric $GABA_A$Rs in the human brain are typically composed of two α subunits, two β subunits, and one γ subunit, and the GABA binding sites are located in the α-β subunit interface [17]. The $\alpha 1$, $\beta 2$, and $\gamma 2$ subunits are expressed most abundantly in human brain [18], and the subunit variants may thus influence ion channel gating, expression, and GABA receptor trafficking to the cell surface. The *GABRA1* and *GABRA6* genes are located in human chromosome 5, whereas *GABRA2* and *GABRA3* are located in chromosome 4 and *GABRA4* and *GABRA5* are located in chromosome X and chromosome 15, respectively [19]. These genes have been proposed to affect certain drug targets and the regulation of neuronal activities in human brain [20]. Several antiepileptic drugs (AEDs) such as phenobarbital and gabapentin bind to $GABA_A$Rs in the CNS with a confined area distribution, and the alterations in $GABA_AR$ subunits may regulate the responses elicited by AEDs [21]. Several AEDs exert agonistic effects on $GABA_A$Rs. AEDs may react with $GABA_A$Rs comprising distinct subunits in diverse manners, and the composition and function of α subunits may influence the treatment efficacy of different AEDs [22]. Targeted proteins of AEDs are involved in the regulation of extracellular K^+ and intracellular Cl^- homeostasis, cell volume, and pH, all of which are important for maintaining normal brain activity [23].

$GABA_AR$ subunit mutations or genetic variations can lead to its dysfunctions, which have been thought to participate in the pathomechanisms of epilepsy [24], in which multiple $GABA_AR$ epilepsy mutations result in protein misfolding and may cause degradation or retention of the protein molecules in cells; Kang et al. found that mutant $GABA_AR$ $\gamma 2$ subunits accumulate and aggregate intracellularly, activated caspase-3, and caused widespread and age-dependent neurodegeneration; these findings suggested the epilepsy-associated mutant $\gamma 2$ subunit played important role in neurodegeneration [25]. The gene mutations or genetic variation of the $\alpha 1$, $\alpha 6$, $\beta 2$, $\beta 3$, $\gamma 2$, or δ subunits (GABRA1, GABRA6, GABRB2, GABRB3, GABRG2, and GABRD, resp.) compromises hyperpolarization through $GABA_A$Rs, and these variations have been associated with human epilepsy with or without febrile seizures [26].

Support vector machine (SVM) is a kind of supervised machine learning algorithms that have been broadly applied for classification and regression analysis [27–32], which is also a type of sparse kernel machines that rely on various data to predict unknown class labels and which has linear or nonlinear learning model for binary classifier [33–35]. Random forest (RF) is an ensemble machine learning technique based on random decision trees for classification and other tasks. Relying on the feature, a data point can be divided into a special category and is assigned a prediction. RF has been broadly applied in novel protein and target identification [36, 37], because it combines the merits of bagging idea and feature selection [38]. Another decision tree learning is gradient boosting decision tree (GBDT), which has been very successfully applied for many fields such as smart city concept [39], and its major advantage is ability to find nonlinear interactions automatically through decision tree learning with the minimality error. GBDT is generally regarded as one of the best out-of-the-box classifiers which has the ability to generalize and can combine weak learners into a single strong learner; it has gradually acquired popularity in the field of machine learning methods although it still possesses many disadvantages [40–43].

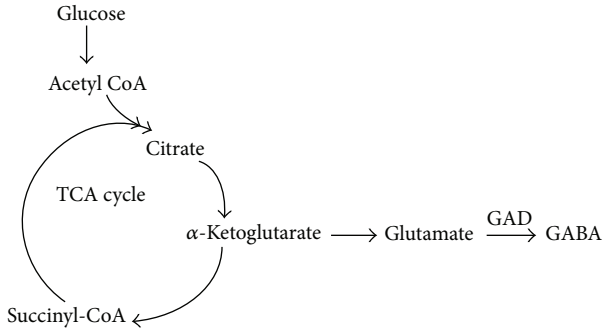


Figure 2: Model of direct GABA production.

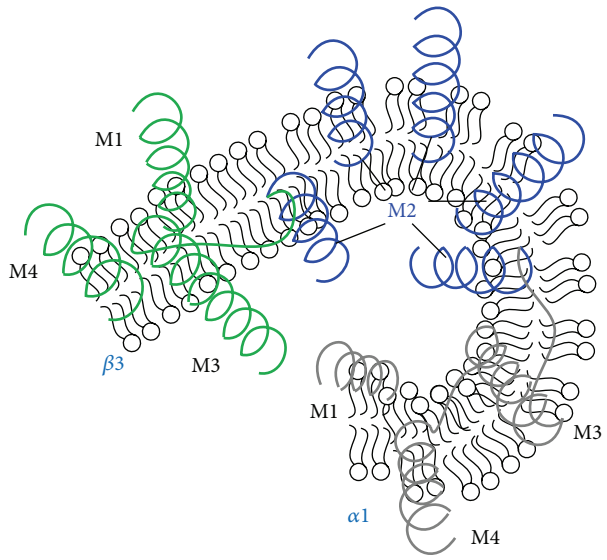


Figure 3: $GABA_AR$ modulation patterns of transmembrane domain, a homology model of the transmembrane domains of a $GABA_AR$ showing the five-M2-helix domains forming the chloride ion channel (blue) and M1, M3, and M4 helices for single $\alpha 1$ (grey) or $\beta 3$ (green) subunit. The helices may embed into the postsynaptic membrane in mammalian CNS.

Here, we performed an *in silico* analysis on the $GABA_AR$s according to sequence information and other physicochemical features, including hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure, and solvent accessibility. Twenty natural amino acids can be divided into 3 different groups based on each of the above eight properties, and thus 188-dimensional (188D) feature vectors of proteins were constructed with an ensemble classifier [44], which performed well in membrane protein prediction [45]. We employed PseAAC and ProtrWeb methods for human $GABA_AR$ to adapt to the web server limit of sequence amounts; we also applied libSVM, RF, GBDT, and widely used k-nearest neighbor (k-NN) algorithms to make comparisons of performance with dataset at rigorous cd-hit filtration [46].

Since *motif*, a conserved short pattern of a protein [47], is one of the fundamental function units of molecular evolution, with regard to DNA, a motif may act as a protein-binding site; in proteins, a motif may directly correspond to the active site of an enzyme or a structural unit of the protein. Therefore, we also conducted motif analysis.

2. Materials and Methods

2.1. Data Retrieval and Treatment. All the primary sequences of both $GABA_AR$ and the control Pfam proteins (in FASTA files) were retrieved from the UniProt database (http://www.uniprot.org/); the raw data are preprocessed by cd-hit program (http://cd-hit.org) to merge the sequence similarities and reduce the complexity [46]. To avoid bias in the classifier, we set the identity at 90% similarity and obtained the results of 2353 $GABA_AR$ sequences as positive dataset; the negative samples were obtained from the control proteins when the positive ones were deleted, and 10652 entries were obtained as negative dataset. When the four classifiers performance was measured, cd-hit was set at rigorous 40% identity and gained 360 $GABA_AR$s and 9598 non-$GABA_AR$s.

2.2. Prediction Analysis for Potential $GABA_AR$ Proteins. Machine learning is often employed in the bioinformatics and proteomics problem. Several important techniques facilitate

the protein classification and identification, such as imbalanced classification strategies [48], ensemble learning [49–51], samples selection strategies [52, 53], features reduction, and ranking methods [54–56].

To predict the potential $GABA_AR$ from the amino acid sequences, we constructed a classifier according to the $GABA_AR$ protein features. First, we extracted the feature vectors from positive versus negative protein sequence dataset by using a novel machine-learning-based method developed by our group, we transformed all the positive and negative sequences into the corresponding protein family (Pfam) information, and the obtained features included sequence evolutional information, *k*-skip-*n*-gram model, physicochemical properties, and local PsePSSM [57]. Altogether, we assembled 188D feature vectors. Afterward, the resulting feature vectors were imported into Weka (http://www.cs.waikato.ac.nz/ml/weka/), which is a machine learning workbench used for automatic classification via visualization and cross-validation analysis [58, 59]. After several preliminary trials with the same dataset, we selected random forest method and set the parameters as default.

2.3. Conserved Motif Analysis of Human $GABA_AR$ Proteins. Conserved motif analyses were implemented using the online MEME Suite (http://meme-suite.org/, 4.11.1 version), a powerful motif-based sequence analysis tool, which integrated a set of web-based tools including Gene Ontology database for studying sequence motifs in proteins, DNA, and RNA [60]. Currently, the MEME Suite has added six new tools and reached thirteen since the "Nucleic Acids Research" Web Server Issue in 2009. Human $GABA_AR$ sequences in FASTA format were used as a file input. The maximum motif width, minimal motif width, and maximum number of motifs were set to 50, 6, and 9, respectively. The remaining parameters were set as default values.

2.4. Pseudo-Amino Acid Composition and ProtrWeb Analysis. Chou et al. [61–63] had proposed the concept of PseAAC to describe global or long-range sequence-order protein information early in 2001; their original design objective was to improve protein subcellular localization prediction and membrane protein type prediction. Since then, the PseAAC approach alone or incorporating other properties had rapidly penetrated many areas of computational proteomics. As the most intuitive features for protein biochemical reactions, the physicochemical properties of amino acids significantly influence the protein classification. Features that incorporate appropriate physicochemical properties can contain much valuable information for improving the performance of predictors. Single feature extraction of our own method has inevitably its own shortcomings and does not always perform well on all circumstances. Thus, we also used the concept of PseAAC and ProtrWeb (http://protrweb.scbdd.com/) to construct feature vectors for human $GABA_AR$ proteins (58 entries) and other proteins (58 entries) in this study.

PseAAC is a web server that can generate numerous pseudo-amino acid compositions including sequence-order information in addition to the conventional 20D amino acid composition. It is a classification algorithm based on the amino acid composition and physicochemical characteristics of proteins; the server was designed in a flexible way to identify various pseudo-amino acid composition information for a given protein sequence by selecting different parameters and their combinations. PseAAC provides three PseAA modes and six amino acid characters for user to choose. ProtrWeb [64] is also a web server based on the R package routine protr, the first version of which was developed in November 2013. This server is dedicated to calculate protein sequence-derived structural and physicochemical descriptors such as amino acid composition. *n*-gram and *k*-skip are based on permutation and combination theory. ProtrWeb can be applied in various protein prediction studies, including protein structural and functional classes, protein subcellular locations, protein-protein interactions, and receptor-ligand interactions. ProtrWeb offers 12 types of commonly used descriptors presented in the web such as amino acid composition, dipeptide composition, and pseudo-amino acid composition. Recently, some studies have shown that the long-range sequence-order effects of DNA [65] can improve the performance of computational predictors [66].

To extract features from the physicochemical properties of proteins by using PseAAC, we considered all six physiochemical properties: hydrophobicity, hydrophilicity, mass, pK1 (alpha-COOH), pK2 (NH3), and pI (at 25°C). We selected type 2 PseAA mode, set Lambda parameter at 10, and set the weight factor as default. The results were shown as 80-dimensional (80D) data for each protein. For ProtrWeb, we chose amino acid composition (20 Dim) and pseudo-amino acid composition (50 Dim) adapted to the restricted parameter measure.

2.5. Prediction Ability Comparison of Four Classifiers on the 40% Identity cd-Hit Filtration Data. We extracted 188D feature vectors from 360 $GABA_AR$s and 9598 non-$GABA_AR$s as input to Weka performing category via RF, *k*-NN, and SVM algorithm which was implemented using libSVM. GBDT classifier was carried out by python program developed by ourselves; the above 4 classifiers have the parameters set as default.

Four common measurements were used to illuminate the performance quality of the predictor more intuitively. Sensitivity (Sn), specificity (Sp), accuracy (Acc), and Matthew's correlation coefficient (MCC) were adopted to evaluate the above three methods and four classifiers. These methods are formulated as follows:

$$\begin{aligned} \mathrm{Sn} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{Sp} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}, \\ \mathrm{Acc} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}, \\ \mathrm{MCC} &= \frac{\mathrm{TP}*\mathrm{TN}-\mathrm{FP}*\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FN})\,(\mathrm{TP}+\mathrm{FP})\,(\mathrm{TN}+\mathrm{FP})\,(\mathrm{TN}+\mathrm{FN})}}, \end{aligned} \tag{1}$$

Table 1: Pfam accession numbers (83 entries) for $GABA_AR$s as positive group.

PF00008, PF00012, PF00018, PF00022, PF00028, PF00053, PF00055, PF00057, PF00059, PF00060, PF00069
PF00078, PF00084, PF00087, PF00090, PF00100, PF00130, PF00147, PF00163, PF00168, PF00169, PF00209
PF00226, PF00240, PF00270, PF00271, PF00335, PF00387, PF00388, PF00397, PF00400, PF00454, PF00520
PF00564, PF00621, PF00627, PF00643, PF00651, PF00665, PF00754, PF00850, PF00892, PF01082, PF01352
PF01436, PF01479, PF01498, PF01529, PF02072, PF02140, PF02214, PF02259, PF02260, PF02460, PF02891
PF02931, PF02932, PF02991, PF03144, PF03416, PF03521, PF04849, PF06220, PF07645, PF07690, PF07707
PF08007, PF08266, PF08377, PF08625, PF08771, PF09279, PF09497, PF11865, PF11938, PF12248, PF12448
PF12662, PF13499, PF15311, PF15974, PF16457, PF16492

where TP, TN, FP, and FN stand for the numbers of true positive, true negative, false positive, and false negative, respectively.

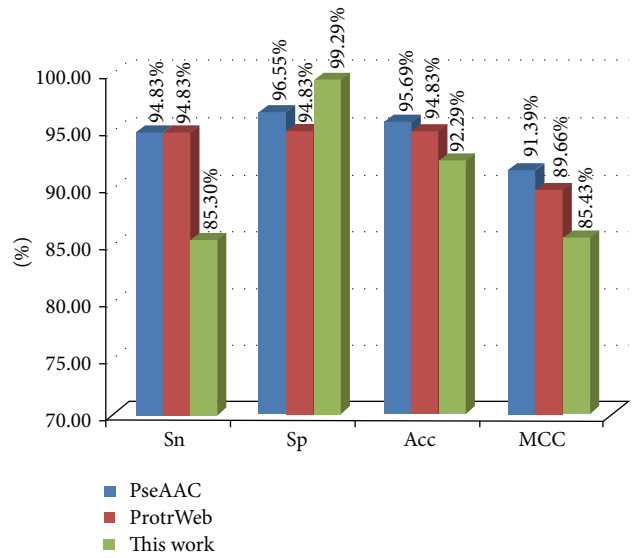


Figure 4: Sn, Sp, Acc, and MCC values listed from PseAAC, ProtrWeb, and our work. Note: PseAAC and ProtrWeb only include human 58 $GABA_AR$s and 58 non-$GABA_AR$s because of the web amount limitation; our method contains all the $GABA_AR$s and non-$GABA_AR$s (2353 versus 10652).

3. Results

3.1. Searching the Protein Family Number. To determine the Pfam families of $GABA_AR$s, we ran the program with the positive and negative protein sequences ($GABA_AR$s versus non-$GABA_AR$s) and obtained nonredundant Pfam numbers after combining the same ones (Table 1). The negative group was very large; thus, we only listed the positive ones.

3.2. Reclassification of Positive and Negative Proteins. We obtained the 188D (this work), 80D (from PseAAC), and 70D (from ProtrWeb) feature vector dataset from both positive and negative groups and used them as input to the Weka explorer (RF algorithm). The results showed that the correctly classified rates were 96.8%, 95.7%, and 94.8%. The confusion matrix is shown in Table 2, and the four common measurement values are illustrated in Figure 4.

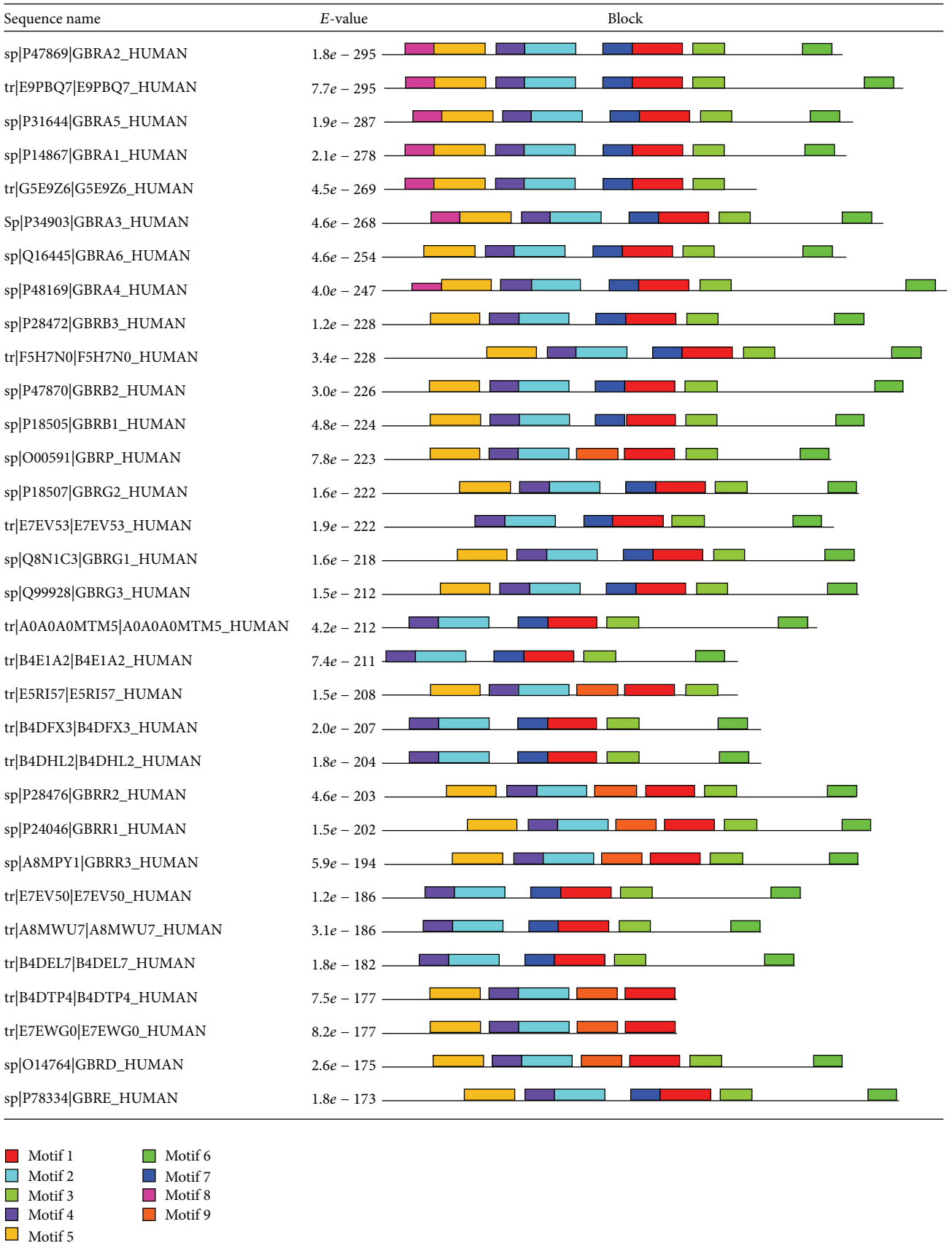
3.3. Four Classifiers' Prediction Ability Comparison. On the four classifiers, they all performed well and got high correctly classified rate over 96%, but GBDT and libSVM had a little better performance than RF and k-NN assessed from all the indicators (Table 3).

3.4. Conserved Motif Analysis of Human $GABA_AR$. To reveal the evolutionary correlation of $GABA_AR$s from the conserved motifs, 92 human protein sequences were analyzed by using MEME software. The nine most significant and conserved motifs are shown in Figure 5 and Table 4.

4. Discussion

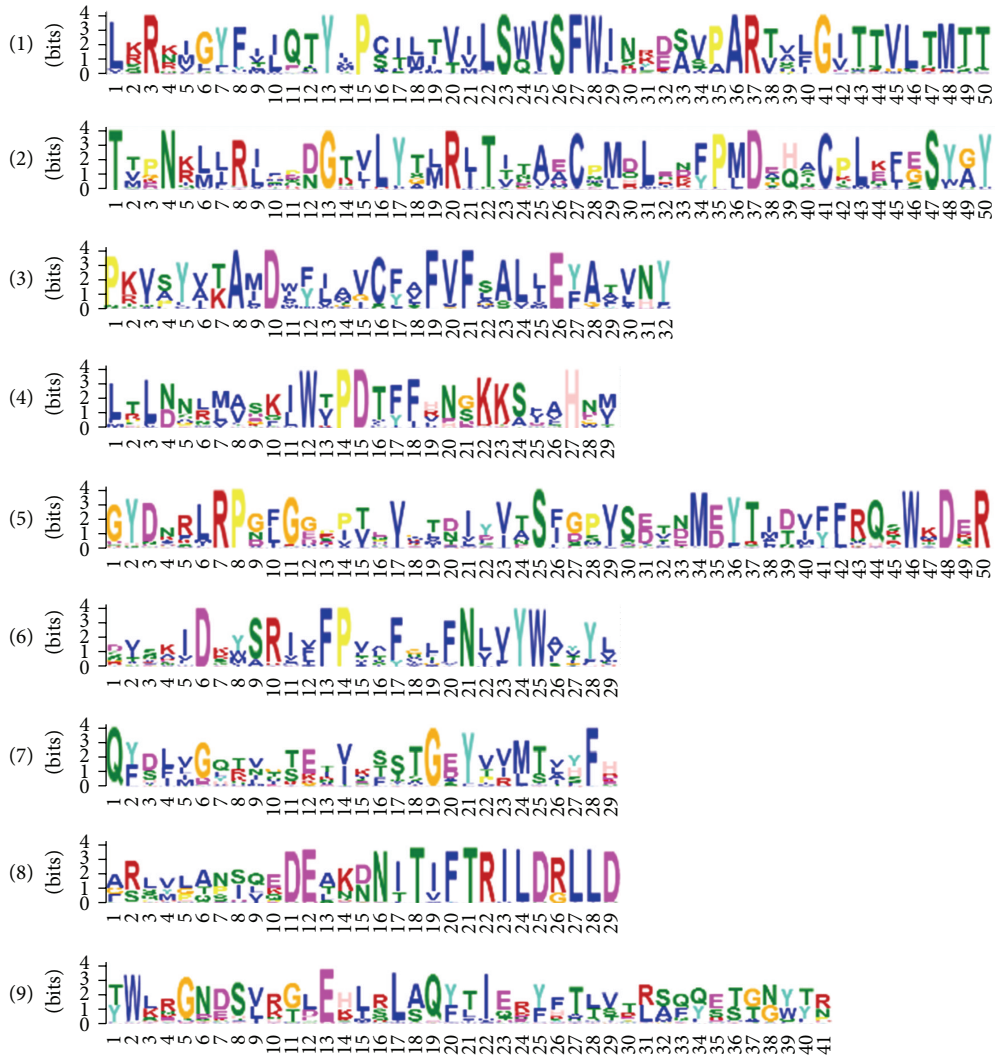
The primary structures of amino acid sequences are often the basis for understanding the three-dimensional conformation and functional properties of proteins [67], which exhibit an intimate relationship between their primary structure and function [68]. Twenty natural α-amino acids commonly constitute the primary sequences of proteins [69, 70]. Amino acids are biologically important organic nitrogenous compounds in the natural world. These compounds contain amine (-NH_2) and carboxylic acid (-COOH) functional groups which link with the same carbon atom called α-carbon, usually along with a side-chain (called R group) specific to each amino acid. The elements of carbon, hydrogen, oxygen, and nitrogen are essential for an amino acid, though other elements are found in the R group. Amino acids can be classified in many ways, such as according to the core structure and side-chain group properties. However, 20 standard and encoding α-carbon amino acids are usually classified into five main groups on the basis of biochemistry [71], namely, a hydrophobe, if the side-chain is nonpolar; a hydrophile, if it is polar but uncharged; aromatic, if it includes an aromatic ring; acidic, if it is negatively charged; and basic, if it is positively charged.

Previous research has extracted information on protein feature according to composition, position, or physicochemical properties [31]. In our work, we adopted 188D algorithm to extract feature vectors by combining amino acid compositions with physicochemical properties in a protein functional classifier [72]. This 188D method includes amino acid composition (20D) and eight types of physicochemical properties, that is, hydrophobicity (21D), normalized van der Waals volume (21D), polarity (21D), polarizability (21D), charge (21D), surface tension (21D), secondary structure (21D), and solvent accessibility (21D). The CTD model was employed to describe global information about the protein sequence, where C represents the percentage of each type of hydrophobic amino acid in an amino acid sequence, T



(a)

Figure 5: Continued.



(b)

Figure 5: Motifs of human $GABA_A$Rs found by the MEME system (for details see Table 3). (a) Locations of the nine discovered motifs (showing the top 32 sequences). (b) Nine motif logos found by MEME.

Table 2: Confusion matrix classifier (RF) from three kinds of feature vector extraction algorithms.

	PseAAC		ProtrWeb		This work	
	Human $GABA_A$Rs	Human non-$GABA_A$Rs	Human $GABA_A$Rs	Human non-$GABA_A$Rs	$GABA_A$R proteins	Non-$GABA_A$R proteins
Positive cases	55	2	55	3	2007	76
Negative cases	3	56	3	55	346	10576

Table 3: Classification results for four classifiers based on 360 $GABA_A$Rs and 9598 non-$GABA_A$Rs.

Classifier	Sensitivity (%)	Specificity (%)	Accuracy (%)	MCC	Correctly classified rate
GDBT	51.39	99.66	75.52	0.5828	0.9791
RF	41.39	99.86	70.63	0.5085	0.9775
libSVM	58.89	97.76	78.32	0.6148	0.9635
k-NN	51.94	98.17	75.06	0.5651	0.9650

Table 4: Human conserved motifs of $GABA_A$Rs found by MEME system (in regular expression).

Motif	Width	*E*-value	Best possible match
1	50	1.6*e* − 1592	L[KRS]R[KNR][IMV]GYF[IV][IL]QTY[IL]P[CS][IT][LM][TI][VT][IV]LS [WQ]VSFW[IL]N[RK][DE][SA][VS][PA]AR[TV][VAS][LF]G[IV] TTVLTMTT
2	50	8.6*e* − 1477	T[TV]PN[KR][LM][LI]R[IL]F[PD][DN]GT[VLI]LYT[LM]R[LI]T[ITV]TA [EA]C[PN][ML][DQ]L[ES][DNR][FY]P[ML]D[EAT][HQ][AST]CPL[KE] [FL][EG]SY[GA]Y
3	32	1.9*e* − 816	P[KR][VI][SA]Y[VAI][TK]A[MI]DW[FY][IL]AVC[FY][AV]FVF[SL]AL [LI]E[YF]A[TA][VL]NY
4	29	2.4*e* − 804	L[TR]L[ND]N[LR][ML][AV]SK[IL]W[TV]PDT[FY]F[HRV]N[GS]KKS[FIV] AHN[MV]
5	50	3.7*e* − 1083	GYDNRLRP[GN][FL]G[GE][PR][PI][TV][EQ][VI]XT[DN]I[YD][VI][TA] S[FI][GD][PS][VI]S[DE][TV][ND]M[ED]YTI[DT][VI][FY][FL]RQ [SKT]WKDER
6	29	1.7*e* − 538	[DS][VI]S[KA]ID[KR][YW]SRI[VFL]FPV[AL]FG[LF]FN[LV]VYW[AVL] [YTV]Y[LV]
7	29	3.5*e* − 413	Q[FY][DS][LFI][VL]G[QL][TR][VN][GST][TS]E[TI][VI]K[STF]STG[ED] Y[VPT][VIR][ML][TS][VLA][YHS]FH
8	29	2.4*e* − 262	[AFG][RS][LQS][VMY][LGP][AQT][NPS][IS][QLV][EKQ]DE[ALT][KN] [DN]N[IT]T[IV]FTRILD[RG]LLD
9	41	5.8*e* − 174	[TY]W[LK]RGN[DE]S[VL][RK][GT][LD]E[HK][LI][RS]L[AS]Q[YF][TL] I[EQ]R[YF][FH]T[LT][VS]T[RL][SA][QF][QY][ES][TS][GT][NG][YW] [TY][RN]

represents the frequency of one hydrophobic amino acid followed by another amino acid with different hydrophobic properties, and D represents the first, 25%, 50%, 75%, and last position of the amino acids that satisfy certain properties in the sequence; for details, see [44]. In addition to this 188D feature vector extraction method, we used two web-based servers, PseAAC and ProtrWeb, for 80D and 70D feature vectors, respectively. The limited amount of sequence on the web allowed the analysis of only human $GABA_A$Rs and the corresponding non-$GABA_A$Rs by using the last two methods.

The abnormities of $GABA_A$Rs are associated with the pathology and progression of several neurological and psychiatric diseases, such as autism, schizophrenia [73], and alcoholism [74], particularly in epilepsy [75–79], Dravet syndrome [80], asthma [81], breast cancer [82], some psychiatric diseases [83], Alzheimer disease [84], and other neurodegenerative diseases. It is recently reported that $GABA_A$R may be involved in apoptosis in preeclampsia [85]. Human $GABA_A$Rs conserved motifs analyses indicate that motifs 1, 3, and 6 are the frame of neurotransmitter-gated ion channel transmembrane region, which form the ion channel for cation transporter by the construction of transmembrane helix. Motifs 2, 4, and 5 are also composed of neurotransmitter-gated ion channel extracellular ligand binding domain by linking closely and forming a pentameric arrangement in the structure [86]. Various GABA receptor genes are associated with many mental-disorder-related phenotypes. Alterations in GABAergic inhibitory actions, such as the subunit amount, composition, and gene expression of $GABA_A$Rs, may demonstrate neurophysiologic and functional consequences related to mental disorders. Some studies on protein prediction using Chou's method have been reported in 2011 because of the importance of $GABA_A$Rs [11]. However, similar studies on $GABA_A$Rs are rarely reported since then.

The current results showed that our method reached the most correctly classified instances at 96.8%; it suggested that our 188D algorithm performed well for classification and could correctly discriminate both positive and negative samples with relative high specificity. However, the Sn, Acc, and MCC indexes were lower than those of the PseAAC and ProtrWeb methods; this might be due to the large dataset size of our work. But the lowest value was higher than 85%. Overall, our project, which is mainly based on physicochemical properties, can reflect the characteristics of protein sequences and can be applied in the prediction of $GABA_A$Rs classification. Definitely, it needs to develop more precise methods based on 188D.

Competing Interests

The authors declare that there are no competing interests.

Acknowledgments

The work was supported by the Natural Science Foundation of Fujian Province of China (no. 2016J01152) and National Natural Science Foundation of China (no. 61573235, no. 61272315, and no. 61302139).

References

[1] K. E. S. Locock, I. Yamamoto, P. Tran et al., "γ-aminobutyric acid(C) (GABAC) selective antagonists derived from the bioisosteric modification of 4-aminocyclopent-1-enecarboxylic acid: amides and hydroxamates," *Journal of Medicinal Chemistry*, vol. 56, no. 13, pp. 5626–5630, 2013.

[2] A. Mayerhofer, B. Höhne-Zell, K. Gamel-Didelon et al., "Gamma-aminobutyric acid (GABA): a para- and/or autocrine hormone in the pituitary," *The FASEB Journal*, vol. 15, no. 6, pp. 1089–1091, 2001.

[3] N. Okai, C. Takahashi, K. Hatada, C. Ogino, and A. Kondo, "Disruption of *pknG* enhances production of gamma-aminobutyric acid by *Corynebacterium glutamicum* expressing glutamate decarboxylase," *AMB Express*, vol. 4, no. 1, article 20, pp. 1–8, 2014.

[4] F. C. Pereira, M. R. Rolo, E. Marques et al., "Acute increase of the glutamate-glutamine cycling in discrete brain areas after administration of a single dose of amphetamine," *Annals of the New York Academy of Sciences*, vol. 1139, pp. 212–221, 2008.

[5] M. Rigby, S. G. Cull-Candy, and M. Farrant, "Transmembrane AMPAR regulatory protein γ-2 is required for the modulation of GABA release by presynaptic AMPARs," *The Journal of Neuroscience*, vol. 35, no. 10, pp. 4203–4214, 2015.

[6] M. Zonouzi, J. Scafidi, P. Li et al., "GABAergic regulation of cerebellar NG2 cell development is altered in perinatal white matter injury," *Nature Neuroscience*, vol. 18, no. 5, pp. 674–682, 2015.

[7] T. Irie, R. Kikura-Hanajiri, M. Usami, N. Uchiyama, Y. Goda, and Y. Sekino, "MAM-2201, a synthetic cannabinoid drug of abuse, suppresses the synaptic input to cerebellar Purkinje cells via activation of presynaptic CB1 receptors," *Neuropharmacology*, vol. 95, pp. 479–491, 2015.

[8] S. R. Zukin, A. B. Young, and S. H. Snyder, "Gamma-aminobutyric acid binding to receptor sites in the rat central nervous system," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 71, no. 12, pp. 4802–4807, 1974.

[9] R. W. Olsen and W. Sieghart, "International union of pharmacology. LXX. Subtypes of γ-aminobutyric acidA receptors: classification on the basis of subunit composition, pharmacology, and function. Update," *Pharmacological Reviews*, vol. 60, no. 3, pp. 243–260, 2008.

[10] J.-M. Fritschy, "Epilepsy, E/I balance and $GABA_A$ receptor plasticity," *Frontiers in Molecular Neuroscience*, vol. 1, article 5, 2008.

[11] H. Mohabatkar, M. Mohammad Beigi, and A. Esmaeili, "Prediction of $GABA_A$ receptor proteins using the concept of Chou's pseudo-amino acid composition and support vector machine," *Journal of Theoretical Biology*, vol. 281, no. 1, pp. 18–23, 2011.

[12] P. S. Miller and T. G. Smart, "Binding, activation and modulation of Cys-loop receptors," *Trends in Pharmacological Sciences*, vol. 31, no. 4, pp. 161–174, 2010.

[13] A. Pöltl, B. Hauer, K. Fuchs, V. Tretter, and W. Sieghart, "Subunit composition and quantitative importance of GABAA receptor subtypes in the cerebellum of mouse and rat," *Journal of Neurochemistry*, vol. 87, no. 6, pp. 1444–1455, 2003.

[14] G. Grenningloh, E. Gundelfinger, B. Schmitt et al., "Glycine vs GABA receptors," *Nature*, vol. 330, no. 6143, pp. 25–26, 1987.

[15] P. S. Miller and A. R. Aricescu, "Crystal structure of a human $GABA_A$ receptor," *Nature*, vol. 512, no. 7514, pp. 270–275, 2014.

[16] W. Sieghart and G. Sperk, "Subunit composition, distribution and function of GABA(A) receptor subtypes," *Current Topics in Medicinal Chemistry*, vol. 2, no. 8, pp. 795–816, 2002.

[17] P. H. Torkkeli, H. Liu, and A. S. French, "Transcriptome analysis of the central and peripheral nervous systems of the spider *Cupiennius salei* reveals multiple putative Cys-loop ligand gated ion channel subunits and an acetylcholine binding protein," *PLoS ONE*, vol. 10, no. 9, Article ID e0138068, 2015.

[18] C. A. Reid, S. F. Berkovic, and S. Petrou, "Mechanisms of human inherited epilepsies," *Progress in Neurobiology*, vol. 87, no. 1, pp. 41–57, 2009.

[19] J. Simon, H. Wakimoto, N. Fujita, M. Lalande, and E. A. Barnard, "Analysis of the set of $GABA_A$ receptor genes in the human genome," *The Journal of Biological Chemistry*, vol. 279, no. 40, pp. 41422–41435, 2004.

[20] I.-C. Chou, C.-C. Lee, C.-H. Tsai et al., "Association of GABRG2 polymorphisms with idiopathic generalized epilepsy," *Pediatric Neurology*, vol. 36, no. 1, pp. 40–44, 2007.

[21] K. Bethmann, J.-M. Fritschy, C. Brandt, and W. Löscher, "Antiepileptic drug resistant rats differ from drug responsive rats in GABAA receptor subunit expression in a model of temporal lobe epilepsy," *Neurobiology of Disease*, vol. 31, no. 2, pp. 169–187, 2008.

[22] M. SidAhmed-Mezi, I. Kurcewicz, C. Rose et al., "Mass spectrometric detection and characterization of atypical membrane-bound zinc-sensitive phosphatases modulating GABAA receptors," *PLoS ONE*, vol. 9, no. 6, Article ID e100612, 2014.

[23] J. Uwera, S. Nedergaard, and M. Andreasen, "A novel mechanism for the anticonvulsant effect of furosemide in rat hippocampus in vitro," *Brain Research*, vol. 1625, pp. 1–8, 2015.

[24] J. L. Fisher, "The anti-convulsant stiripentol acts directly on the $GABA_A$ receptor as a positive allosteric modulator," *Neuropharmacology*, vol. 56, no. 1, pp. 190–197, 2009.

[25] J.-Q. Kang, W. Shen, C. Zhou, D. Xu, and R. L. Macdonald, "The human epilepsy mutation GABRG2(Q390X) causes chronic subunit accumulation and neurodegeneration," *Nature Neuroscience*, vol. 18, no. 7, pp. 988–996, 2015.

[26] S. Hirose, "Mutant GABA(A) receptor subunits in genetic (idiopathic) epilepsy," *Progress in Brain Research*, vol. 213, pp. 55–85, 2014.

[27] H. Ding, S.-H. Guo, E.-Z. Deng et al., "Prediction of Golgi-resident protein types by using feature selection technique," *Chemometrics and Intelligent Laboratory Systems*, vol. 124, pp. 9–13, 2013.

[28] W.-C. Li, E.-Z. Deng, H. Ding, W. Chen, and H. Lin, "iORI-PseKNC: a predictor for identifying origin of replication with pseudo *k*-tuple nucleotide composition," *Chemometrics and Intelligent Laboratory Systems*, vol. 141, pp. 100–106, 2015.

[29] H. Lin, W. Chen, and H. Ding, "AcalPred: a sequence-based tool for discriminating between acidic and alkaline enzymes," *PLoS ONE*, vol. 8, no. 10, Article ID e75726, 2013.

[30] L.-F. Yuan, C. Ding, S.-H. Guo, H. Ding, W. Chen, and H. Lin, "Prediction of the types of ion channel-targeted conotoxins based on radial basis function network," *Toxicology in Vitro*, vol. 27, no. 2, pp. 852–856, 2013.

[31] B. Liu, D. Zhang, R. Xu et al., "Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection," *Bioinformatics*, vol. 30, no. 4, pp. 472–479, 2014.

[32] J. Chen, X. Wang, and B. Liu, "IMiRNA-SSF: improving the identification of MicroRNA precursors by combining negative sets with different distributions," *Scientific Reports*, vol. 6, Article ID 19062, 2016.

[33] A. Besga, I. Gonzalez, E. Echeburua et al., "Discrimination between Alzheimer's disease and late onset bipolar disorder using multivariate analysis," *Frontiers in Aging Neuroscience*, vol. 7, article 231, 2015.

[34] Q. Yang, H.-Y. Zou, Y. Zhang et al., "Multiplex protein pattern unmixing using a non-linear variable-weighted support vector machine as optimized by a particle swarm optimization algorithm," *Talanta*, vol. 147, pp. 609–614, 2016.

[35] R. Wang, Y. Xu, and B. Liu, "Recombination spot identification Based on gapped k-mers," *Scientific Reports*, vol. 6, article 23934, 2016.

[36] A. K. Sharma, S. Kumar, K. Harish, D. B. Dhakan, and V. K. Sharma, "Prediction of peptidoglycan hydrolases—a new class of antibacterial proteins," *BMC Genomics*, vol. 17, no. 1, article 411, 2016.

[37] Z. C. Li, M. H. Huang, W. Q. Zhong et al., "Identification of drug-target interaction from interactome network with 'guilt-by-association' principle and topology features," *Bioinformatics*, vol. 32, no. 7, pp. 1057–1064, 2016.

[38] J. J. Jones, B. E. Wilcox, R. W. Benz et al., "A plasma-based protein marker panel for colorectal cancer detection identified by multiplex targeted mass spectrometry," *Clinical Colorectal Cancer*, vol. 15, no. 2, pp. 186–194.e13, 2016.

[39] I. Semanjski and S. Gautama, "Smart city mobility application—gradient boosting trees for mobility prediction and analysis based on crowdsourced data," *Sensors*, vol. 15, no. 7, pp. 15974–15987, 2015.

[40] R. Johnson and T. Zhang, "Learning nonlinear functions using regularized greedy forest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 942–954, 2014.

[41] K. P. Singh and S. Gupta, "In silico prediction of toxicity of non-congeneric industrial chemicals using ensemble learning based modeling approaches," *Toxicology and Applied Pharmacology*, vol. 275, no. 3, pp. 198–212, 2014.

[42] Y. Chen, Z. Jia, D. Mercola, and X. Xie, "A gradient boosting algorithm for survival analysis via direct optimization of concordance index," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 873595, 8 pages, 2013.

[43] A. Decruyenaere, P. Decruyenaere, P. Peeters, F. Vermassen, T. Dhaene, and I. Couckuyt, "Prediction of delayed graft function after kidney transplantation: comparison between logistic regression and machine learning methods," *BMC Medical Informatics and Decision Making*, vol. 15, article 83, 2015.

[44] C. Lin, Y. Zou, J. Qin et al., "Hierarchical classification of protein folds using a novel ensemble classifier," *PLoS ONE*, vol. 8, no. 2, Article ID e56499, 2013.

[45] Q. Zou, X. Li, Y. Jiang, Y. Zhao, and G. Wang, "Binmempredict: a web server and software for predicting membrane protein types," *Current Proteomics*, vol. 10, no. 1, pp. 2–9, 2013.

[46] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT Suite: a web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, 2010.

[47] C. Liu, P. Su, R. Li et al., "Molecular cloning, expression pattern, and molecular evolution of the spleen tyrosine kinase in lamprey, *Lampetra japonica*," *Development Genes and Evolution*, vol. 225, no. 2, pp. 113–120, 2015.

[48] L. Song, D. Li, X. Zeng, Y. Wu, L. Guo, and Q. Zou, "nDNA-prot: identification of DNA-binding proteins based on unbalanced classification," *BMC Bioinformatics*, vol. 15, article 298, 2014.

[49] Q. Zou, J. Guo, Y. Ju, M. Wu, X. Zeng, and Z. Hong, "Improving tRNAscan-SE annotation results via ensemble classifiers," *Molecular Informatics*, vol. 34, no. 11-12, pp. 761–770, 2015.

[50] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, 2014.

[51] Q. Zou, Z. Wang, X. Guan, B. Liu, Y. Wu, and Z. Lin, "An approach for identifying cytokines based on a novel ensemble classifier," *BioMed Research International*, vol. 2013, Article ID 686090, 11 pages, 2013.

[52] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human microRNAs by incorporatinga high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.

[53] X. Zeng, S. Yuan, X. Huang, and Q. Zou, "Identification of cytokine via an improved genetic algorithm," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 643–651, 2015.

[54] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[55] H. Ding, Z. Y. Liang, F. B. Guo, J. Huang, W. Chen, and H. Lin, "Predicting bacteriophage proteins located in host cell with feature selection technique," *Computers in Biology and Medicine*, vol. 71, pp. 156–161, 2016.

[56] H. Tang, W. Chen, and H. Lin, "Identification of immunoglobulins using Chou's pseudo amino acid composition with feature selection technique," *Molecular BioSystems*, vol. 12, no. 4, pp. 1269–1275, 2016.

[57] L. Wei, Q. Zou, M. Liao, H. Lu, and Y. Zhao, "A novel machine learning method for cytokine-receptor interaction prediction," *Combinatorial Chemistry & High Throughput Screening*, vol. 19, no. 2, pp. 144–152, 2016.

[58] E. Frank, M. Hall, L. Trigg, G. Holmes, and I. H. Witten, "Data mining in bioinformatics using Weka," *Bioinformatics*, vol. 20, no. 15, pp. 2479–2481, 2004.

[59] T. C. Smith and E. Frank, "Introducing machine learning concepts with WEKA," in *Statistical Genomics*, E. Mathé and S. Davis, Eds., vol. 1418 of *Methods in Molecular Biology*, pp. 353–378, Springer, Berlin, Germany, 2016.

[60] T. L. Bailey, J. Johnson, C. E. Grant, and W. S. Noble, "The MEME Suite," *Nucleic Acids Research*, vol. 43, no. W1, pp. W39–W49, 2015.

[61] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins*, vol. 43, no. 3, pp. 246–255, 2001.

[62] H.-B. Shen and K.-C. Chou, "PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition," *Analytical Biochemistry*, vol. 373, no. 2, pp. 386–388, 2008.

[63] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. W1, pp. W65–W71, 2015.

[64] N. Xiao, D.-S. Cao, M.-F. Zhu, and Q.-S. Xu, "Protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences," *Bioinformatics*, vol. 31, no. 11, pp. 1857–1859, 2015.

[65] W. Chen, P.-M. Feng, E.-Z. Deng, H. Lin, and K.-C. Chou, "iTIS-PseTNC: a sequence-based predictor for identifying translation initiation site in human genes using pseudo trinucleotide composition," *Analytical Biochemistry*, vol. 462, pp. 76–83, 2014.

[66] B. Liu, F. L. Liu, L. Y. Fang, X. L. Wang, and K.-C. Chou, "repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects," *Bioinformatics*, vol. 31, no. 8, pp. 1307–1309, 2015.

[67] L. Au and D. F. Green, "Direct calculation of protein fitness landscapes through computational protein design," *Biophysical Journal*, vol. 110, no. 1, pp. 75–84, 2016.

[68] J. T. S. Hopper and C. V. Robinson, "Mass spectrometry quantifies protein interactions-from molecular chaperones to membrane porins," *Angewandte Chemie—International Edition*, vol. 53, no. 51, pp. 14002–14215, 2014.

[69] K. Kržišnik and T. Urbic, "Amino acid correlation functions in protein structures," *Acta Chimica Slovenica*, vol. 62, no. 3, pp. 574–581, 2015.

[70] A. Olivera-Nappa, B. A. Andrews, and J. A. Asenjo, "Mutagenesis Objective Search and Selection Tool (MOSST): an algorithm to predict structure-function related mutations in proteins," *BMC Bioinformatics*, vol. 12, article 122, 2011.

[71] C. B. Pinheiro, M. Shah, E. L. Soares et al., "Proteome analysis of plastids from developing seeds of *Jatropha curcas* L.," *Journal of Proteome Research*, vol. 12, no. 11, pp. 5137–5145, 2013.

[72] C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen, "SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3692–3697, 2003.

[73] J. R. Glausier and D. A. Lewis, "Selective pyramidal cell reduction of GABA(A) receptor α1 subunit messenger RNA expression in schizophrenia," *Neuropsychopharmacology*, vol. 36, no. 10, pp. 2103–2110, 2011.

[74] N. Onori, C. Turchi, G. Solito, R. Gesuita, L. Buscemi, and A. Tagliabracci, "GABRA2 and alcohol use disorders: no evidence of an association in an Italian case-control study," *Alcoholism: Clinical and Experimental Research*, vol. 34, no. 4, pp. 659–668, 2010.

[75] H. Yuan, C.-M. Low, O. A. Moody, A. Jenkins, and S. F. Traynelis, "Ionotropic GABA and glutamate receptor mutations and human neurologic diseases," *Molecular Pharmacology*, vol. 88, no. 1, pp. 203–217, 2015.

[76] J. Richetto, M. A. Labouesse, M. M. Poe et al., "Behavioral effects of the benzodiazepine-positive allosteric modulator SH-053-2′F-S-CH_3 in an immune-mediated neurodevelopmental disruption model," *The International Journal of Neuropsychopharmacology*, vol. 18, no. 4, pp. 1–11, 2014.

[77] R. J. Hatch, C. A. Reid, and S. Petrou, "Enhanced in vitro CA1 network activity in a sodium channel β1(C121W) subunit model of genetic epilepsy," *Epilepsia*, vol. 55, no. 4, pp. 601–608, 2014.

[78] R. Kumari, R. Lakhan, J. Kalita, R. K. Garg, U. K. Misra, and B. Mittal, "Potential role of $GABA_A$ receptor subunit; *GABRA6*, *GABRB2* and *GABRR2* gene polymorphisms in epilepsy susceptibility and pharmacotherapy in North Indian population," *Clinica Chimica Acta*, vol. 412, no. 13-14, pp. 1244–1248, 2011.

[79] Y. L. Murashima and M. Yoshii, "New therapeutic approaches for epilepsies, focusing on reorganization of the GABAA receptor subunits by neurosteroids," *Epilepsia*, vol. 51, no. 3, pp. 131–134, 2010.

[80] C. Chiron, "Current therapeutic procedures in Dravet syndrome," *Developmental Medicine and Child Neurology*, vol. 53, supplement 2, pp. 16–18, 2011.

[81] G. Gallos, P. Yim, S. Chang et al., "Targeting the restricted α-subunit repertoire of airway smooth muscle GABAA receptors augments airway smooth muscle relaxation," *American Journal of Physiology—Lung Cellular and Molecular Physiology*, vol. 302, no. 2, pp. L248–L256, 2012.

[82] G. M. Sizemore, S. T. Sizemore, D. D. Seachrist, and R. A. Keri, "GABA(A) receptor Pi (GABRP) stimulates basal-like breast cancer cell migration through activation of extracellular-regulated kinase 1/2 (ERK1/2)," *Journal of Biological Chemistry*, vol. 289, no. 35, pp. 24102–24113, 2014.

[83] L. I. Sinclair, P. T. Dineen, and A. L. Malizia, "Modulation of ion channels in clinical psychopharmacology: adults and younger people," *Expert Review of Clinical Pharmacology*, vol. 3, no. 3, pp. 397–416, 2010.

[84] A. S. Al Mansouri, D. E. Lorke, S. M. Nurulain et al., "Methylene blue inhibits the function of α7-nicotinic acetylcholine receptors," *CNS and Neurological Disorders—Drug Targets*, vol. 11, no. 6, pp. 791–800, 2012.

[85] J. Lu, Q. Zhang, D. Tan et al., "GABA A receptor π subunit promotes apoptosis of HTR-8/SVneo trophoblastic cells: implications in preeclampsia," *International Journal of Molecular Medicine*, vol. 38, no. 1, pp. 105–112, 2016.

[86] A. P. Hanek, H. A. Lester, and D. A. Dougherty, "Photochemical proteolysis of an unstructured linker of the GABAAR extracellular domain prevents GABA but not pentobarbital activation," *Molecular Pharmacology*, vol. 78, no. 1, pp. 29–35, 2010.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 4525786, 5 pages
http://dx.doi.org/10.1155/2016/4525786

Research Article

Positive-Unlabeled Learning for Pupylation Sites Prediction

Ming Jiang[1] and Jun-Zhe Cao[2]

[1]*School of Electronic Engineering, Dongguan University of Technology, Dongguan 523808, China*
[2]*School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Ming Jiang; jiangm@dgut.edu.cn

Received 11 May 2016; Revised 26 June 2016; Accepted 5 July 2016

Academic Editor: Qin Ma

Pupylation plays a key role in regulating various protein functions as a crucial posttranslational modification of prokaryotes. In order to understand the molecular mechanism of pupylation, it is important to identify pupylation substrates and sites accurately. Several computational methods have been developed to identify pupylation sites because the traditional experimental methods are time-consuming and labor-sensitive. With the existing computational methods, the experimentally annotated pupylation sites are used as the positive training set and the remaining nonannotated lysine residues as the negative training set to build classifiers to predict new pupylation sites from the unknown proteins. However, the remaining nonannotated lysine residues may contain pupylation sites which have not been experimentally validated yet. Unlike previous methods, in this study, the experimentally annotated pupylation sites were used as the positive training set whereas the remaining nonannotated lysine residues were used as the unlabeled training set. A novel method named PUL-PUP was proposed to predict pupylation sites by using positive-unlabeled learning technique. Our experimental results indicated that PUL-PUP outperforms the other methods significantly for the prediction of pupylation sites. As an application, PUL-PUP was also used to predict the most likely pupylation sites in nonannotated lysine sites.

1. Introduction

Recently, a prokaryotic ubiquitin-like protein (Pup) has been identified in prokaryotes [1, 2]. Pup is an intrinsically disordered protein with 64 amino acids and marks the target proteins which are needed to be degraded [3, 4]. The process of Pup linking substrate lysine by isopeptide bonds is named pupylation which plays an important role in regulating protein degradation and signal transduction in prokaryotic cells [5]. Although pupylation and ubiquitylation are functional analogues, the enzymology involved in them is different [6]. In contrast to ubiquitylation requiring three enzymes E1 (activating enzyme), E2 (conjugating enzyme), and E3 (protein ligase), pupylation requires only two enzymes: the deamidase of Pup (DOP) and the proteasome accessory factor A (PafA) [7].

To understand the molecular mechanisms of pupylation, it is important to identify pupylation substrates and sites accurately. As the large-scale proteomics methods [8–11] are usually time-consuming and labor-intensive, several computational methods have been developed to predict the pupylation sites in recent researches. Liu et al. had developed the first predictor GPS-PUP for the prediction of the pupylation sites on the basis of group-based prediction system (GPS) 2.2 algorithm [12]; Tung developed a predictor, iPUP, by using SVM algorithm and the composition of k-space amino acid pairs (CKSAAPs) feature [13]; Chen et al. also proposed SVM-based predictor named PupPred, in which amino acid pairs feature was employed to encode lysine-centered peptides [14]. Recently, Hasan et al. introduced a Profile-Based Composition of k-Spaced Amino Acid Pairs for the prediction of protein pupylation sites and built a web server named pbPUP [15].

Note that in the aforementioned three existing computational methods, the experimentally annotated pupylation sites are used as the positive training set and the remaining nonannotated lysine residues are used as the negative training set to build classifiers for prediction of new pupylation sites from the unknown proteins. However, due to the limitations of experimental condition and technique, the remaining

nonannotated lysine residues may contain some pupylation sites which are not experimentally validated yet [13, 14]. Thus, the classifiers are actually trained on a noisy negative set. As a result, the performance of the classifiers may not be as good as it was supposed to be.

In contrast to existing prediction methods, experimentally annotated pupylation sites were used as the positive training set and the remaining nonannotated lysine residues were used as the unlabeled training set in this study. We developed a novel method to predict pupylation sites by using the positive-unlabeled (PU) learning technique. This method was called PUL-PUP (PU learning for pupylation sites prediction). Experimental results show that the performance of our method significantly outperforms the other methods on both training and test sets. As an application, the most likely pupylation sites were predicted in nonannotated lysine sites by the method we proposed in this paper. PUL-PUP Matlab software package is freely accessible at https://pul-pup.github.io/.

2. Materials and Methods

2.1. Dataset. Tung's training set and independent test set [13] were used in this study. The training set consisted of 162 proteins with 183 experimentally annotated pupylation sites and 2258 nonannotated pupylation sites; the independent test set consisted of 20 proteins with 29 experimentally annotated pupylation sites and 408 nonannotated pupylation sites. Sliding window method was used to encode every lysine residue K of dataset because pupylation only occurred in lysine residues K. According to [13], window size was selected as 21 in our study.

2.2. Feature Extraction and Feature Selection. The CKSAAP encoding has been widely used to various posttranslational modifications' site prediction [16–18]. The CKSAAP features [13, 19] with $k = 0$, 1, 2, 3, and 4 were used to encode each residue of lysine fragment in this study. Thus, each sample was represented by 2205 features. In Tung's paper [13], chi-square test and backward feature elimination algorithm were used to remove the irrelevant and redundant features. Firstly, chi-square test was employed to rank the importance of the 2205 features. Then, the backward feature selection algorithm was used to eliminate 50 features with the lowest ranks in each iteration. Here, the top 150 CKSAAP features were selected as optimal feature set which were also same as Tung's paper [13].

2.3. Development of PUL-PUP. The experimentally annotated pupylation sites were used as the positive training set and the remaining nonannotated lysine residues were used as the unlabeled training set to build classifier in this study. In this way, two types of subset were received in the training set: (1) the positive dataset P and (2) the unlabeled dataset U. Thus our problem became learning from positive and unlabeled samples. We proposed a novel PU learning algorithm named PUL-PUP to predict pupylation sites. The core learning algorithm of PUL-PUP is support vector machine (SVM) which has been widely used in various biological problems [20–22]. The flowchart of PUL-PUP algorithm is shown as follows:

Input

(i) positive training data P

(ii) unlabeled data U

Output

(i) final classifier f

Stage 1 (selection of initial reliable negatives).

(i) $\mathrm{RN}^0 = \arg\max_{N \subset U, |N|=|P|} d(N, P)$

Stage 2 (expansion of reliable negative example set).

(i) $i = 0$;

(ii) Repeat

(iii) $U = U \setminus \mathrm{RN}^i$;

(iv) Construct two-class SVM f^i based on P and RN^i;

(v) Classify U by f^i;

(vi) N^i_{pred} is the predicted negative set, where $|N^i_{\mathrm{pred}}| \leq 2 * |P|$ and $f^i(N^i_{\mathrm{pred}}) < -0.25$;

(vii) $\mathrm{RN}^{i+1} = N^i_{\mathrm{pred}} \cup N^i_{\mathrm{sv}} \cup \widetilde{N}^i_{\mathrm{sv}}$ where N^i_{sv} is the negative SVs of f^i, $\widetilde{N}^i_{\mathrm{sv}}$ is the surrounding points of N^i_{sv} in N^i and $|N^i_{\mathrm{sv}}| = |\widetilde{N}^i_{\mathrm{sv}}|$;

(viii) $i = i + 1$;

(ix) until $|U| \leq 4 * |P|$;

Stage 3 (acquisition of final classifier).

(i) A final SVM classifier f was trained on positive set P and representative reliable negative set RN

There are three stages in PUL-PUP algorithm as follows.

Stage 1 (selection of initial reliable negatives). PUL-PUP selected the initial reliable negative set RN^0 from unlabeled set U by maximum distance rule. RN^0 should be located as far away from P as possible to ensure that the reliable negative set was the most dissimilar from the positive set P. Therefore, RN^0 would satisfy the formula described below:

$$\mathrm{RN}^0 = \arg \max_{\substack{N \subset U \\ |N|=|P|}} d(N, P), \tag{1}$$

where $d(N, P)$ is Euclidean distance between N and P:

$$d(N, P) = \min_{p \in P} \sum_{n \in N} \|n - p\|. \tag{2}$$

Stage 2 (expansion of reliable negative example set). After the selection of initial reliable negative set, PUL-PUP algorithm iteratively trained a series of two-class SVM classifiers and gradually extended reliable negative set. Specifically, at the ith

iteration, an SVM classifier f^i was firstly trained in positive set P and current reliable negative training set RN^i; then, f^i would be used to classify the current unlabeled set U^i and calculate its decision value. To guarantee the reliability of the negative set, samples with the decision value less than a threshold (T) were selected as newly predicted negatives N^i_{pred}; here T was set to −0.25. To overcome the problem of imbalance during the iteration, the negative support vectors N^i_{sv} and their surrounding points in RN^i, named $\widetilde{N}^i_{\mathrm{sv}}$, were used to represent the existing negative set RN^i, and the size of N^i_{pred} was controlled less than $2 * |P|$. At the $i+1$th iteration, $U^{i+1} = U^i \backslash N^i_{\mathrm{pred}}$; $\mathrm{RN}^{i+1} = N^i_{\mathrm{pred}} \cup N^i_{\mathrm{sv}} \cup \widetilde{N}^i_{\mathrm{sv}}$. Classifier f^{i+1} was trained in positive set P and current reliable negative training set RN^{i+1}. As this process continues, RN^{i} may contain more and more false positive examples; therefore, iteration should be terminated at some point. Iteration was repeated until the size of U^i goes below a threshold $r * |P|$; here r was set to 4.

Stage 3 (acquisition of final classifier). After the extraction of representative reliable negative set, a final SVM classifier f was trained on positive set P and representative reliable negative set RN.

2.4. SVM Parameter Selection. The core learning algorithm of PUL-PUP is support vector machine (SVM) with radial basis function (RBF) kernel. Libsvm [23] was used for training SVM models, and the grid search method was applied to tune the parameters in cross-validation. Parameter C was selected from {0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000}; and kernel parameter γ was selected from {0.00001, 0.00002, 0.00005, 0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1}. The parameters of SVM were fixed during the expansion of reliable negative example set.

2.5. Performance Evaluation of PUL-PUP. Five widely accepted measurements, including sensitivity (Sn), specificity (Sp), accuracy (ACC), Matthew's correlation coefficient (MCC), and area under receiver operating characteristic curve (AUC), were used to evaluate prediction performances of PUL-PUP. They are defined as

$$\begin{aligned}
\mathrm{Sn} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\\
\mathrm{Sp} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}},\\
\mathrm{ACC} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}},\\
\mathrm{MCC} &= \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FN}\times\mathrm{FP}}{\sqrt{(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})}},
\end{aligned} \tag{3}$$

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively.

Table 1: 10-fold cross-validation performance of PUL-PUP, PSoL, SVM, and SVM_balance.

Method	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
PUL-PUP	82.24	91.57	88.92	0.74	0.92
PSoL	67.50	73.60	70.55	0.42	0.80
SVM_balance	76.71	63.65	69.88	0.40	0.77

Table 2: Independent test performance of PUL-PUP, PSoL, SVM, and SVM_balance.

Method	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
PUL-PUP	68.97	70.83	70.71	0.22	0.77
PSoL	51.72	73.14	71.62	0.13	0.74
SVM_balance	62.07	67.40	67.05	0.15	0.70

3. Results and Discussions

3.1. Performance of 10-Fold Cross-Validation on Training Set. In order to evaluate the effectiveness of the selected representative reliable negative samples on pupylation sites prediction, we compared our method with two other methods including SVM_balance and PSoL [24] on training set because the core learning algorithm of our method was SVM and our method was inspired by PSoL. For PUL-PUP and PSoL algorithms, the nonannotated lysine sites were used as the unlabeled training samples. The 10-fold cross-validation of them was performed on positive set P and representative reliable negative set RN. For SVM_balance, a balanced negative training set which had the same size with the positive training set was randomly selected from the nonannotated lysine sites and the 10-fold cross-validation was also performed on the positive training set and the balanced negative training set to find the best parameters of SVM. The 10-fold cross-validation of the four methods was shown in Table 1. As shown in Table 1, PUL-PUP reached the highest Sn, Sp, ACC, MCC, and AUC values of 82.24%, 91.57%, 88.92%, 0.74, and 0.92, respectively, on training dataset. As the selected representative reliable negative samples, the PUL-PUP achieved an excellent performance on training set.

3.2. Comparison of PUL-PUP with Other Methods on Independent Test Set. To further evaluate the performance of pupylation sites prediction by PUL-PUP, we firstly compared it with PSoL and SVM_balance on independent test set. The compared results of different methods are shown in Table 2. Although SVM_balance can avoid the imbalanced problem, the performance of SVM_balance cannot be as good as the PUL-PUP because the negative training set in SVM_balance is randomly selected and cannot truly reflect the distribution of negative set well. It should be pointed out that stage 2 of PUL-PUP was similar to the negative set expansion in PSoL. But, in PUL-PUP, RN^i was represented by $N^i_{\mathrm{sv}} \cup \widetilde{N}^i_{\mathrm{sv}}$ rather than N^i_{sv} merely. Thus, more information in RN^i is included and makes our algorithm more effective than PSoL.

We also compared our method with three existing pupylation sites predictors: GPS-PUP [12], iPUP [13], and pbPUP [15] on independent test set. Three thresholds of "High,"

Table 3: Independent test performance of PUL-PUP and three existing pupylation sites predictors.

Method	Threshold	Sn (%)	Sp (%)	ACC (%)	MCC	AUC
GPS-PUP	High	31.03	89.46	85.62	0.16	0.60
	Medium	34.48	85.54	82.19	0.14	
	Low	41.38	76.72	74.43	0.10	
iPUP	High	48.28	82.84	80.55	0.20	0.66
	Medium	51.72	76.47	74.83	0.16	
	Low	55.17	72.06	70.94	0.15	
pbPUP	High	17.24	88.48	83.75	0.04	0.60
	Medium	31.03	80.15	76.89	0.07	
	Low	41.38	69.85	67.96	0.07	
PUL-PUP	High	51.72	83.33	81.24	0.22	0.77
	Medium	65.52	76.72	75.97	0.24	
	Low	68.97	72.79	72.54	0.23	

"Medium," and "Low" were defined for PUL-PUP according to the SVM scores which were higher than 0.9672, 0.4032, and 0.1088, respectively. The performances of PUL-PUP and three existing pupylation sites predictors were shown in Table 3. As we can see from Table 3, the performance of our algorithm outperformed the existing three predictors significantly. Taking threshold "Medium," for example, the MCC of PUL-PUP (0.24) was higher than that of GPS-PUP (0.14), iPUP (0.16), and pbPUP (0.07). Moreover, PUL-PUP achieved the highest AUC value (0.77). As our classifier is iteratively trained on the positive and reliable negative set in this paper, the performance of our algorithm outperformed the existing three predictors significantly. This demonstrates that PUL-PUP is more suitable for predicting the pupylation sites than other methods.

3.3. Prediction of the Most Likely Pupylation Sites in Nonannotated Lysine Sites. For the 183 pupylated proteins in PupDB [6], there are 212 experimentally annotated pupylation sites and 2666 nonannotated lysine sites. As mentioned earlier, those nonannotated lysine sites may contain some pupylation sites which have not been experimentally validated yet. To predict the most likely pupylation sites in nonannotated lysine sites, we run PUL-PUP algorithm on all data of the PupDB. The top 20 most likely pupylation sites in nonannotated lysine sites were listed in Supplementary S1 (see Supplementary Material available online at http://dx.doi.org/10.1155/2016/4525786). Here, we just give a possible hypothesis; whether those sites will cause pupylation or not remains to be experimentally verified.

4. Conclusions

In this study, we have developed novel pupylation sites prediction method PUL-PUP by using the PU learning. To the best of our knowledge, this is the first time PU learning has been applied to predict the pupylation sites. Experimental results have shown that our method outperformed the existing pupylation sites predictors significantly. Moreover, the most likely pupylation sites were predicted in nonannotated lysine sites by using PUL-PUP. We believe that our method can also be applied to predict the other types of posttranslational modification sites. In future research, we will develop a web server for the PUL-PUP.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61502074), the Social Science and Technology Development Program of Dongguan, China (2013108101007), "Strategy of Enhancing School with Innovation" in Higher Education of Guangdong, China (2014KQNCX221), Dalian University of Technology Fundamental Research Fund (no. DUT15RC(3)030), and the China Postdoctoral Science Foundation (Grant no. 2016M591430).

References

[1] M. J. Pearce, J. Mintseris, J. Ferreyra, S. P. Gygi, and K. H. Darwin, "Ubiquitin-like protein involved in the proteasome pathway of *Mycobacterium tuberculosis*," *Science*, vol. 322, no. 5904, pp. 1104–1107, 2008.

[2] K. E. Burns, W.-T. Liu, H. I. M. Boshoff, P. C. Dorrestein, and C. E. Barry III, "Proteasomal protein degradation in mycobacteria is dependent upon a prokaryotic ubiquitin-like protein," *The Journal of Biological Chemistry*, vol. 284, no. 5, pp. 3069–3075, 2009.

[3] X. Chen, W. C. Solomon, Y. Kang, F. Cerda-Maira, K. H. Darwin, and K. J. Walters, "Prokaryotic ubiquitin-like protein pup is intrinsically disordered," *Journal of Molecular Biology*, vol. 392, no. 1, pp. 208–217, 2009.

[4] S. Liao, Q. Shang, X. Zhang, J. Zhang, C. Xu, and X. Tu, "Pup, a prokaryotic ubiquitin-like protein, is an intrinsically disordered protein," *Biochemical Journal*, vol. 422, no. 2, pp. 207–215, 2009.

[5] J. Herrmann, L. O. Lerman, and A. Lerman, "Ubiquitin and ubiquitin-like proteins in protein regulation," *Circulation Research*, vol. 100, no. 9, pp. 1276–1291, 2007.

[6] C.-W. Tung, "PupDB: a database of pupylated proteins," *BMC Bioinformatics*, vol. 13, no. 1, article 40, 2012.

[7] F. Striebel, F. Imkamp, M. Sutter, M. Steiner, A. Mamedov, and E. Weber-Ban, "Bacterial ubiquitin-like modifier Pup is deamidated and conjugated to substrates by distinct but homologous enzymes," *Nature Structural & Molecular Biology*, vol. 16, no. 6, pp. 647–651, 2009.

[8] C. Poulsen, Y. Akhter, A. H.-W. Jeon et al., "Proteome-wide identification of mycobacterial pupylation targets," *Molecular Systems Biology*, vol. 6, no. 1, 2010.

[9] R. A. Festa, F. McAllister, M. J. Pearce et al., "Prokayrotic ubiquitin-like protein (Pup) proteome of *Mycobacterium tuberculosis*," *PLoS ONE*, vol. 5, no. 1, Article ID e8589, 2010.

[10] J. Watrous, K. Burns, W.-T. Liu et al., "Expansion of the mycobacterial 'PUPylome'," *Molecular BioSystems*, vol. 6, no. 2, pp. 376–385, 2010.

[11] F. A. Cerda-Maira, F. McAllister, N. J. Bode, K. E. Burns, S. P. Gygi, and K. H. Darwin, "Reconstitution of the *Mycobacterium*

tuberculosis pupylation pathway in *Escherichia coli*," *EMBO Reports*, vol. 12, no. 8, pp. 863–870, 2011.

[12] Z. Liu, Q. Ma, J. Cao, X. Gao, J. Ren, and Y. Xue, "GPS-PUP: computational prediction of pupylation sites in prokaryotic proteins," *Molecular BioSystems*, vol. 7, no. 10, pp. 2737–2740, 2011.

[13] C.-W. Tung, "Prediction of pupylation sites using the composition of *k*-spaced amino acid pairs," *Journal of Theoretical Biology*, vol. 336, pp. 11–17, 2013.

[14] X. Chen, J.-D. Qiu, S.-P. Shi, S.-B. Suo, and R.-P. Liang, "Systematic analysis and prediction of pupylation sites in prokaryotic proteins," *PLoS ONE*, vol. 8, no. 9, Article ID e74002, 2013.

[15] M. M. Hasan, Y. Zhou, X. Lu, J. Li, J. Song, and Z. Zhang, "Computational identification of protein pupylation sites by using profile-based composition of k-spaced amino acid pairs," *PLoS ONE*, vol. 10, no. 6, article e0129635, 2015.

[16] Z. Ju, J. Z. Cao, and H. Gu, "iLM-2L: a two-level predictor for identifying protein lysine methylation sites and their methylation degrees by incorporating K-gap amino acid pairs into Chou's general PseAAC," *Journal of Theoretical Biology*, vol. 385, pp. 50–57, 2015.

[17] Z. Ju, J.-Z. Cao, and H. Gu, "Predicting lysine phosphoglycerylation with fuzzy SVM by incorporating k-spaced amino acid pairs into Chou's general PseAAC," *Journal of Theoretical Biology*, vol. 397, pp. 145–150, 2016.

[18] Z. Ju and H. Gu, "Predicting pupylation sites in prokaryotic proteins using semi-supervised self-training support vector machine algorithm," *Analytical Biochemistry*, vol. 507, pp. 1–6, 2016.

[19] X.-B. Wang, L.-Y. Wu, Y.-C. Wang, and N.-Y. Deng, "Prediction of palmitoylation sites using the composition of k-spaced amino acid pairs," *Protein Engineering, Design and Selection*, vol. 22, no. 11, pp. 707–712, 2009.

[20] J. Z. Zeng, Y. L. Liao, Y. S. Liu, and Q. Zou, "Prediction and validation of disease genes using HeteSim Scores," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[21] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identificationof human microRNAs by incorporatinga high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.

[22] Q. Zou, J. Z. Zeng, L. J. Cao, and R. R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, part 2, pp. 346–354, 2016.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[24] C. Wang, C. Ding, R. F. Meraz, and S. R. Holbrook, "PSoL: a positive sample only learning algorithm for finding non-coding RNA genes," *Bioinformatics*, vol. 22, no. 21, pp. 2590–2596, 2006.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 4236858, 7 pages
http://dx.doi.org/10.1155/2016/4236858

Research Article

Constructing Phylogenetic Networks Based on the Isomorphism of Datasets

Juan Wang,[1] Zhibin Zhang,[1] and Yanjuan Li[2]

[1]*School of Computer Science, Inner Mongolia University, Hohhot 010021, China*
[2]*Department of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China*

Correspondence should be addressed to Juan Wang; wangjuanangle@hit.edu.cn

Received 31 May 2016; Accepted 30 June 2016

Academic Editor: Yungang Xu

Constructing rooted phylogenetic networks from rooted phylogenetic trees has become an important problem in molecular evolution. So far, many methods have been presented in this area, in which most efficient methods are based on the incompatible graph, such as the CASS, the LNETWORK, and the BIMLR. This paper will research the commonness of the methods based on the incompatible graph, the relationship between incompatible graph and the phylogenetic network, and the topologies of incompatible graphs. We can find out all the simplest datasets for a topology G and construct a network for every dataset. For any one dataset $\mathscr{C}$, we can compute a network from the network representing the simplest dataset which is isomorphic to $\mathscr{C}$. This process will save more time for the algorithms when constructing networks.

1. Introduction

The evolutionary history of species is usually represented as a (rooted) phylogenetic tree, in which one species has only one parent. Actually, the evolution of species has caused reticulate events such as hybridizations, horizontal gene transfers, and recombinations [1–5], so species may have more than one parent. Then, the phylogenetic trees cannot describe well the evolutionary history of species. However, phylogenetic networks can represent the reticulate events, and they are a generalization of phylogenetic trees. Phylogenetic networks can also represent the conflicting evolution information that may be from different datasets or different trees [6–9].

Phylogenetic networks can be classified into unrooted [10–12] and rooted networks [4, 13–19]. An unrooted phylogenetic network is an unrooted graph whose leaves are bijectively labelled by the taxa. A rooted phylogenetic network is a rooted directed acyclic graph (DAG for short) whose leaves are bijectively labelled by taxa [20–22]. The rooted phylogenetic networks have been studied widely for representing the evolution of taxa, as evolution of species is inherently directed. The paper will study relevant properties of the rooted phylogenetic networks constructed from the rooted trees.

The algorithms constructing rooted phylogenetic networks from rooted phylogenetic trees are mainly classified into three types: the cluster network [17] based on the Hasse diagram; the galled network [16] based on the seed-growing algorithm; the Cass [23], the Lnetwork [24], and the BIMLR [25] based on the decomposition property of networks. In particular, the third type of methods (Cass, Lnetwork, and BIMLR) can construct more precise networks than the other methods. In the following, unless otherwise specified, we refer to rooted phylogenetic networks as networks.

Let $\mathscr{X}$ be a set of taxa. A proper subset of $\mathscr{X}$ (except for both $\emptyset$ and $\mathscr{X}$) is called a cluster. A cluster C is trivial if $|C| = 1$; otherwise, it is nontrivial. Let T be a rooted phylogenetic tree on $\mathscr{X}$; if there is an edge $e = (u, v)$ in T such that the set of taxa which are descendants of v equals C, we say that T represents C. Figure 1 shows two rooted phylogenetic trees T_1 and T_2 and all nontrivial clusters represented by T_1 and T_2. Here, all trivial clusters are not listed. Given a network N and a cluster C, when just connecting one incoming edge and disconnecting all other incoming edges for each reticulate node (i.e., its incoming edges >1), if there is a tree edge $e = (u, v)$ (i.e., incoming edge of $v \leq 1$) in N such that the set of taxa which are descendants of v equals C, we say that N represents C in the softwired sense. On the other hand, if

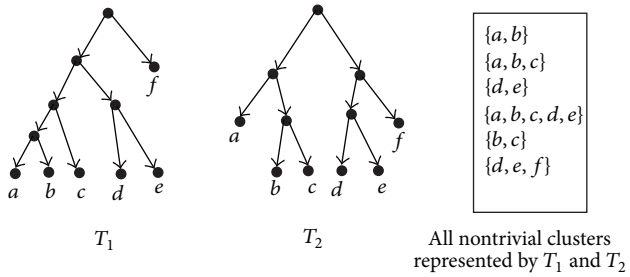


Figure 1: Two rooted phylogenetic trees T_1 and T_2.

there is a tree edge $e = (u, v)$ in N such that the set of taxa which are descendants of v equals C, we say that N represents C in the hardwired sense.

The abovementioned three types of methods constructing networks are based on clusters; that is, they first compute all of the clusters represented by input trees and then construct a network representing all clusters in the softwired sense. In this process, the third type of methods (Cass, Lnetwork, and BIMLR) will recur to the incompatibility graph (will be discussed in the following). This paper will discuss the relationship between the incompatibility graphs and the constructed networks.

2. Preliminaries

A rooted phylogenetic network $N = (V, E)$ on $\mathcal{X}$ is a rooted DAG, and its leaves are bijectively labelled as $\mathcal{X}$. The indegree of a node $v \in V$ is denoted by $\text{indeg}(v)$. A node v with $\text{indeg}(v) \geq 2$ is called a reticulate node, a node v with $\text{indeg}(v) \leq 1$ is called a tree node, and, specially, the tree node with indegree 0 is the root node. The reticulation number in a network $N = (V, E)$ is $\sum_{\text{indeg}(v)>0}(\text{indeg}(v) - 1) = |E| - |V| + 1$.

Given a set of taxa $\mathcal{X}$, two clusters C_1 and C_2 on $\mathcal{X}$ are called compatible, if they are disjoint or one contains the other; that is, $C_1 \cap C_2 = \emptyset$ or $C_1 \subseteq C_2$ or $C_2 \subseteq C_1$; otherwise, they are incompatible. Obviously, a trivial cluster and any one cluster are compatible. Given two incompatible clusters C_1 and C_2, $C_1 \cap C_2$ is called the incompatible taxa with respect to C_1 and C_2. A set of clusters $\mathscr{C}$ on $\mathcal{X}$ is called compatible, if $\mathscr{C}$ is pairwise compatible; otherwise, it is incompatible. For a set of clusters $\mathscr{C}$, its incompatibility graph $\text{IG}(\mathscr{C}) = (V, E)$ is an undirected graph with node set $V = C$ and edge set E, where an edge connects two incompatible clusters.

Given a cluster set $\mathscr{C}$ on $\mathcal{X}$ and a subset S of $\mathcal{X}$, the result of removing all elements in $\mathcal{X} \setminus S$ from each cluster in $\mathscr{C}$ is called the restriction of $\mathscr{C}$ to S, denoted by $\mathscr{C}|_S$. If S (where $|S| > 1$) and any one cluster $C \in \mathscr{C}$ are compatible and $\mathscr{C}|_S$ is also compatible, then we say that S is an ST-set (Strict Tree Set) with respect to $\mathscr{C}$. If there are no other ST-sets containing S except itself, we say that S is maximal. For a maximal ST-set S, there is a subtree constructed by the set of clusters $\{C \mid C \in \mathscr{C}, C \subset S\} \cup S$.

For each maximal ST-set S with respect to $\mathscr{C}$, after collapsing it into a single taxon S, the result set is denoted as $\text{Collapse}(\mathscr{C})$. For example, $\mathscr{C} = \{\{1, 2\}, \{1, 2, 3\}, \{3, 4\}\}, \{1, 2\}$ is the only maximal ST-set; then, $\text{Collapse}(\mathscr{C}) = \{\{3, 4\}, \{\{1, 2\}, 3\}\}$. Then, the taxa of $\text{Collapse}(\mathscr{C})$ are $\{\{1, 2\}, 3, 4\}$, denoted as $\mathcal{X}(\text{Collapse}(\mathscr{C}))$. A set of clusters $\mathscr{C}$ is called the simplest if it has no maximal ST-set with respect to $\mathscr{C}$.

Let $\mathscr{C}$ be a set of clusters on $\mathcal{X}$ and let N be a network representing $\mathscr{C}$. Usually, a tree edge in N can represent more than one cluster in $\mathscr{C}$ and a cluster in $\mathscr{C}$ can be represented by more than one tree edge in N. A mapping ϵ is defined from $\mathscr{C}$ to the set of tree edges of N, such that $\epsilon(C)$ is a tree edge of N that represents C for any one cluster $C \in \mathscr{C}$. A network N is decomposable with respect to $\mathscr{C}$ if there exists a mapping $\epsilon : \mathscr{C} \to E'$ (E' is the set of tree edges of N) such that

(i) for any two clusters $C_1, C_2 \in \mathscr{C}$, C_1 and C_2 lie in the same connected component of the incompatibility graph $\text{IG}(\mathscr{C})$ if and only if two tree edges $\epsilon(C_1)$ and $\epsilon(C_2)$ are contained in the same biconnected component of N.

Then, we also say that the network N has the decomposition property. The decomposition property makes the network constructed by an appropriate divide-and-conquer (DC for short) strategy; that is, it first constructs a subnetwork for each one connected component of the incompatibility graph and then merges all subnetworks into a whole network. Then, the constructed network is called DC network, and the algorithms are called DC algorithms. The paper [23] has proven the DC networks satisfying the decomposition property.

Given a set of clusters $\mathscr{C}$, the DC algorithms first compute the incompatibility graph $\text{IG}(\mathscr{C})$ and then compute the subnetwork for the result set after collapsing each one maximal ST-set into one taxon for each biconnected component of $\text{IG}(\mathscr{C})$; next, "decollapse," that is, replace each leaf labelled by a maximal ST-set by a maximal subtree, and finally integrate those subnetworks into a final network. The paper [25] has proven that there exists a DC network N for any one set of clusters $\mathscr{C}$. Figure 2 shows the construction process of the DC algorithms for the set of clusters in Figure 1, in which constructing subnetwork for each one connected component (i.e., Step 2) is crucial.

The Cass, the Lnetwork, and the BIMLR algorithms are the DC algorithms, which can construct the networks with fewer reticulations than other algorithms. The networks constructed by the BIMLR and the Lnetwork have fewer redundant clusters except for the input clusters than other available methods. When constructing phylogenetic networks, the BIMLR and the Lnetwork are faster than the Cass, and the constructed networks are more stable, that is, the difference between constructed networks for the same dataset when different input orders are used is smaller than the Cass. Figure 3 shows three networks constructed by the Cass for the same dataset with different input orders, while BIMLR and Lnetwork can construct only one network N_1 for the dataset with different input orders [25].

3. Topologies of Incompatibility Graphs

Definition 1. Two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ on $\mathcal{X}$ are isomorphic if and only if there exists a bijection H from V_1 to V_2 such that

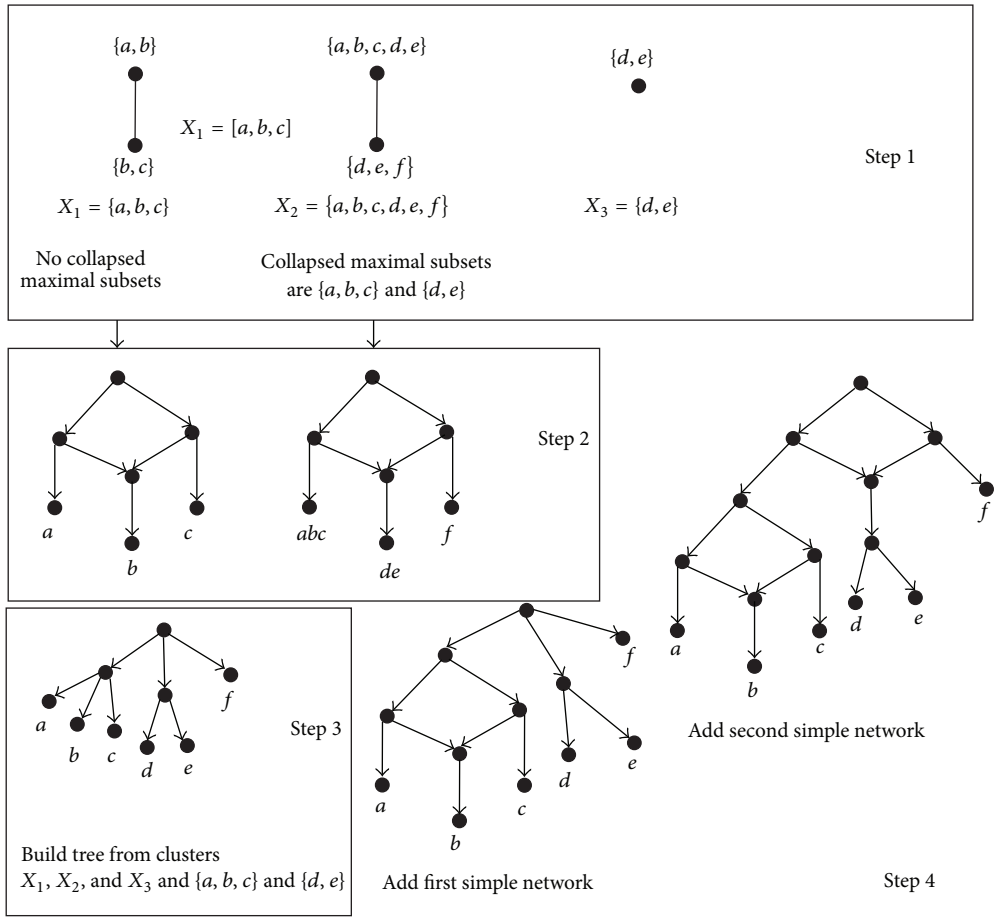


Figure 2: A network constructed by the DC algorithms for the set of clusters in Figure 1.

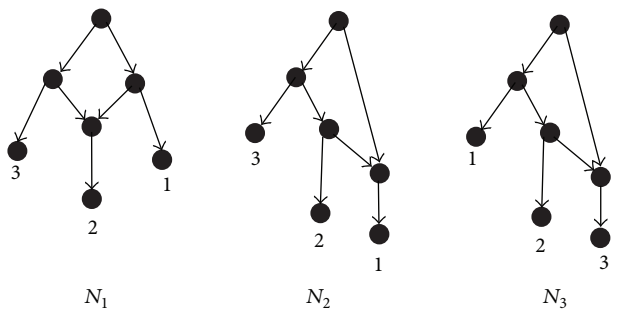


Figure 3: All networks constructed by the Cass for the set of clusters $\mathscr{C} = \{\{1, 2\}, \{2, 3\}\}$.

(i) (u, v) is an edge in E_1 if and only if $(H(u), H(v))$ is an edge in E_2;

(ii) the label of w is equal to the label of $H(w)$ for any one leaf $w \in V_1$.

Given two sets of clusters $\mathscr{C}_1$ on $\mathscr{X}_1$ and $\mathscr{C}_2$ on $\mathscr{X}_2$, let $\mathscr{C}_1'$ and $\mathscr{C}_2'$ be the results after collapsing all maximal ST-sets of $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, $\mathscr{C}_1'$ on $\mathscr{X}_1'$ and $\mathscr{C}_2'$ on $\mathscr{X}_2'$.

Definition 2. $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic, if and only if there is a bijection G from $\mathscr{X}_1'$ to $\mathscr{X}_2'$ such that

(i) a and b are in the same cluster $C_1 \in \mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster $C_2 \in \mathscr{C}_2'$.

By Definition 2, we have that the isomorphism of the cluster sets is an equivalence relation; that is, it is reflexive, symmetric, and transitive.

Lemma 3. *Given a DC network N representing the set of clusters $\mathscr{C}$, then any one maximal ST-set with respect to $\mathscr{C}$ is a maximal subtree in N.*

Proof. From the constructing process of DC networks, this conclusion is obvious. □

Lemma 4. *Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be two sets of clusters on $\mathscr{X}_1$ and $\mathscr{X}_2$, respectively. $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic. There exists a DC network N_1 representing $\mathscr{C}_1$ if and only if there exists a DC network N_2 representing $\mathscr{C}_2$.*

Proof. There must exist a DC network N_1 for $\mathscr{C}_1$. Given a tree edge $e = (u, v)$, the subtree of the root v in N_1 is a maximal subtree if and only if the set of taxa S is a maximal ST-set with

respect to $\mathscr{C}_1$, where the taxa in S are labels of leaves which are descendants of v. Replace each maximal subtree of N_1 by a node, and then denote the result network as N_1'. Obviously, N_1' represents the set of clusters $\mathscr{C}_1'$. From Definition 2, there exists a bijection G from $\mathscr{X}_1'$ to $\mathscr{X}_2'$ such that a and b are in the same cluster $C_1 \in \mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster $C_2 \in \mathscr{C}_2'$.

Then, we can obtain a network N_2' from N_1' by replacing each one taxon a in $\mathscr{X}_1'$ by $G(a)$ in $\mathscr{X}_2'$. Obviously, N_2' represents $\mathscr{C}_2'$. Finally, we replace each leaf labelled by a maximal ST-set with respect to $\mathscr{C}_2$ in N_2' by a maximal subtree, and the result network is denoted as N_2 which represents $\mathscr{C}_2$. □

For two isomorphic sets of clusters $\mathscr{C}_1$ and $\mathscr{C}_2$, let N_1 be a DC network representing $\mathscr{C}_1$. Lemma 4 tells us that there is a DC network N_2 representing $\mathscr{C}_2$, which can be obtained from N_1.

Lemma 5. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a biconnected component with two nodes. Then, any one element $\mathscr{C}$ in $\mathfrak{C}$ is isomorphic to $\mathscr{C}_0 = \{\{1,2\},\{2,3\}\}$.*

Proof. Any one element $\mathscr{C} \in \mathfrak{C}$ has two incompatible clusters. Let $\mathscr{C}_1 = \{C_{11}, C_{12}\}$ and $\mathscr{C}_2 = \{C_{21}, C_{22}\}$ be two sets of clusters in $\mathfrak{C}$, where C_{11} and C_{12} are incompatible and C_{21} and C_{22} are incompatible. Let $A_1 = C_{11} \cap C_{12}$ be the incompatible taxa with respect to C_{11} and C_{12}, and let $A_2 = C_{21} \cap C_{22}$ be the incompatible taxa with respect to C_{21} and C_{22}. Let $B_{11} = C_{11} \setminus A_1$, $B_{12} = C_{12} \setminus A_1$, $B_{21} = C_{21} \setminus A_2$, and $B_{22} = C_{22} \setminus A_2$; then, $\mathscr{C}_1 = \{\{B_{11}, A_1\}, \{B_{12}, A_1\}\}$ and $\mathscr{C}_2 = \{\{B_{21}, A_2\}, \{B_{22}, A_2\}\}$.

Each one of B_{11}, A_1, B_{12}, B_{21}, A_2, and B_{22} is a maximal ST-set if it contains more than one taxon; then, we can collapse it into one taxon which is also denoted by itself. Denote the set of clusters after collapsing all maximal ST-sets as $\mathscr{C}_1'$ and $\mathscr{C}_2'$. Obviously, there is a bijection G from $\mathscr{X}_1' = \{B_{11}, A_1, B_{12}\}$ to $\mathscr{X}_2' = \{B_{21}, A_2, B_{22}\}$, and any two taxa $a, b \in \mathscr{X}_1'$ are in the same cluster in $\mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster in $\mathscr{C}_2'$. Hence, $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic. Accordingly, any one set of clusters $\mathscr{C} \in \mathfrak{C}$ is isomorphic to $\mathscr{C}_0 = \{\{1,2\},\{2,3\}\}$ because $\mathscr{C}_0 \in \mathfrak{C}$. □

For a cluster set $\mathscr{C}$, there may be several cluster sets isomorphic to $\mathscr{C}$, but the simplest set of clusters isomorphic to $\mathscr{C}$ is only one, denoted as $\mathscr{C}_0$. Let N_0 be the DC network representing $\mathscr{C}_0$. Then, we can obtain a DC network representing $\mathscr{C}$ from N_0. Lemmas 4 and 5 show there is a DC network for any one set of clusters whose incompatible graph is a biconnected component with two nodes, and it is obtained from the network N_0 (see Figure 3) representing $\mathscr{C}_0$.

Lemma 6. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a linear biconnected component with three nodes (see Figure 4). Let $\mathscr{C}_1 = \{\{1,3\},\{1,2\},\{1,3,4\}\}$, $\mathscr{C}_2 = \{\{1,3\}, \{1,2,4\},\{1,2,3\}\}$, $\mathscr{C}_3 = \{\{1,2\},\{2,3\},\{3,4\}\}$, and $\mathscr{C}_4 = \{\{1,2\}, \{2,3,5\},\{3,4\}\}$. Then, any one set of clusters $\mathscr{C}$ ($\mathscr{C} \in \mathfrak{C}$) is isomorphic to one of $\mathscr{C}_1$, $\mathscr{C}_2$, $\mathscr{C}_3$, and $\mathscr{C}_4$.*

Proof. Figure 4 shows the topology of the linear biconnected component with three nodes. $\mathscr{C}_i$ is the simplest set of clusters,

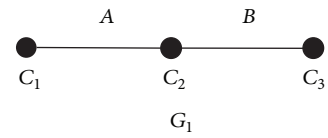


Figure 4: The topology of the linear biconnected component with three nodes.

and its incompatible graph is the topology in Figure 4. Next, we will prove that $\mathscr{C}_i$ ($1 \le i \le 4$) are all simplest sets of clusters for the topology in Figure 4.

Any one set of clusters in $\mathfrak{C}$ has three clusters denoted as C_1, C_2, and C_3. Let A be the incompatible taxa with respect to C_1 and C_2, and let B be the incompatible taxa with respect to C_2 and C_3; then A and B have the following cases: (i) $A = B$; (ii) $A \subset B$; (iii) $B \subset A$; (iv) $A \cap B = \emptyset$; (v) $A \cap B \neq \emptyset$, $A \not\subseteq B$ and $B \not\subseteq A$.

(i) $A = B$. Since there is no edge between C_1 and C_3, C_1 and C_3 are compatible; that is, $C_1 \cap C_3 = \emptyset$, or $C_1 \subseteq C_3$, or $C_3 \subseteq C_1$. Because $A \subseteq C_1$ and $A \subseteq C_3$, we have that $C_1 \cap C_3 \neq \emptyset$. Therefore, $C_1 \subseteq C_3$ or $C_3 \subseteq C_1$. Then, we have the simplest set of clusters $\mathscr{C}_1 = \{\{1,3\},\{1,2\},\{1,3,4\}\}$, and any one set of clusters in this case is isomorphic to $\mathscr{C}_1$.

(ii) $A \subset B$. Assume that $B = \{A, B_0\}$. It is similar to the case (i), and we have that $C_1 \subseteq C_3$. Then, the simplest set of clusters is $\mathscr{C}_2 = \{\{1,3\},\{1,2,4\},\{1,2,3\}\}$, and any one set of clusters in this case is isomorphic to $\mathscr{C}_1$.

(iii) $B \subset A$. This case is similar to case (ii). The sets of clusters are in case (ii) if and only if they are in case (iii). Hence, any one set of clusters in case (iii) and $\mathscr{C}_2$ are isomorphic.

(iv) $A \cap B = \emptyset$. Then, $C_1 \cap C_3 = \emptyset$. We have that $|A| = 1$ and $|B| = 1$ in the simplest set of clusters, since they can be collapsed if $|A| \ge 2$ or $|B| \ge 2$. Assume that $C_1 = \{A, B_1\}$ and $C_3 = \{B, B_2\}$. We have that $|B_1| = 1$ and $|B_2| = 1$ in the simplest set of clusters, since they can be collapsed if $|B_1| \ge 2$ or $|B_2| \ge 2$. Then, $|C_1| = 2$ and $|C_3| = 2$ in the simplest set of clusters. $\mathscr{C}_3 = \{\{1,2\},\{2,3\},\{3,4\}\}$ and $\mathscr{C}_4 = \{\{1,2\},\{2,3,5\},\{3,4\}\}$ are the simplest sets of clusters in this case. Therefore, any one set of clusters in this case is isomorphic to $\mathscr{C}_3$ or $\mathscr{C}_4$.

(v) $A \cap B \neq \emptyset$, $A \not\subseteq B$ and $B \not\subseteq A$. Let $A = \{A_0, A_1\}$ and $B = \{A_1, B_0\}$, where A_0, A_1, and B_0 are not empty. We have $\{A_0, A_1, B_0\} \subseteq C_2$, and $C_1 \subseteq C_3$ or $C_3 \subseteq C_1$. If $C_1 \subseteq C_3$, then $A_1 \subseteq C_3$. So $A_1 \subseteq B$, which contradicts the case that $A \not\subseteq B$. Similarly, we can get the contradiction when $C_3 \subseteq C_1$. Thus, there exists no set of clusters in this case. □

Figure 5 shows the DC networks for the simplest sets of clusters $\mathscr{C}_1$, $\mathscr{C}_2$, $\mathscr{C}_3$, and $\mathscr{C}_4$, respectively.

Lemma 7. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a nonlinear biconnected component with three nodes*

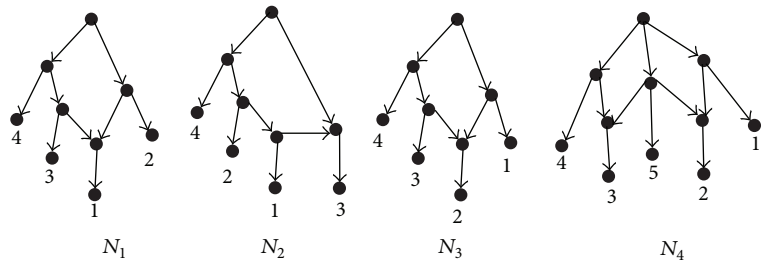


Figure 5: The DC networks for all simplest cluster sets whose incompatible graphs are topologies in Figure 4.

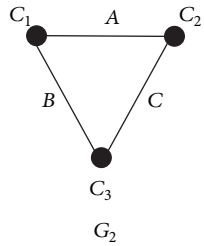


Figure 6: The topology of the nonlinear biconnected component with three nodes.

(see Figure 6). Let $\mathscr{C}_1 = \{\{1,3\},\{1,2,4\},\{1,2,5\}\}$, $\mathscr{C}_2 = \{\{1,2\},\{1,3\},\{1,4\}\}$, $\mathscr{C}_3 = \{\{1,2,4\},\{1,3,5\},\{1,2,3\}\}$, $\mathscr{C}_4 = \{\{1,2,4\},\{1,3,5\},\{1,2,3,6\}\}$, $\mathscr{C}_5 = \{\{1,2\},\{2,3\},\{1,3\}\}$, $\mathscr{C}_6 = \{\{1,2,4\},\{1,3\},\{2,3\}\}$, $\mathscr{C}_7 = \{\{1,2,4\},\{1,3,5\},\{2,3\}\}$, $\mathscr{C}_8 = \{\{1,2,4\},\{1,3,5\},\{2,3,6\}\}$, $\mathscr{C}_9 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\}\}$, $\mathscr{C}_{10} = \{\{1,2,3,5\},\{1,2,4\},\{1,3,4\}\}$, $\mathscr{C}_{11} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4\}\}$, *and* $\mathscr{C}_{12} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4,7\}\}$. *Then, any one set of clusters in* $\mathfrak{C}$ *is isomorphic to one of* $\mathscr{C}_i$ $(1 \le i \le 12)$.

Proof. Figure 6 shows the topology of the nonlinear biconnected component with three nodes. Here, C_1, C_2, and C_3 are the clusters, and A, B, and C are the incompatible taxa corresponding to them. All cases are as follows: (i) $A = B$; then, $A \subseteq C$ or $A = C$; (ii) $A \subset B$; then, $A \subset C$, and $C \cap B = A$; (iii) $A \cap B = \emptyset$; then, $A \cap C = \emptyset$ and $B \cap C = \emptyset$; (iv) $A \cap B \neq \emptyset$, $A \not\subseteq B$, $B \not\subseteq A$; then, $A \cap C \neq \emptyset$ and $B \cap C \neq \emptyset$.

(i) $A = B$. If $A \subseteq C$, then $A \subseteq C_1$, $C \subseteq C_2$, and $C \subseteq C_3$. We have $|A| = 1$ in the simplest set of clusters; otherwise, A can be collapsed into one taxon. Similarly, we have $|C| = 2$ in the simplest set of clusters. Let $A = \{1\}$ and $C = \{1,2\}$; then, we can obtain the only simplest set of clusters $\mathscr{C}_1 = \{\{1,3\},\{1,2,4\},\{1,2,5\}\}$. Any one set of clusters meeting this case will be isomorphic to $\mathscr{C}_1$.

If $A = C$, then $A = B = C$. There is $|A| = 1$ in the simplest set of clusters; otherwise, A can be collapsed into one taxon. Let $A = B = C = \{1\}$; then, we can obtain the only simplest set of clusters $\mathscr{C}_2 = \{\{1,2\},\{1,3\},\{1,4\}\}$. Any one set of clusters in this case will be isomorphic to $\mathscr{C}_2$.

(ii) $A \subset B$, $A \subset C$, *and* $C \cap B = A$. Then, we can obtain the simplest sets of clusters $\mathscr{C}_3 = \{\{1,2,4\},\{1,3,5\},\{1,2,3\}\}$ and $\mathscr{C}_4 = \{\{1,2,4\},\{1,3,5\},\{1,2,3,6\}\}$. Any one set of clusters in this case will be isomorphic to $\mathscr{C}_3$ or $\mathscr{C}_4$.

(iii) $A \cap B = \emptyset$; *then,* $A \cap C = \emptyset$ *and* $B \cap C = \emptyset$. Then, we can obtain the simplest sets of clusters $\mathscr{C}_5 = \{\{1,2\},\{2,3\},\{1,3\}\}$ and $\mathscr{C}_6 = \{\{1,2,4\},\{1,3\},\{2,3\}\}$ and $\mathscr{C}_7 = \{\{1,2,4\},\{1,3,5\},\{2,3\}\}$ and $\mathscr{C}_8 = \{\{1,2,4\},\{1,3,5\},\{2,3,6\}\}$. Any one set of clusters in this case will be isomorphic to one of $\mathscr{C}_5$, $\mathscr{C}_6$, $\mathscr{C}_7$, and $\mathscr{C}_8$.

(iv) $A \cap B \neq \emptyset$, $A \not\subseteq B$, $B \not\subseteq A$; *then,* $A \cap C \neq \emptyset$ *and* $B \cap C \neq \emptyset$. Let $A \cap B = A_0$; then, $A \cap C = A_0$ and $B \cap C = A_0$. We have $|A_0| = 1$ in the simplest set of clusters; otherwise, A_0 can be collapsed into one taxon. Let $A_0 = \{1\}$. Then, $A = \{1,2\}$, $B = \{1,3\}$, and $C = \{1,4\}$. For the first case, we can obtain the simplest sets of clusters $\mathscr{C}_9 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\}\}$ and $\mathscr{C}_{10} = \{\{1,2,3,5\},\{1,2,4\},\{1,3,4\}\}$ and $\mathscr{C}_{11} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4\}\}$ and $\mathscr{C}_{12} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4,7\}\}$. Any one set of clusters in this case will be isomorphic to one of them. □

Figure 7 shows the DC networks for the simplest sets of clusters $\mathscr{C}_i$ $(1 \le i \le 12)$, respectively. Lemmas 5, 6, and 7 compute all simplest sets of clusters, whose incompatible graphs are the biconnected components with two nodes or three nodes. Figures 6 and 7 show the DC networks constructed by the BIMLR algorithm for all simplest sets of clusters; then, the DC network for a set of clusters $\mathscr{C}$ can be obtained from the DC network representing the simplest set of clusters which is isomorphic to $\mathscr{C}$; that is, it does not need to be constructed once again. This conclusion is very important to the construction of networks.

4. Conclusion

This paper computes all simplest sets of clusters for the topologies of incompatible graph with two nodes and three nodes. We can construct the DC networks for those simplest sets of clusters and save them. When constructing DC networks for any one set of clusters $\mathscr{C}$, algorithms only need to read the DC network N_0 of the simplest set of clusters isomorphic to $\mathscr{C}$ and then compute the DC network for $\mathscr{C}$ from N_0 by replacing labels of leaves in N_0 by the taxa in $\mathscr{C}$, which will save more time for the algorithms.

We will compute the simplest sets of clusters for more topologies of incompatible graph in the future.

Competing Interests

The authors declare that they have no competing interests.

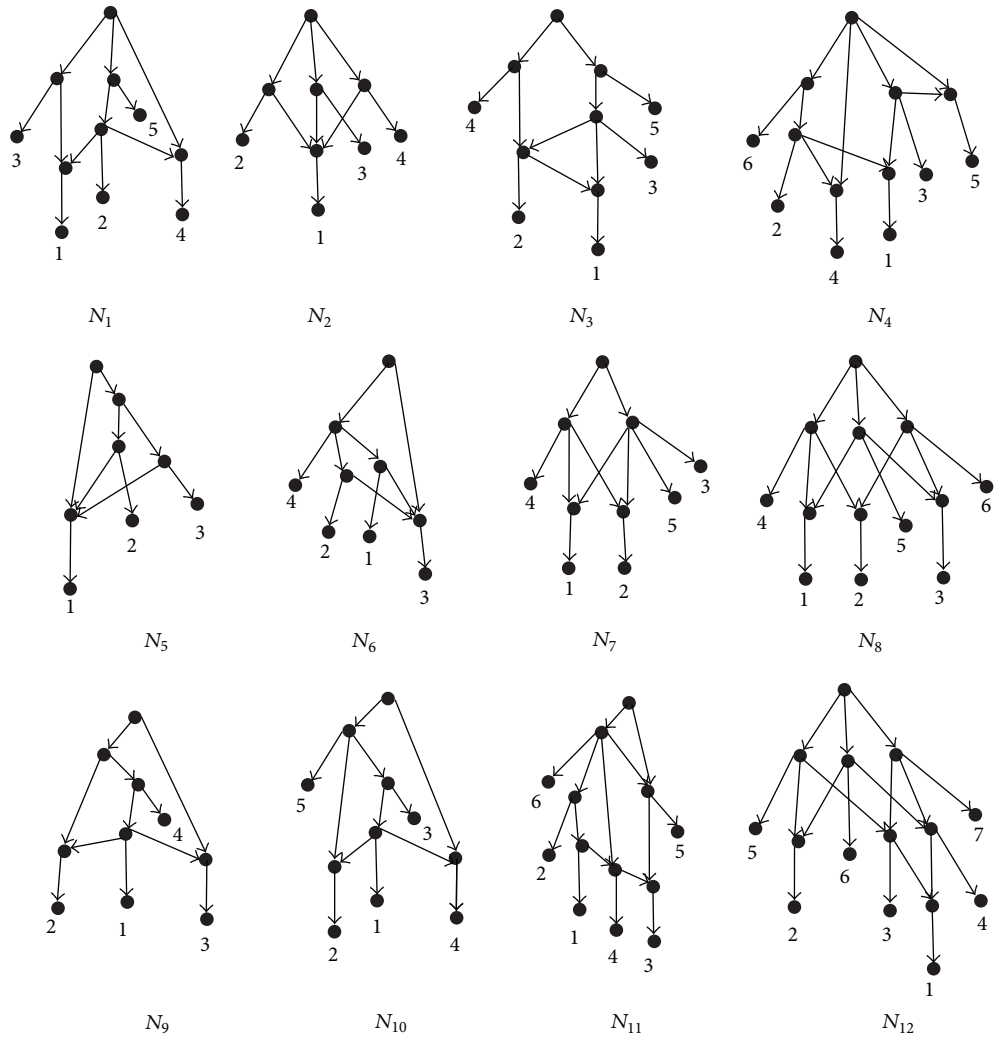


Figure 7: The DC networks for all simplest cluster sets whose incompatible graphs are topologies in Figure 6.

Acknowledgments

The work was supported by the Natural Science Foundation of Inner Mongolia Province of China (2015BS0601) and the National Natural Science Foundation of China (61300098, 31360289).

References

[1] Q. Zou, Q. Hu, M. Guo, and G. Wang, "Halign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[2] D. Mrozek, M. Brozek, and B. Małysiak-Mrozek, "Parallel implementation of 3D protein structure similarity searches using a GPU and the CUDA," *Journal of Molecular Modeling*, vol. 20, no. 2, pp. 1–17, 2014.

[3] D. Gusfield, D. Hickerson, and S. Eddhu, "An efficiently computed lower bound on the number of recombinations in phylogenetic networks: theory and empirical study," *Discrete Applied Mathematics*, vol. 155, no. 6-7, pp. 806–830, 2007.

[4] Q. Zou, X.-B. Li, W.-R. Jiang, Z.-Y. Lin, G.-L. Li, and K. Chen, "Survey of MapReduce frame operation in bioinformatics," *Briefings in Bioinformatics*, vol. 15, no. 4, pp. 637–647, 2014.

[5] Y. Liu, X. Zeng, Z. He, and Q. Zou, "Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[6] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic Networks: Concepts, Algorithms and Applications*, Cambridge University Press, New York, NY, USA, 2011.

[7] Q. Zou, Q. Hu, M. Guo, and G. Wang, "HAlign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[8] J. Wang, M.-Z. Guo, and L. L. Xing, "FastJoin, an improved neighbor-joining algorithm," *Genetics and Molecular Research*, vol. 11, no. 3, pp. 1909–1922, 2012.

[9] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[10] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: a survey," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 55–64, 2016.

[11] D. Bryant and V. Moulton, "Neighbor-net: an agglomerative method for the construction of phylogenetic networks," *Molecular Biology and Evolution*, vol. 21, no. 2, pp. 255–265, 2004.

[12] D. H. Huson, T. Dezulian, T. Klopper, and M. A. Steel, "Phylogenetic super-networks from partial trees," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 4, pp. 151–158, 2004.

[13] D. H. Huson and T. H. Kloepper, "Computing recombination networks from binary sequences," *Bioinformatics*, vol. 21, supplement 2, pp. ii159–ii165, 2005.

[14] L. van Iersel, S. Kelk, R. Rupp, and D. Huson, "Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters," *Bioinformatics*, vol. 26, no. 12, pp. i124–i131, 2010.

[15] Y. Wu, "Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees," *Bioinformatics*, vol. 26, no. 12, Article ID btq198, pp. i140–i148, 2010.

[16] D. H. Huson, R. Rupp, V. Berry, P. Gambette, and C. Paul, "Computing galled networks from real data," *Bioinformatics*, vol. 25, no. 12, pp. i85–i93, 2009.

[17] D. H. Huson and R. Rupp, "Summarizing multiple gene trees using cluster networks," in *Algorithms in Bioinformatics*, K. A. Crandall and J. Lagergren, Eds., vol. 5251, pp. 296–305, Springer, New York, NY, USA, 2008.

[18] L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, F. Hagen, and T. Boekhout, "Constructing level-2 phylogenetic networks from triplets, computational Biology and Bioinformatics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 4, pp. 667–681, 2009.

[19] L. van Iersel and S. Kelk, "Constructing the simplest possible phylogenetic network from triplets," *Algorithmica*, vol. 60, no. 2, pp. 207–235, 2011.

[20] J. Wang, "A new algorithm to construct phylogenetic networks from trees," *Genetics and Molecular Research*, vol. 13, no. 1, pp. 1456–1464, 2014.

[21] D. Mrozek, *High-Performance Computational Solutions in Protein Bioinformatics*, Springer Publishing Company, Incorporated, 2014.

[22] D. Mrozek, P. Gosk, and B. Małysiak-Mrozek, "Scaling Ab initio predictions of 3D protein structures in microsoft azure cloud," *Journal of Grid Computing*, vol. 13, no. 4, pp. 561–585, 2015.

[23] L. Van Iersel, S. Kelk, R. Rupp, and D. Huson, "Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters," *Bioinformatics*, vol. 26, no. 12, Article ID btq202, pp. i124–i131, 2010.

[24] J. Wang, M. Guo, X. Liu et al., "Lnetwork: an efficient and effective method for constructing phylogenetic networks," *Bioinformatics*, vol. 29, no. 18, pp. 2269–2276, 2013.

[25] J. Wang, M. Guo, L. Xing, K. Che, X. Liu, and C. Wang, "BIMLR: a method for constructing rooted phylogenetic networks from rooted phylogenetic trees," *Gene*, vol. 527, no. 1, pp. 344–351, 2013.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 1654623, 8 pages
http://dx.doi.org/10.1155/2016/1654623

Research Article

Identification of Bacterial Cell Wall Lyases via Pseudo Amino Acid Composition

Xin-Xin Chen,[1] Hua Tang,[2] Wen-Chao Li,[1] Hao Wu,[3] Wei Chen,[1,4] Hui Ding,[1] and Hao Lin[1]

[1] *Key Laboratory for Neuro-Information of Ministry of Education, Center of Bioinformatics and Center for Information in Biomedicine, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu 610054, China*
[2] *Department of Pathophysiology, Southwest Medical University, Luzhou 646000, China*
[3] *School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong 518055, China*
[4] *Department of Physics, School of Sciences, and Center for Genomics and Computational Biology, North China University of Science and Technology, Tangshan 063000, China*

Correspondence should be addressed to Wei Chen; greatchen@heuu.edu.cn, Hui Ding; hding@uestc.edu.cn, and Hao Lin; hlin@uestc.edu.cn

Received 24 April 2016; Accepted 30 May 2016

Academic Editor: Qin Ma

Owing to the abuse of antibiotics, drug resistance of pathogenic bacteria becomes more and more serious. Therefore, it is interesting to develop a more reasonable way to solve this issue. Because they can destroy the bacterial cell structure and then kill the infectious bacterium, the bacterial cell wall lyases are suitable candidates of antibacteria sources. Thus, it is urgent to develop an accurate and efficient computational method to predict the lyases. Based on the consideration, in this paper, a set of objective and rigorous data was collected by searching through the Universal Protein Resource (the UniProt database), whereafter a feature selection technique based on the analysis of variance (ANOVA) was used to acquire optimal feature subset. Finally, the support vector machine (SVM) was used to perform prediction. The jackknife cross-validated results showed that the optimal average accuracy of 84.82% was achieved with the sensitivity of 76.47% and the specificity of 93.16%. For the convenience of other scholars, we built a free online server called *Lypred*. We believe that *Lypred* will become a practical tool for the research of cell wall lyases and development of antimicrobial agents.

1. Introduction

Bacteria are widely distributed on the earth, a significant proportion of which can cause disease. The antibiotic can efficiently treat infectious diseases caused by pathogens. However, antibiotics abuse may cause bacterial drug resistance. Thus, there is an ever-increasing need to find new ways to address this important issue [1, 2]. In the search for more effective therapeutic strategies, great effort has been placed on the study and development of lyases, which benefits from high potency activity toward drug-resistant strains and a low inherent susceptibility to emergence of new resistance phenotypes [3–7].

In 1896, the British bacteriologist Hankin found that the bacteriophage has antibacterial activity [3]. Subsequently, in 1921, Brunoghe and Maisin used bacteriophage to treat staphylococcal skin disease in France, which was the first reported application of bacteriophage to treat infectious diseases [8]. Maxted [9], Krause [10], and Fischetti et al. [11] found that the lysates of Group C streptococci infected with C1 bacteriophage contain an enzyme which has the ability to lyse streptococci and their isolated cell walls. The enzyme is called endolysin which is encoded by bacteriophage gene. It can cause bacteria death by degrading cell wall. It has been reported that 10 ng endolysins can lead to 10^7 bacteria's lysis within 30 seconds [4, 12].

Autolysins are another kind of lyases that are functionally similar to endolysins except they are bacteria-encoded enzymes [13]. It has been reported that autolysins play important roles in several fundamental biological phenomena, such as cell wall enlargement, genetic transformation, flagella extrusion, cell division, and lysis induced by fl-lactam antibiotics, as well as in the "suicidal tendencies" of pneumococci [14–16].

Due to their special biological activity, lyases have been applied in antibacteria drug development. Thus, it is necessary to perform intensive research on lyases to understand the antibacterial mechanism. Although wet experiments are an objective approach for accurately recognizing the lyases, they are often time-consuming and costly. Due to the convenience and high efficiency, computational methods have attracted more and more attention. Many algorithms such as common support vector machine (SVM) [17–19], structured SVM [20], artificial neural network (ANN) [21], Random Forest (RF) [22], K-nearest neighbor (KNN) [23–25], Bayesian classifier [26, 27], Mahalanobis discriminant [28, 29], LibD3C [30], genetic algorithm [31], imbalanced classifier [32], learning to rank [33], and ensemble learning [34, 35] have been developed for protein function prediction. Various sequence features descriptors such as amino acid composition [36, 37], pseudo amino acid composition (PseAAC) [38], physicochemical properties [39], secondary structure features [40], and N-peptide composition [41] were proposed to represent protein sequences [42].

To deal with the problem about lyases prediction, recently, a method was developed to identify cell wall enzymes by using PseAAC and Fisher discriminant [43]. A maximum overall accuracy of 80.4% was obtained with the sensitivity of 66.7% and the specificity of 88.6% [43]. However, further work is needed due to the following reasons. (i) The prediction quality can be further improved. (ii) No web server for the prediction method in [43] was provided, and hence its usage is quite limited, especially for the majority of experimental scientists.

The present study was devoted to development of a new predictor for identifying lyases. For this purpose, an objective and strict benchmark dataset was constructed for training and testing the proposed model in which protein sequences were formulated by using an improved PseAAC. For the convenience of other scholars, a free online server called *Lypred* (at http://lin.uestc.edu.cn/server/Lypred/) was established.

2. Material and Method

2.1. Benchmark Dataset. A high quality dataset is the key to building a robust and accurate predictor. The lyases in bacteria or bacteriophage were regarded as positive samples which were derived from the UniProt [44]. Negative samples, namely, the nonlyases, were also derived from bacteriophage and downloaded from the UniProt. In order to guarantee the reliability of the benchmark dataset, we optimized the data according to the following standards: firstly, the sequences whose protein was with annotations of "Inferred from homology" or "Predicted" were excluded; secondly, we removed the sequences which are the fragments of other proteins; thirdly, the protein sequences containing unknown residues, such as "B," "J," "O," "U," "X," and "Z," were eliminated; fourthly, to avoid overestimation of prediction model that resulted from the high sequence identity, the CD-HIT program [45] was adopted to eliminate redundant sequence by setting the cutoff of sequence identity to 40%. As a result, a total of 68 lyases and 307 nonlyases were obtained to form the final benchmark dataset.

2.2. Features Extraction. A sequence can be represented by two different forms: one is the sequential form and the other is the discrete form [46]. The most common and straightforward way to characterize a protein is to use all the residues in its sequence written as follows:

$$P = R_1 R_2 R_3 R_4, \ldots, R_{L-1} R_L, \tag{1}$$

where R_1, R_2, and R_L are the 1st, 2nd, and Lth amino acid residue of protein P, respectively. Based on the information, a query protein can be predicted by the BLAST or FASTA program. The results are always good for the query sequence which has high similar sequences in benchmark dataset; however, it fails to work when the similar sequences for the query sequence are not found in the training dataset [47]. Therefore, the similarity-based method is not suitable for the case that no homologue was found in the benchmark dataset. The discrete form can overcome the shortcoming and is easy to be treated in statistical prediction. Thus, it has been widely used in protein and DNA formulation [48, 49]. The PseAAC is a typical discrete form that has been widely used for protein function prediction [46, 50, 51].

It is well known that the polypeptide chains fold to tertiary structures based on the physicochemical properties of residues. Thus, it is not enough to analyze the residue compositions of protein molecules. Hence, we proposed to represent protein samples by using an improved PseAAC which includes not only g-gap dipeptide composition, but also correlation of physicochemical property between two residues.

According to the concept of PseAAC, a protein P with the length of L can be formulated in a $(400+n\delta)$ dimension space as given by

$$D = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{400} \\ f_{401} \\ \vdots \\ f_{400+n\delta} \end{bmatrix}, \tag{2}$$

where

$$f_i = \begin{cases} \varphi_i, & 1 \le i \le 400 \\ \varepsilon_i, & 400 < x \le 400 + n\delta, \end{cases} \tag{3}$$

where φ_i denotes the normalized occurrence frequency of the ith kind of g-gap dipeptide in protein P formulated as

$$\varphi_i = \frac{n_i^g}{\sum_{i=1}^{400} n_i^g} = \frac{n_i^g}{L-g-1}, \tag{4}$$

where n_i^g $(i = 1, 2, \ldots, 400)$ denotes the number of the ith g-gap dipeptide in P.

ε_i in (3) is the i-tier sequence correlation factor calculated by the following formulas:

$$\begin{aligned}
\varepsilon_{400+1} &= \frac{1}{L-1}\sum_{t=1}^{L-1}\theta_{t,t+1}^1 \\
\varepsilon_{400+2} &= \frac{1}{L-1}\sum_{t=1}^{L-1}\theta_{t,t+1}^2 \\
&\vdots \\
\varepsilon_{400+n} &= \frac{1}{L-1}\sum_{t=1}^{L-1}\theta_{t,t+1}^n \\
\varepsilon_{400+n+1} &= \frac{1}{L-2}\sum_{t=1}^{L-2}\theta_{t,t+2}^1 \\
\varepsilon_{400+n+2} &= \frac{1}{L-2}\sum_{t=1}^{L-2}\theta_{t,t+2}^2 \\
&\vdots \\
\varepsilon_{400+n+n} &= \frac{1}{L-2}\sum_{t=1}^{L-2}\theta_{t,t+2}^n \\
&\vdots \\
\varepsilon_{400+n\delta} &= \frac{1}{L-\delta}\sum_{t=1}^{L-\delta}\theta_{t,t+\delta}^n \\
&\qquad (\delta < L).
\end{aligned} \tag{5}$$

The correlation $\theta_{x,y}^n$ of physicochemical property between two residues is given by

$$\theta_{x,y}^n = \rho^n(R_x)\,\rho^n\left(R_y\right), \tag{6}$$

where $\rho^n(R_x)$ denotes the nth physicochemical value of amino acid residue R_x. The value is obtained by

$$\rho^n(R_x) = \frac{\rho_0^n(R_x) - \sum_{k=1}^{20}\rho_0^n(R_k)/20}{\sqrt{\sum_{t=1}^{20}\left(\rho_0^n(R_t) - \sum_{k=1}^{20}\rho_0^n(R_k)/20\right)^2/20}}, \tag{7}$$

where $\rho_0^n(R_x)$ is the nth physicochemical original value of amino acid R_x.

Thus, each protein sample can be expressed by $400 + n\delta$ kinds of features according to (2)–(7).

2.3. Feature Selection. Some features are noise or redundant information which will reduce the predictive performance of classification models. Thus, it is very important to develop a method to evaluate the contribution of every feature to the classification. Here, we used ANOVA [52] to rank features defined as

$$\begin{aligned}
&F(i) \\
&= \frac{\sum_{j=1}^{2} m_j\left(\sum_{s=1}^{m_j} f_i(s,j)/m_j - \sum_{j=1}^{2}\sum_{s=1}^{m_j} f_i(s,j)/\sum_{j=1}^{2} m_j\right)^2}{\sum_{j=1}^{2}\sum_{s=1}^{m_j}\left(f_i(s,j) - \sum_{s=1}^{m_j} f_i(s,j)/m_j\right)^2/\left(\sum_{j=1}^{2} m_j - 2\right)},
\end{aligned} \tag{8}$$

where $F(i)$ represents the F-score of the ith feature type, $f_i(s, j)$ is the feature value of the ith feature type of the sth sample in the jth protein type, and m_j is the number of samples in the jth protein type. It is obvious that the larger the $F(i)$ value, the better the discriminative capability the ith feature has.

In order to eliminate the redundant features, we firstly ranked all features according to their F-score from high to low. The first feature subset only contained the feature with the largest F-score; then, a new feature subset was generated when the feature with the second largest F-score was added. The process was repeated until all features were added. The SVM was used to evaluate the performance for each feature subset. The feature subset with the best performance is deemed the optimal feature subset which does not contain redundant features.

2.4. Support Vector Machine. The SVM is a linear-classifier-based supervised machine learning method, which has been successfully used in many bioinformatics fields [48–51, 53–57]. To attain the goal of classification, SVM utilizes the kernel function to deal with the nonlinear transformation, and thus linear inseparable can be converted to a linear problem in high-dimension Hilbert space. In this work, the software LIBSVM [58] was used to execute SVM.

2.5. Performance Standard. To provide a more intuitive and easier-to-understand method to evaluate the prediction performance, we used the following criteria: the sensitivity (Sn), the specificity (Sp), Mathew's correlation coefficient (MCC), the overall accuracy (OA), and the average accuracy (AA), which were defined as

$$\begin{aligned}
\mathrm{Sn} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\
\mathrm{Sp} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \\
&\mathrm{MCC} \\
&= \frac{\mathrm{TP}\times\mathrm{TN} - \mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})\times(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})}} \\
\mathrm{OA} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}} \\
\mathrm{AA} &= \frac{1}{2}\times\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right),
\end{aligned} \tag{9}$$

Table 1: The original values of nine physicochemical properties used in this study.

Amino acids	Hydrophobicity	Hydrophilicity	Rigidity	Flexibility	Irreplaceability	Mass	pI	pK(α-COOH)	pK(α-NH_3^+)
A	0.62	−0.5	−1.338	−3.102	0.52	15	6.11	2.35	9.87
C	0.29	−1	−1.511	0.957	1.12	47	5.02	1.71	10.78
D	−0.9	3	−0.204	0.424	0.77	59	2.98	1.88	9.6
E	−0.74	3	−0.365	2.009	0.76	73	3.08	2.19	9.67
F	1.19	−2.5	2.877	−0.466	0.86	91	5.91	2.58	9.24
G	0.48	0	−1.097	−2.746	0.56	1	6.06	2.34	9.6
H	−0.4	−0.5	2.269	−0.223	0.94	82	7.64	1.78	8.97
I	1.38	−1.8	−1.741	0.424	0.65	57	6.04	2.32	9.76
K	−1.5	3	−1.822	3.950	0.81	73	9.47	2.2	8.9
L	1.06	−1.8	−1.741	0.424	0.58	57	6.04	2.36	9.6
M	0.64	−1.3	−1.741	2.484	1.25	75	5.74	2.28	9.21
N	−0.78	0.2	−0.204	0.424	0.79	58	10.76	2.18	9.09
P	0.12	0	1.979	−2.404	0.61	42	6.3	1.99	10.6
Q	−0.85	0.2	−0.365	2.009	0.86	72	5.65	2.17	9.13
R	−2.53	3	1.169	3.060	0.60	101	10.76	2.18	9.09
S	−0.18	0.3	−1.511	0.957	0.64	31	5.68	2.21	9.15
T	−0.05	−0.4	−1.641	−1.339	0.56	45	5.6	2.15	9.12
V	1.08	−1.5	−1.641	−1.339	0.54	43	6.02	2.29	9.74
W	0.81	−3.4	5.913	−1.000	1.82	130	5.88	2.38	9.39
Y	0.26	−2.3	2.714	−0.672	0.98	107	5.63	2.2	9.11

where TP is the number of lyases that were correctly predicted, FN denotes the number of lyases that were predicted as the nonlyases, TN is the number of nonlyases that were correctly predicted, and FP denotes the number of nonlyases that were predicted as the lyases.

In addition, we also chose the receiver operating characteristic curve (ROC curve) to measure the performance of the proposed model. ROC curve is a kind of comprehensive index that is drawn by using (1 − Sp) as the abscissa and Sn as the ordinate. Thus, it reveals the continuous variable of Sn and Sp. Generally, we only need to calculate the area under the ROC curve (auROC). The greater the auROC is, the better the discriminate capability the prediction model has is.

3. Results and Discussion

3.1. Forecasting Accuracy. In this work, 9 kinds of physicochemical properties were selected in improved PseAAC [47]. The nine physicochemical properties are hydrophobicity, hydrophilicity, rigidity, flexibility, irreplaceability, side chain mass, pI at 25°C, pK of the α-COOH group, and pK of the α-NH_3^+ group [47], respectively. The original values of the physicochemical properties for 20 amino acids were all listed in Table 1. According to (2)–(7), each protein sample can be formulated by a $(400 + 9\delta)$ dimension vector including 400 g-gap dipeptide compositions and 9δ correlation factors based on physicochemical properties between two residues. From (3)–(5), the prediction performance of our method was influenced by two parameters, namely, g and δ, where g describes the local sequence-order effect and δ represents the global sequence-order effect. The current study searched for the optimal values for the two parameters according to the following standard:

$$\begin{aligned} &0 \le g \le 9, \quad \text{with step } \Delta = 1 \\ &1 \le \delta \le 10, \quad \text{with step } \Delta = 1. \end{aligned} \tag{10}$$

In cross-validation test, n-fold cross-validation, jackknife cross-validation, and independent dataset test are often used for measuring the performance of prediction model. Although the jackknife cross-validation is deemed the most objective because it can always yield a unique result for benchmark dataset given [59, 60] and it has been more and more widely used, it also has obvious drawbacks, such as the large calculation and being time-consuming. Hence, the 5-fold cross-validation was adopted in this work for searching the optimal parameters and the optimal feature subset. Once the optimal feature subset was determined, we used jackknife cross-validation for verification ulteriorly.

Based on (10), a total of $10 \times 10 = 100$ groups of parameters (g, δ) were investigated. For each parameter group (g, δ), there are $400 + 9\delta$ feature subsets. Then, we used feature selection technique defined in (8) to find out the best one in each parameter group. Thus, we obtained the 100 highest OAs for 100 groups of parameters (g, δ). To provide an overall and intuitive analysis, the best OAs were drawn into a heat map, where the column and row of the heat map represent the parameters g and δ, respectively. Each element in the heat map represents one of the 100 groups of parameters (g, δ) and was colorized according to its highest overall accuracy in feature selection process. From Figure 1, we noticed that several elements are red indicating the maximum overall

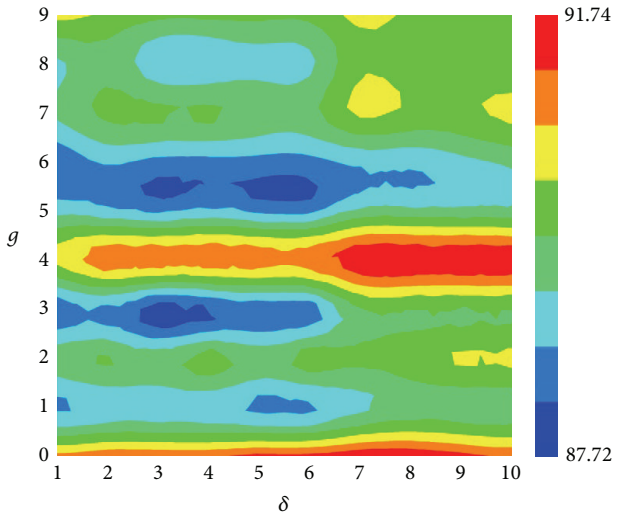


Figure 1: A heat map to show the overall accuracy in 5-fold cross-validation with different parameter groups (g, δ).

accuracy of 91.73% when g equals 0 or 4 and δ equals 7, 8, 9, and 10. Generally, a model with a small number of features can reduce the risk of overfitting. After checking the feature selected results, we found that when using feature selection technique to optimize parameter group ($g = 4$ and $\delta = 7$), the optimal feature subset contains 63 features, which is less than the optimal feature subset in other groups. Thus, the final model was established based on the 63 features from parameter group ($g = 4$ and $\delta = 7$).

Because there is imbalance in our benchmark dataset, the average accuracy and ROC curve were employed to evaluate the model. Thus, we set a series of different classification thresholds to seek the maximum of average accuracy. The maximum AA and corresponding Sn, Sp, MCC, and OA were listed in Table 2. The ROC curve can demonstrate the predictive capability of the proposed method across the entire range of SVM decision values. Thus, we plotted the ROC curve in Figure 2. It shows that auROC is 0.926, demonstrating that our model has capability to predict cell wall lyases.

To investigate whether other algorithms have the same or higher discriminate capability in the same feature space, the performances of Random Forest, Naïve Bayes, and LibD3C were examined by using jackknife cross-validation. Random Forest and Naïve Bayes were executed by using free package WEKA [61]. The LibD3C, a new selective ensemble algorithm, is a hybrid model of ensemble pruning that is based on k-means clustering and the framework of dynamic selection and circulating in combination with a sequential search method [30]. We used default parameters in LibD3C to perform classification.

The jackknife cross-validated results were also recorded in Table 2 for clear comparison. Note that the result for each algorithm in Table 2 was calculated with the maximum AA. As can be seen from the table, although Sn's of Random Forest and Naïve Bayes are no lower than SVM, other indicators (Sp, MCC, OA, AA, and auROC) of SVM are the best.

Table 2: Comparison among the performances of different algorithms.

Algorithm	Sn (%)	Sp (%)	MCC	OA (%)	AA (%)	auROC
SVM	76.47	93.16	0.678	90.13	84.82	0.926
Random Forest	80.88	85.02	0.572	84.27	82.95	0.905
Naïve Bayes	76.47	83.06	0.512	81.87	79.77	0.881
LibD3C	66.18	88.60	0.515	84.53	77.39	0.859

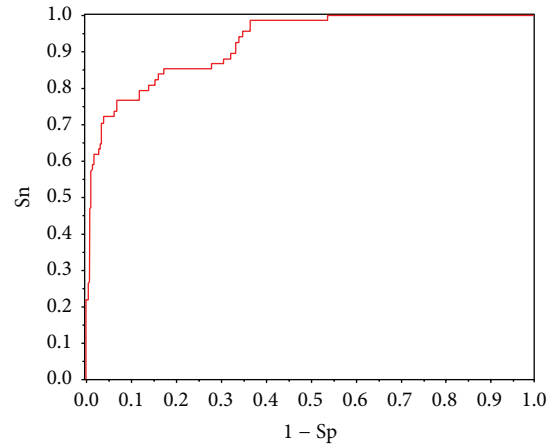


Figure 2: The ROC curve for the proposed model with the 63 optimal features in jackknife cross-validation using SVM.

3.2. Online-Server Guide. A user-friendly online server called *Lypred* was established. A simple guide about the server was given below in order to further make it easier for the users.

Lypred has five pages. Users can browse the server at http://lin.uestc.edu.cn/server/Lypred/ and see the home page on the screen as shown in Figure 3. The Read Me page provides a brief introduction about *Lypred* and the caveat when being used. The Data page shows a brief description about the benchmark dataset and the optimal feature subset used in this work and provides links for downloading. The relevant paper about the detailed development and algorithm of *Lypred* can be seen by clicking the Citation button. Example sequences in FASTA format can be found by clicking the Example button right above the input box.

Users can not only type or copy/paste the query protein sequences into the input box, but also upload FASTA/txt file containing the query protein sequences at the center of the home page of *Lypred*. Note that *Lypred* also has some constraints so as to guarantee the reliability of the results: firstly, protein sequences must be in FASTA format consisting of a single initial line beginning with a greater-than symbol (">") in the first column, followed by lines of sequence data, and the sequence is deemed to end if there is another line starting with ">"; secondly, the query protein sequence should only contain 20 kinds of amino acids; thirdly, the length of a query protein sequence should be no less than eight.

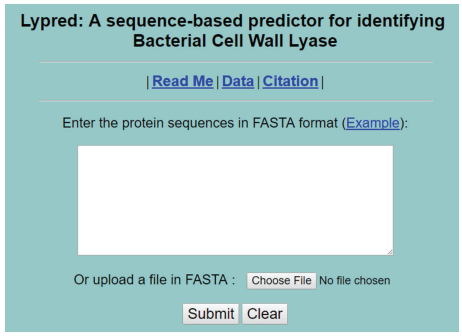


Figure 3: A semiscreenshot to show the home page of Lypred. Its website address is http://lin.uestc.edu.cn/server/Lypred/.

4. Conclusions

With growing drug resistance of pathogenic bacteria, great effort has been placed on the study and development of lyases. Effective identification of lyases will provide convenience for development of new antimicrobials. In this work, we used an improved PseAAC including g-gap dipeptide compositions and correlation factors of the physicochemical properties to extract the characteristics of protein sequences. A feature selection technique based on ANOVA was used to optimize features. The results of AA of 84.82% and auROC of 0.926 make us believe that *Lypred* will become a powerful and useful tool for the experimental study of bacterial cell wall lyase.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Applied Basic Research Program of Sichuan Province (nos. 2015JY0100 and LZ-LY-45), the Scientific Research Foundation of the Education Department of Sichuan Province (11ZB122), the Nature Scientific Foundation of Hebei Province (no. C2013209105), the Fundamental Research Funds for the Central Universities of China (nos. ZYGX2015J144 and ZYGX2015Z006), and the Program for the Top Young Innovative Talents of Higher Learning Institutions of Hebei Province (no. BJ2014028).

References

[1] D. Trudil, "Phage lytic enzymes: a history," *Virologica Sinica*, vol. 30, no. 1, pp. 26–32, 2015.

[2] Y. Li, C. Wang, Z. Miao et al., "ViRBase: a resource for virus–host ncRNA-associated interactions," *Nucleic Acids Research*, vol. 43, no. 1, pp. D578–D582, 2015.

[3] E. Hankin, "L'action bactericide des eaux de la Jumna et du Gange sur le vibrion du cholera," *Annales de l'Institut Pasteur*, vol. 10, pp. 511–523, 1896.

[4] V. A. Fischetti, "Bacteriophage lytic enzymes: novel anti-infectives," *Trends in Microbiology*, vol. 13, no. 10, pp. 491–496, 2005.

[5] D. C. Osipovitch, S. Therrien, and K. E. Griswold, "Discovery of novel *S. aureus* autolysins and molecular engineering to enhance bacteriolytic activity," *Applied Microbiology and Biotechnology*, vol. 99, no. 15, pp. 6315–6326, 2015.

[6] C. C. Kietzman, G. Gao, B. Mann, L. Myers, and E. I. Tuomanen, "Dynamic capsule restructuring by the main pneumococcal autolysin LytA in response to the epithelium," *Nature Communications*, vol. 7, article 10859, 2016.

[7] H. Oliveir, L. D. R. Melo, S. B. Santos et al., "Molecular aspects and comparative genomics of bacteriophage endolysins," *Journal of Virology*, vol. 87, no. 8, pp. 4558–4570, 2013.

[8] R. Brunoghe and J. Maisin, "Essais de therapeutique au moyen du bacteriophage du staphylocoque," *Journal des Comptes Rendus de la Société de Biologie*, vol. 85, pp. 1029–1121, 1921.

[9] W. R. Maxted, "The active agent in nascent phage lysis of streptococci," *Microbiology*, vol. 16, no. 3, pp. 584–595, 1957.

[10] R. M. Krause, "Studies on the bacteriophages of hemolytic streptococci. II. Antigens released from the streptococcal cell wall by a phage-associated lysin," *The Journal of Experimental Medicine*, vol. 108, no. 6, pp. 803–821, 1958.

[11] V. A. Fischetti, E. C. Gotschlich, and A. W. Bernheimer, "Purification and physical properties of group C streptococcal phage-associated lysin," *The Journal of Experimental Medicine*, vol. 133, no. 5, pp. 1105–1117, 1971.

[12] R. Schuch, D. Nelson, and V. A. Fischetti, "A bacteriolytic agent that detects and kills Bacillus anthracis," *Nature*, vol. 418, no. 6900, pp. 884–889, 2002.

[13] O. Salazar and J. A. Asenjo, "Enzymatic lysis of microbial cells," *Biotechnology Letters*, vol. 29, no. 7, pp. 985–994, 2007.

[14] H. J. Rogers, H. R. Perkins, and J. B. Ward, *Microbial Cell Walls and Membranes*, Chapman and Hall London, 1980.

[15] M. McCarty, *The Transforming Principle: Discovering That Genes Are Made of DNA*, W. W. Norton & Company, New York, NY, USA, 1986.

[16] J. M. Sanchez-Puelles, C. Ronda, J. L. Garcia, P. Garcia, R. Lopez, and E. Garcia, "Searching for autolysin functions. Characterization of a pneumococcal mutant deleted in the lytA gene," *European Journal of Biochemistry*, vol. 158, no. 2, pp. 289–293, 1986.

[17] K.-C. Chou and H.-B. Shen, "Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms," *Nature Protocols*, vol. 3, no. 2, pp. 153–162, 2008.

[18] M. K. Leong and T.-H. Chen, "Prediction of cytochrome P450 2B6-substrate interactions using pharmacophore ensemble/support vector machine (PhE/SVM) approach," *Medicinal Chemistry*, vol. 4, no. 4, pp. 396–406, 2008.

[19] B. Liu, D. Zhang, R. Xu et al., "Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection," *Bioinformatics*, vol. 30, no. 4, pp. 472–479, 2014.

[20] D. Li, Y. Ju, and Q. Zou, "Protein folds prediction with hierarchical structured SVM," *Current Proteomics*, vol. 13, no. 2, pp. 79–85, 2016.

[21] A. Reinhardt and T. Hubbard, "Using neural networks for prediction of the subcellular location of proteins," *Nucleic Acids Research*, vol. 26, no. 9, pp. 2230–2236, 1998.

[22] X. Zhao, Q. Zou, B. Liu, and X. Liu, "Exploratory predicting protein folding model with random forest and hybrid features," *Current Proteomics*, vol. 11, no. 4, pp. 289–299, 2014.

[23] H. Shen and K.-C. Chou, "Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types," *Biochemical and Biophysical Research Communications*, vol. 334, no. 1, pp. 288–292, 2005.

[24] C. Yan, J. Hu, and Y. Wang, "Discrimination of outer membrane proteins using a K-nearest neighbor method," *Amino Acids*, vol. 35, no. 1, pp. 65–73, 2008.

[25] T.-L. Zhang, Y.-S. Ding, and K.-C. Chou, "Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern," *Journal of Theoretical Biology*, vol. 250, no. 1, pp. 186–193, 2008.

[26] A. Bulashevska, M. Stein, D. Jackson, and R. Eils, "Prediction of small molecule binding property of protein domains with Bayesian classifiers based on Markov chains," *Computational Biology and Chemistry*, vol. 33, no. 6, pp. 457–460, 2009.

[27] A. Bulashevska and R. Eils, "Using Bayesian multinomial classifier to predict whether a given protein sequence is intrinsically disordered," *Journal of Theoretical Biology*, vol. 254, no. 4, pp. 799–803, 2008.

[28] H. Lin and Q.-Z. Li, "Using pseudo amino acid composition to predict protein structural class: approached by incorporating 400 dipeptide components," *Journal of Computational Chemistry*, vol. 28, no. 9, pp. 1463–1466, 2007.

[29] H. Lin, "The modified Mahalanobis discriminant for predicting outer membrane proteins by using Chou's pseudo amino acid composition," *Journal of Theoretical Biology*, vol. 252, no. 2, pp. 350–356, 2008.

[30] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, 2014.

[31] X. Zeng, S. Yuan, X. Huang, and Q. Zou, "Identification of cytokine via an improved genetic algorithm," *Frontiers of Computer Science*, vol. 9, no. 4, pp. 643–651, 2015.

[32] L. Song, D. Li, X. Zeng, Y. Wu, L. Guo, and Q. Zou, "nDNA-prot: identification of DNA-binding proteins based on unbalanced classification," *BMC Bioinformatics*, vol. 15, article 298, 2014.

[33] B. Liu, J. Chen, and X. Wang, "Application of learning to rank to protein remote homology detection," *Bioinformatics*, vol. 31, no. 21, pp. 3492–3498, 2015.

[34] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, pp. 1–15, Springer, 2000.

[35] T. G. Dietterich, "Ensemble learning," in *The Handbook of Brain Theory and Neural Networks*, vol. 2, pp. 110–125, MIT Press, 2002.

[36] M. H. Smith, "The amino acid composition of proteins," *Journal of Theoretical Biology*, vol. 13, pp. 261–282, 1966.

[37] J. Cedano, P. Aloy, J. A. Perez-Pons, and E. Querol, "Relation between amino acid composition and cellular location of proteins," *Journal of Molecular Biology*, vol. 266, no. 3, pp. 594–600, 1997.

[38] K.-C. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, 2001.

[39] S. Saha and G. P. S. Raghava, "BcePred: prediction of continuous B-cell epitopes in antigenic sequences using physico-chemical properties," in *Artificial Immune Systems*, G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, Eds., vol. 3239 of *Lecture Notes in Computer Science*, pp. 197–204, Springer, New York, NY, USA, 2004.

[40] L. Wei, M. Liao, X. Gao, and Q. Zou, "An improved protein structural classes prediction method by incorporating both sequence and structure information," *IEEE Transactions on NanoBioscience*, vol. 14, no. 4, pp. 339–349, 2015.

[41] C.-S. Yu, C.-J. Lin, and J.-K. Hwang, "Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions," *Protein Science*, vol. 13, no. 5, pp. 1402–1406, 2004.

[42] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, and K. Chou, "Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Nucleic Acids Research*, vol. 43, no. W1, pp. W65–W71, 2015.

[43] H. Ding, L. Luo, and H. Lin, "Prediction of cell wall lytic enzymes using Chou's amphiphilic pseudo amino acid composition," *Protein and Peptide Letters*, vol. 16, no. 4, pp. 351–355, 2009.

[44] A. M. Bairoch, R. Apweiler, C. H. Wu et al., "The universal protein resource (UniProt)," *Nucleic Acids Research*, vol. 33, pp. D154–D159, 2005.

[45] W. Li and A. Godzik, "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[46] K.-C. Chou, "Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes," *Bioinformatics*, vol. 21, no. 1, pp. 10–19, 2005.

[47] H. Tang, W. Chen, and H. Lin, "Identification of immunoglobulins using Chou's pseudo amino acid composition with feature selection technique," *Molecular BioSystems*, vol. 12, no. 4, pp. 1269–1275, 2016.

[48] P.-P. Zhu, W.-C. Li, Z.-J. Zhong et al., "Predicting the subcellular localization of mycobacterial proteins by incorporating the optimal tripeptides into the general form of pseudo amino acid composition," *Molecular BioSystems*, vol. 11, no. 2, pp. 558–563, 2015.

[49] W.-C. Li, E.-Z. Deng, H. Ding, W. Chen, and H. Lin, "IORI-PseKNC: a predictor for identifying origin of replication with pseudo k-tuple nucleotide composition," *Chemometrics and Intelligent Laboratory Systems*, vol. 141, pp. 100–106, 2015.

[50] M. Esmaeili, H. Mohabatkar, and S. Mohsenzadeh, "Using the concept of Chou's pseudo amino acid composition for risk type prediction of human papillomaviruses," *Journal of Theoretical Biology*, vol. 263, no. 2, pp. 203–209, 2010.

[51] C. Chen, X. Zhou, Y. Tian, X. Zou, and P. Cai, "Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network," *Analytical Biochemistry*, vol. 357, no. 1, pp. 116–121, 2006.

[52] M. J. Anderson, "A new method for non-parametric multivariate analysis of variance," *Austral Ecology*, vol. 26, no. 1, pp. 32–46, 2001.

[53] R. Wang, Y. Xu, and B. Liu, "Recombination spot identification Based on gapped k-mers," *Scientific Reports*, vol. 6, article 23934, 2016.

[54] J. Chen, X. Wang, and B. Liu, "IMiRNA-SSF: improving the identification of microrna precursors by combining negative sets with different distributions," *Scientific Reports*, vol. 6, article 19062, 2016.

[55] P. Feng, H. Lin, W. Chen, and Y. Zuo, "Predicting the types of J-proteins using clustered amino acids," *BioMed Research International*, vol. 2014, Article ID 935719, 8 pages, 2014.

[56] W. Chen and H. Lin, "Identification of voltage-gated potassium channel subfamilies from sequence information using support vector machine," *Computers in Biology and Medicine*, vol. 42, no. 4, pp. 504–507, 2012.

[57] W. Chen and H. Lin, "Prediction of midbody, centrosome and kinetochore proteins based on gene ontology information," *Biochemical and Biophysical Research Communications*, vol. 401, no. 3, pp. 382–384, 2010.

[58] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[59] K.-C. Chou and C.-T. Zhang, "Prediction of protein structural classes," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 30, no. 4, pp. 275–349, 1995.

[60] Y.-C. Wang, X.-B. Wang, Z.-X. Yang, and N.-Y. Deng, "Prediction of enzyme subfamily class via pseudo amino acid composition by incorporating the conjoint triad feature," *Protein and Peptide Letters*, vol. 17, no. 11, pp. 1441–1449, 2010.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 7534258, 9 pages
http://dx.doi.org/10.1155/2016/7534258

Research Article

A Metric on the Space of Partly Reduced Phylogenetic Networks

Juan Wang

School of Computer Science, Inner Mongolia University, Hohhot 010021, China

Correspondence should be addressed to Juan Wang; wangjuanangle@hit.edu.cn

Received 30 March 2016; Accepted 23 May 2016

Academic Editor: Dariusz Mrozek

Phylogenetic networks are a generalization of phylogenetic trees that allow for the representation of evolutionary events acting at the population level, such as recombination between genes, hybridization between lineages, and horizontal gene transfer. The researchers have designed several measures for computing the dissimilarity between two phylogenetic networks, and each measure has been proven to be a metric on a special kind of phylogenetic networks. However, none of the existing measures is a metric on the space of partly reduced phylogenetic networks. In this paper, we provide a metric, d_e-distance, on the space of partly reduced phylogenetic networks, which is polynomial-time computable.

1. Introduction

Phylogenies reveal the history of evolutionary events of a group of species, and they are central to comparative analysis methods for testing hypotheses in evolutionary biology [1]. Computing the distance between a pair of phylogenies is very important for understanding the evolutionary history of species.

A metric d on a space S satisfies four properties for all $a, b, c \in S$:

(I) $d(a, b) \geq 0$ (nonnegative property);

(II) $d(a, b) = 0$ if and only if $a = b$ (separation property);

(III) $d(a, b) = d(b, a)$ (symmetry property);

(IV) $d(a, b) + d(b, c) \geq d(a, c)$ (triangle inequality).

Phylogenetic network can represent reticulate evolutionary events, such as recombinations between genes, hybridization between lineages, and horizontal gene transfer [2–5]. For the comparison of phylogenetic networks, there are many metrics on the restricted subclasses of networks including the tripartition metric on the space of tree-child phylogenetic networks [6–9], the μ-distance on the space of tree-sibling phylogenetic networks [10], and the m-distance on the space of reduced phylogenetic networks [11]. Later the m-distance was also proved to be a metric on the space of tree-child phylogenetic networks, semibinary tree-sibling time consistent phylogenetic networks, and multilabeled phylogenetic trees [12–15].

For any rooted phylogenetic network N, we can obtain its reduced version by removing all nodes in maximal convergent sets (will be discussed in the following) and all the nodes, with indegree 1 and outdegree 1, from N. The reduced versions of all rooted phylogenetic networks form the space of reduced phylogenetic networks (m-distance, defined by Nakhleh, is on this space). In this paper, we will discuss the partly reduced version of a phylogenetic network by removing the nodes in a part of the convergent sets and all the nodes, with indegree 1 and outdegree 1, from the phylogenetic network. The partly reduced versions of all rooted phylogenetic networks form the space of partly reduced phylogenetic networks. Then we will introduce a novel metric on the space of partly reduced phylogenetic networks. The space is not the space of rooted phylogenetic networks, but it is the largest space on which a polynomial-time computable metric has been defined so for. The papers [16, 17] have proved that the isomorphism for rooted phylogenetic networks is graph isomorphism-complete. Unless the graph isomorphism problem belongs to P, there is no hope of defining a polynomial-time computable metric on the space of all rooted phylogenetic networks. However, our paper's aim is mainly to find a larger space on which a polynomial-time computable metric can be defined such that the space is closer to the space of rooted phylogenetic networks.

2. Preliminaries

Let $N = (V, E)$ be a directed acyclic graph, or DAG for short. We denote the indegree of a node u as indeg(u) and the outdegree of u as outdeg(u). We will say that a node u is a *tree node* if indeg(u) ≤ 1. Particularly, u is a *root* of N if indeg(u) $= 0$ of N. If a single root exists, we will say that the DAG is *rooted*. We will say that a node u is a *reticulate node* if indeg(u) ≥ 2. A tree node u is a *leaf* if outdeg(u) $= 0$. A node is called an *internal node* if its outdeg ≥ 1. For a DAG $N = (V, E)$, we will say that v is a *child* of u if $(u, v) \in E$; in this case, we will also say that u is a *parent* of v. Note that any tree node has a single parent, except for the root of the graph. Whenever there is a directed path from a node u to v, we will say that v is a *descendant* of u or u is an *ancestor* of v.

The *height* of a node is the length of a longest path starting at the node and ending in a leaf. The absence of cycles implies that the nodes of a DAG N can be stratified by means of their heights: the nodes of height 0 are the leaves; if a node has height $a > 0$, then all its children have heights that are smaller than a and at least one of them has height exactly $a - 1$.

The *depth* of a node is the length of a longest path starting at the root and ending in the node. Similarly, the absence of cycles implies that the nodes of a DAG N can also be stratified according to their depths: the node of depth 0 is the root; if a node has depth $b > 0$, then all its parents have depths that are smaller than b and at least one of them has depth exactly $b - 1$.

Let $\mathcal{X}$ be a set of taxa. A rooted phylogenetic network N on $\mathcal{X}$ is a rooted DAG such that

(i) no tree node has outdeg 1;

(ii) its leaves are labeled by $\mathcal{X}$ by a bijective mapping f.

We use the notation $N = ((V, E), f)$ (or $N = (V, E)$) for the rooted phylogenetic network N and the notation V_N for its leaf set.

Definition 1. Two rooted phylogenetic networks $N_1 = ((V_1, E_1), f_1)$ and $N_2 = ((V_2, E_2), f_2)$ are isomorphic if and only if there is a bijection G from V_1 to V_2 such that

(i) (u, v) is an edge in E_1 if and only if $(G(u), G(v))$ is an edge in E_2;

(ii) $f_1(w) = f_2(G(w))$ for all $w \in V_{N_1}$.

Moret et al. (2004) discussed the concept of reduced phylogenetic networks from a reconstruction standpoint. Subsequently, we briefly review the concept of reduced phylogenetic networks and introduce a new definition of partly reduced phylogenetic networks. In the following section, we present a metric on the space of all partly reduced phylogenetic networks. First we review the concept of a maximal convergent set that has been given in [7, 11].

Definition 2. Given a network $N = (V, E)$, we say that a set U of internal nodes in V is convergent if $|U| \geq 2$ and

every leaf reachable from some node in U is reachable from all nodes in U.

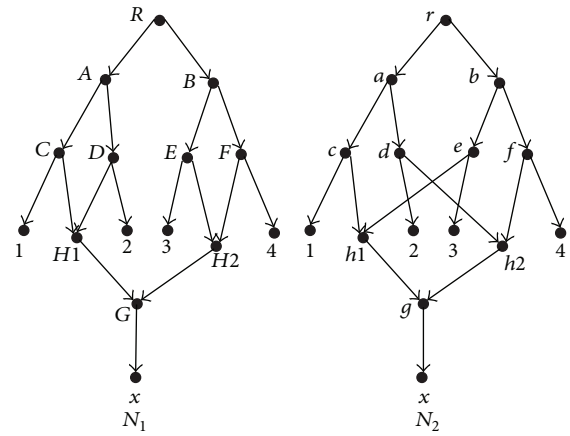


FIGURE 1: Networks N_1 and N_2 from refinements (1) and (2) in Table 1 in [11]. $H1$ and $H2$ (resp., $h1$ and $h2$) are the reticulate nodes, A~G (resp., a~g); $H1$ and $H2$ (resp., $h1$ and $h2$) as well as the root R (resp., r) are the internal tree nodes in network N_1 (resp., N_2).

If there is no convergent set U_0 containing U except U itself, we say that U is a maximal convergent set.

Here the leaf set reachable from the nodes in a convergent set U is called the leaf set of U.

We will take Figure 1 as an example in the following. The two networks N_1, N_2 on $\{1, 2, 3, 4, x\}$ are adapted from refinements (1) and (2) in Table 1 in [11].

Example 3. Consider the networks in Figure 1. The set $\{H1, H2, G\}$ is the only maximal convergent set of N_1 and the set $\{h1, h2, g\}$ is the only maximal convergent set of N_2.

For a phylogenetic network $N = ((V, E), f)$ on $\mathcal{X}$, the reduced version of N can be obtained by the following reduction procedures:

(1) For each maximal pendant subtree (i.e., the maximal clade that includes no reticulate nodes) t, rooted at node r_t, create a new node h_t and an edge (p_t, h_t), where p_t is the parent of r_t, delete the edge (p_t, r_t) and the subtree t, and label h_t as t. Then we denote the resulting network as N_0.

(2) Repeat the following two steps on N_0 until no change occurs:

(I) For each maximal convergent set U with leaf set $L_U \subseteq V_{N_0}$, remove all nodes and edges on the paths from a node in U to the parent of leaf in L_U, including all nodes in U and excluding the parent of leaf in L_U. For each edge (p, v), where p lies outside the deleted set and v lies inside the deleted set, replace it with a set of edges $\{(p, q)$: q is the parent of leaf in $L_U\}$.

(II) For each node w in the network, with indeg(w) = outdeg(w) = 1, remove the edges (u, w), (w, v) and the node w, add an edge (u, v), where u is the parent of w and v is the child of w. Repeat this step until no such node can be removed.

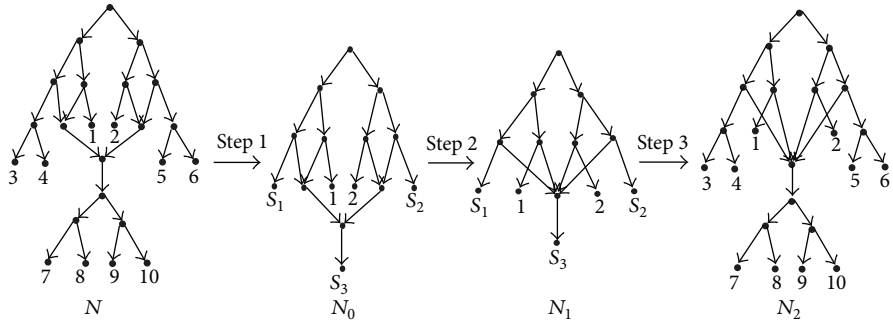


FIGURE 2: The rooted phylogenetic network N is on $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. N_0, N_1, and N_2 are the networks obtained by applying each one of the three reduction procedures to N, respectively.

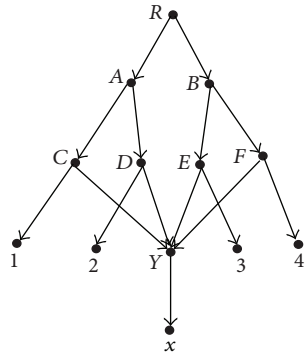


FIGURE 3: The reduced version of the networks in Figure 1.

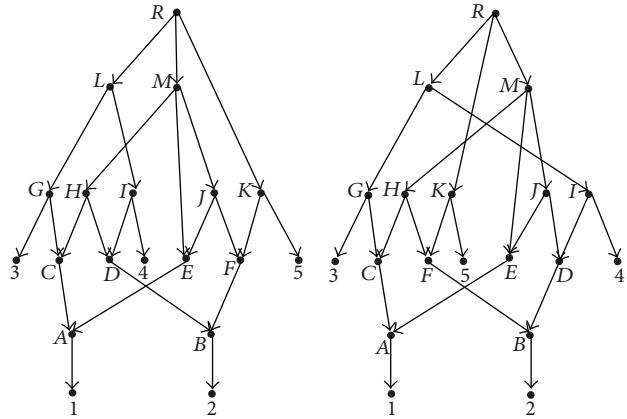


FIGURE 4: Networks N_1 and N_2 are not isomorphic.

(3) Replace each leaf labeled by the subtree t by its root r_t.

Figure 2 shows the results of applying the reduction procedures to the network N. For the networks in Figure 1, their reduced versions are the same (see Figure 3). The reduced versions of all rooted phylogenetic networks form the space of reduced phylogenetic networks. Nakhleh has introduced a polynomial-time computable metric on this space [11]. In order to enlarge the space in which a polynomial-time computable metric can be defined, we will introduce a new metric and a new space that contains the space of reduced phylogenetic networks.

Definition 4. Given a network $N = (V, E)$, let $\mathcal{P}(v)$ be the set of parents of a node v in V. We say that $U \subset V$ is a super convergent set, if

(i) U is a convergent set;

(ii) $\mathcal{P}(u_1) = \mathcal{P}(u_2)$ for any two nodes $u_1, u_2 \in U$;

(iii) $\mathcal{P}(u)$ is a convergent set for a node $u \in U$, if $|\mathcal{P}(u)| \geq 2$.

Example 5. The set $\{H, J\}$ is the only superconvergent set for any one network in Figure 4, while the networks in Figure 1 have no superconvergent set.

We will obtain the new reduction procedures, called partial reduction procedures, from the above reduction procedures by just processing superconvergent sets rather than maximal convergent sets in step (I) of step (2). After applying the partial reduction procedures to a rooted phylogenetic network N, the partly reduced version of N is obtained. The partly reduced versions of all rooted phylogenetic networks form the space of partly reduced phylogenetic networks. This space contains the space of reduced phylogenetic networks, but they are not identical. Next we will introduce a polynomial-time computable metric for the partly reduced phylogenetic networks.

We begin with the notion of node semiequivalence. For the sake of simplicity, we will hereafter refer to the rooted phylogenetic networks as the networks.

3. A Metric

Definition 6. Given a network $N = ((V, E), f)$, we say that two nodes $u, v \in V$ (not necessarily different) are semiequivalent, denoted by $u \triangleq v$, if

(i) $u, v \in V_N$ and $f(u) = f(v)$ or

(ii) node u has k (≥ 1) children $u_1, u_2, \ldots, u_k$; node v has k children $v_1, v_2, \ldots, v_k$, and $u_i \triangleq v_i$ for $1 \leq i \leq k$.

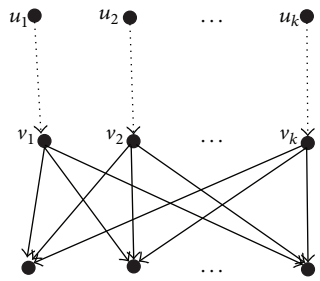


Figure 5: The topology relation of semiequivalent nodes.

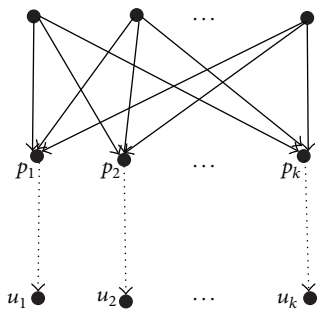


Figure 6: The topology relation of equivalent nodes.

By the definition, it follows that the semiequivalence of nodes is an equivalence relation; that is, it is reflexive, symmetric, and transitive, and the semiequivalent nodes must have the same height.

Example 7. Consider the network N_1 in Figure 1. For any node $u \in V_1 \setminus \{H1, H2\}$, u is only semiequivalent to u itself, while the nodes $H1$ and $H2$ are semiequivalent.

Property 1. If $u_1, u_2, \ldots, u_k$ are semiequivalent from the network $N = ((V, E), f)$, then $u_1, u_2, \ldots, u_k$ are the same nodes or there are the nodes v_1 (u_1 or a descendant of u_1), v_2 (u_2 or a descendant of u_2), $\ldots$, v_k (u_k or a descendant of u_k) such that $v_1, v_2, \ldots, v_k$ have the same children. See Figure 5.

Proof. We use induction on the height a of u_1 to prove it. If $a = 0$, obviously $u_1, u_2, \ldots, u_k$ are the only leaf. Thus, in this case, the property holds. We assume that the result is tenable when $a \leq n$, and let $a = n + 1$. Then the children of $u_1, u_2, \ldots, u_k$ are semiequivalent, respectively (let the children of u_i be $a_{i1}, a_{i2}, \ldots, a_{il}$ for $1 \leq i \leq k$; then $a_{1j}, a_{2j}, \ldots, a_{kj}$ are semiequivalent for $1 \leq j \leq l$), and their height is at most n by the property of node height. By the induction hypothesis, the children of $u_1, u_2, \ldots, u_k$ satisfy the property. The descendants of children of $u_1, u_2, \ldots, u_k$ are the descendants of $u_1, u_2, \ldots, u_k$. Thus, the property holds. □

Definition 8. Given a network $N = (V, E)$, we say that two nodes $u, v \in V$ (not necessarily different) are equivalent, denoted by $u \equiv v$, if $u \triangleq v$, and

(i) u, v are the root or

(ii) node u has l (≥ 1) parents $u_1, u_2, \ldots, u_l$; node v has l parents $v_1, v_2, \ldots, v_l$, and $u_i \equiv v_i$ for $1 \leq i \leq l$.

For any node u in N, it is equivalent to itself. The equivalence of nodes is also an equivalence relation. The equivalent nodes have the same height and depth.

Example 9. Consider the network N_1 in Figure 1. For any node $u \in V_1$, it is equivalent to itself. Consider the network N_1 in Figure 4. For any node $u \in V_1 \setminus \{H, J\}$, it is equivalent to itself, while the nodes H and J are equivalent to each other.

Property 2. If $u_1, u_2, \ldots, u_k$ are equivalent in the network $N = ((V, E), f)$, then $u_1, u_2, \ldots, u_k$ are the same nodes or there are the nodes p_1 (u_1 or an ancestor of u_1), p_2 (u_2 or an ancestor of u_2), $\ldots$, p_k (u_k or an ancestor of u_k) such that $p_1, p_2, \ldots, p_k$ have the same parents. See Figure 6.

Proof. We use induction on the depth b of u_1 to prove it. If $b = 0$, then $u_1, u_2, \ldots, u_k$ are the unique root node. Thus, in this case, the property holds. We assume that the result is tenable when $b \leq n$, and let $b = n + 1$. Then the parents of $u_1, u_2, \ldots, u_k$ are equivalent, respectively (let the parents of u_i be $a_{i1}, a_{i2}, \ldots, a_{il}$ for $1 \leq i \leq k$; then $a_{1j}, a_{2j}, \ldots, a_{kj}$ are equivalent for $1 \leq j \leq l$), and their depth is at most n by the property of node depth. By the induction hypothesis, the parents of $u_1, u_2, \ldots, u_k$ satisfy the property. The ancestors of the parents of $u_1, u_2, \ldots, u_k$ are the ancestors of $u_1, u_2, \ldots, u_k$. Thus, the property holds. □

In this paper, we are mainly concerned with comparing networks; the notion of node semiequivalence and equivalence will be extended to nodes from two different networks, as established in the semiequivalence and equivalence mapping of Definitions 10 and 13, respectively.

Given a set V, we use $P(V)$ to denote the set of all subsets of V.

Definition 10. Let $N_1 = ((V_1, E_1), f_1)$ and $N_2 = ((V_2, E_2), f_2)$ be two networks on $\mathcal{X}$. We define the semiequivalence mapping between N_1 and N_2, $h : V_1 \to P(V_2)$, such that $v \in h(u)$, for $u \in V_1$ and $v \in V_2$, if

(i) $u \in V_{N_1}$, $v \in V_{N_2}$, and $f_1(u) = f_2(v)$ or

(ii) node u has k (≥ 1) children $u_1, u_2, \ldots, u_k$; node v has k children $v_1, v_2, \ldots, v_k$, and $v_i \in h(u_i)$ for $1 \leq i \leq k$.

Further, while inequation $|h(u_1)| \leq 1$ holds in phylogenetic trees, it is not always the case for general phylogenetic networks.

Example 11. Consider the networks in Figure 1. h is a semiequivalence mapping between N_1 and N_2. For the reticulate nodes $H1$ and $H2$ in N_1, $h(H1) = \{h1, h2\}$ and $h(H2) = \{h1, h2\}$. For the other nodes in N_1, $h(A) = \{a\}$, $h(B) = \{b\}, \ldots, h(G) = \{g\}$, $h(1) = \{1\}, \ldots, h(4) = \{4\}$, $h(x) = \{x\}$, and $h(R) = \{r\}$.

Theorem 12. *Let $N_1 = ((V_1, E_1), f_1)$ and $N_2 = ((V_2, E_2), f_2)$ be two networks on $\mathcal{X}$, and let u_1, u_2 be two nodes in V_1 and h a semiequivalence mapping between N_1 and N_2. Assume that*

$h(u_1) \neq \emptyset$ *and* $h(u_2) \neq \emptyset$. *Then,* $u_1 \triangleq u_2$ *if and only if* $v_1 \triangleq v_2$, *for* $v_1 \in h(u_1)$ *and* $v_2 \in h(u_2)$.

Proof. For the "only if" direction, let $v_1 \in h(u_1)$, $v_2 \in h(u_2)$, and $u_1 \triangleq u_2$. Obviously, u_1, u_2, v_1, and v_2 have the same height a. Then, we use induction on such height a to prove $v_1 \triangleq v_2$. In particular, if $a = 0$, that is, $u_1, u_2 \in V_{N_1}$, and $f_1(u_1) = f_1(u_2)$, then $v_1, v_2 \in V_{N_2}$ and $f_2(v_1) = f_1(u_1) = f_1(u_2) = f_2(v_2)$. Thus, in this case, $v_1 \triangleq v_2$. We assume that the result is tenable when $a \leq n$, and let $a = n+1$. We assume that node u_1 has k children $p_1, p_2, \ldots, p_k$. Due to $u_1 \triangleq u_2$, it follows that node u_2 has k children $q_1, q_2, \ldots, q_k$, and $p_i \triangleq q_i$ $(1 \leq i \leq k)$. Due to $v_1 \in h(u_1)$ and $v_2 \in h(u_2)$, it follows that v_1 has k children $w_1, w_2, \ldots, w_k$, and $w_i \in h(p_i)$ $(1 \leq i \leq k)$, v_2 has k children $y_1, y_2, \ldots, y_k$, and $y_i \in h(q_i)$ $(1 \leq i \leq k)$. The height of p_i, q_i, w_i, and y_i is at most n. By the induction hypothesis, $w_i \triangleq y_i$. Thus, $v_1 \triangleq v_2$.

For the "if" direction, let $v_1 \in h(u_1)$, $v_2 \in h(u_2)$, and $v_1 \triangleq v_2$. Similarly, we also use induction on the same height a of u_1, u_2, v_1, and v_2 to prove $u_1 \triangleq u_2$. If $a = 0$, that is, $v_1, v_2 \in V_{N_2}$, and $f_2(v_1) = f_2(v_2)$, then $u_1, u_2 \in V_{N_1}$ and $f_1(u_1) = f_2(v_1) = f_2(v_2) = f_1(u_2)$. Thus, in this case, $u_1 \triangleq u_2$. We assume that the result is tenable when $a \leq n$, and let $a = n+1$. We assume that node v_1 has k children $w_1, w_2, \ldots, w_k$. Since $v_1 \triangleq v_2$, node v_2 has k children $y_1, y_2, \ldots, y_k$, and $w_i \triangleq y_i$ $(1 \leq i \leq k)$. Since $v_1 \in h(u_1)$ and $v_2 \in h(u_2)$, u_1 has k children $p_1, p_2, \ldots, p_k$, and $w_i \in h(p_i)$ $(1 \leq i \leq k)$, u_2 has k children $q_1, q_2, \ldots, q_k$, and $y_i \in h(q_i)$ $(1 \leq i \leq k)$. The height of p_i, q_i, w_i, and y_i is at most n by the property of node height. By the induction hypothesis, $p_i \triangleq q_i$. Thus, $u_1 \triangleq u_2$. □

Theorem 12 tells us that the semiequivalence mapping keeps the semiequivalence of nodes. Thus, all nodes in $h(u)$ are semiequivalent. Sometimes we use $h(u)$ to denote an arbitrary node in the set. We say that the nodes in $h(u)$ are semiequivalent with u.

Definition 13. Let $N_1 = ((V_1, E_1), f_1)$ and $N_2 = ((V_2, E_2), f_2)$ be two networks on $\mathcal{X}$. We define the equivalence mapping between N_1 and N_2, $g : V_1 \to P(V_2)$, such that $v \in g(u)$, for $u \in V_1$ and $v \in V_2$, if $v \in h(u)$, and

(i) u, v are the roots or

(ii) node u has l (≥ 1) parents $u_1, u_2, \ldots, u_l$; node v has l parents $v_1, v_2, \ldots, v_l$, and $v_i \in g(u_i)$, for $1 \leq i \leq l$,

where h is a semiequivalence mapping between N_1 and N_2.

Example 14. Consider the networks in Figure 1. h is the semiequivalence mapping between N_1 and N_2 discussed in Example 11. g is an equivalence mapping between N_1 and N_2 defined in Definition 13. For any node $u \in V_1 \setminus \{H1, H2, G \text{ and } x\}$, $g(u) = h(u)$, while $g(v) = \emptyset$ when $v \in \{H1, H2, G \text{ and } x\}$.

Theorem 15. *Let* $N_1 = ((V_1, E_1), f_1)$ *and* $N_2 = ((V_2, E_2), f_2)$ *be two networks on* $\mathcal{X}$, *and let* u_1, u_2 *be two nodes in* V_1. g *is an equivalence mapping between* N_1 *and* N_2. *Assume that* $g(u_1) \neq \emptyset$ *and* $g(u_2) \neq \emptyset$. *Then,* $u_1 \equiv u_2$ *if and only if* $v_1 \equiv v_2$, *for* $v_1 \in g(u_1)$ *and* $v_2 \in g(u_2)$.

Proof. Let $v_1 \in g(u_1)$, $v_2 \in g(u_2)$. Then $v_1 \in h(u_1)$, $v_2 \in h(u_2)$ based on Definition 13. For the "only if" direction, let $u_1 \equiv u_2$. We can deduce that $v_1 \triangleq v_2$ according to Theorem 12, and u_1, u_2 and v_1 and v_2 have the same depth b. Then, we use induction on b to prove that $v_1 \equiv v_2$. If $b = 0$, that is, u_1, u_2 are the unique root node of N_1, then v_1, v_2 are the unique root node of N_2. Thus, in this case, $v_1 \equiv v_2$. We assume that the result is tenable when $b \leq n$, and let $b = n+1$. We assume that node u_1 has l parents $p_1, p_2, \ldots, p_l$. Due to $u_1 \equiv u_2$, node u_2 has l parents $q_1, q_2, \ldots, q_l$, and $p_i \equiv q_i$ $(1 \leq i \leq l)$. Due to $v_1 \in g(u_1)$ and $v_2 \in g(u_2)$, v_1 has l parents $w_1, w_2, \ldots, w_l$, and $w_i \in g(p_i)$ $(1 \leq i \leq l)$, v_2 has l parents $y_1, y_2, \ldots, y_l$, and $y_i \in g(q_i)$ $(1 \leq i \leq l)$. The depth of p_i, q_i, w_i, and y_i is at most n by the property of node depth. By the induction hypothesis, $w_i \equiv y_i$. Thus, $v_1 \equiv v_2$.

For the "if" direction, let $v_1 \in g(u_1)$, $v_2 \in g(u_2)$, and $v_1 \equiv v_2$. We can deduce first that $u_1 \triangleq u_2$ according to Theorem 12. Similarly, we also use induction on the same depth b of u_1, u_2 and v_1, v_2 to prove that $u_1 \equiv u_2$. If $b = 0$, that is, v_1, v_2 are the unique root node of N_2, then u_1, u_2 are the unique root node of N_1. Thus, in this case, $u_1 \equiv u_2$. We assume that the result is tenable when $b \leq n$, and let $b = n+1$. We assume that node v_1 has l parents $w_1, w_2, \ldots, w_l$. Due to $v_1 \equiv v_2$, node v_2 has l parents $y_1, y_2, \ldots, y_l$, and $w_i \equiv y_i$ $(1 \leq i \leq l)$. Due to $v_1 \in g(u_1)$ and $v_2 \in g(u_2)$, u_1 has l parents $p_1, p_2, \ldots, p_l$, and $w_i \in g(p_i)$ $(1 \leq i \leq l)$, u_2 has l parents $q_1, q_2, \ldots, q_l$, and $y_i \in g(q_i)$ $(1 \leq i \leq l)$. The depth of p_i, q_i, w_i, and y_i is at most n. So, by the induction hypothesis, $p_i \equiv q_i$. Thus, $u_1 \equiv u_2$. □

Theorem 15 tells us that the equivalence mapping keeps the equivalence of nodes. Thus, all nodes in $g(u)$ are equivalent. Sometimes we use $g(u)$ to denote an arbitrary node in the set. We say that the nodes in $g(u)$ are equivalent to u.

Lemma 16. *Let* $N = ((V, E), f)$ *be a network and* $u, v \in V$ *two equivalent nodes. Then* u, v *belong to a superconvergent set.*

Proof. This lemma is obtained easily from Properties 1 and 2. □

Lemma 17. *Let* $N = ((V, E), f)$ *be a partly reduced phylogenetic network. Then* $u_1 \not\equiv u_2$ *for any two nodes* $u_1, u_2 \in V$.

Proof. From the partial reduction procedures of the network, we have that all superconvergent sets in a partly reduced network have been deleted. □

Given two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$, assume that $V_1 = \{v_1, v_2, \ldots, v_p\}$. The unique nodes of N_1, denoted by $L(N_1)$, is defined by the following processes. First let $L(N_1) = \emptyset$. Then for each one node $u \in V_1$, if there exists no node $u' \in L(N_1)$ such that $u' \equiv u$, add u to $L(N_1)$. We define $L(N_2)$ in a similar way. Further for each node $v_i \in L(N_1)$, we define $e_{N_1}(v_i) = |\{v \in V_1 : v \equiv v_i\}|$ and $e_{N_2}(u_i)$ similarly for each node $u_i \in V_2$. We define $e(\emptyset) = 0$ for any network N. When the context is clear, we drop the subscript of e. We are now in a position to define the measure on pairs of partly reduced phylogenetic networks.

```
(1)  input: nodes u and v
(2)  if the outdeg of u and the outdeg of v are not equal then
(3)     return
(4)  end if
(5)  if u and v are leaves and f1(u) = f1(v) (or f1(u) = f2(v) i.e., u and v are from two networks) then
(6)     add v to the ISE of u
(7)     add u to the ISE of v
(8)  else
(9)     flag := false
(10)    for each child a of u do
(11)       for each child b of v do
(12)          if b.label = true then
(13)             continue
(14)          end if
(15)          if the ISE of a has b then
(16)             flag = true
(17)             b.label = true
(18)          end if
(19)       end for
(20)       if flag = false then
(21)          return
(22)       else
(23)          flag = false
(24)       end if
(25)    end for
(26)    add v to the ISE of u
(27)    add u to the ISE of v
(28) end if
```

Algorithm 1: Deciding semiequivalence for two nodes u and v.

Definition 18. Let $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ be two phylogenetic networks on $\mathcal{X}$. Then $d_e(N_1, N_2)$ equals

$$\frac{1}{2}\left[\sum_{v\in L(N_1)} \max\left\{0, e(v) - e\left(v'\right)\right\} + \sum_{u\in L(N_2)} \max\left\{0, e(u) - e\left(u'\right)\right\}\right], \tag{1}$$

where $v'(u')$ is a node in $L(N_2)(L(N_1))$ that is equivalent to $v(u)$, and if no such equivalent node exists, then $v'(u') = \emptyset$.

Lemma 19. *If $d_e(N_1, N_2) = 0$ for two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$, then $|V_1| = |V_2|$.*

Proof. Let $g_1 : V_1 \to P(V_2)$ and $g_2 : V_2 \to P(V_1)$ be two equivalence mappings from Definition 13. Since $d_e(N_1, N_2) = 0$, it follows that $e(v_1) = e(g_1(v_1))$ (where $g_1(v_1)$ denotes a node u, which is equivalent to $g_1(v_1)$ and in $L(N_2)$) along with $|g_1(v_1)| > 0$ for all $v_1 \in L(N_1)$ and $e(v_2) = e(g_2(v_2))$ (where $g_2(v_2)$ denotes a node u, which is equivalent to $g_2(v_2)$ and in $L(N_1)$) along with $|g_2(v_2)| > 0$ for all $v_2 \in L(N_2)$. From this and Theorem 15, we have that $|V_1| = \sum_{v_1\in L(N_1)} e(v_1) = \sum_{v_1\in L(N_1)} e(g_1(v_1)) \le |V_2|$ (due to $g_1(v_1) \in V_2$) and $|V_2| = \sum_{v_2\in L(N_2)} e(v_2) = \sum_{v_2\in L(N_2)} e(g_2(v_2)) \le |V_1|$ (due to $g_2(v_2) \in V_1$). Thus $|V_1| = |V_2|$. □

Theorem 20. *Let $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ be two partly reduced networks. Then, N_1 and N_2 are isomorphic if and only if $d_e(N_1, N_2) = 0$.*

Proof. Let $g : V_1 \to P(V_2)$ be an equivalence mapping, as given in Definition 13. From Lemma 19, it follows that $|V_1| = |V_2|$ and $e(v) = e(g(v))$ for all $v \in L(N_1)$. From Lemmas 16 and 17, we have that $g(v_1)$ is defined and unique for each $v_1 \in V_1$. We now prove that if $(u, v) \in E_1$, then $(u_0, v_0) \in E_2$, where $v_0 = g(v)$ and $u_0 = g(u)$. Given that $v_0 = g(v)$, that is, v and v_0 are equivalent, this implies that v_0 and v have equivalent parents. Since $u_0 = g(u)$ is defined and unique, u_0 is a parent of v_0. Thus, $(u_0, v_0) \in E_2$. It shows that the mapping g is bijective, which also preserves the labels of the leaves and the edges of networks. Thus, N_1 and N_2 are isomorphic.

The converse implication is obvious. □

From the definition of the measure, the symmetry property follows immediately.

Lemma 21. *For any pair networks N_1 and N_2, one has $d_e(N_1, N_2) = d_e(N_2, N_1)$.*

The measure $d_e(N_1, N_2)$ can be viewed as half of the symmetric difference of two multisets on the same set of elements, where the multiplicity of element u in N_1 is $e_{N_1}(u)$ and similarly for N_2. Since the symmetric difference defines a metric on multisets [12], we have the following triangle inequality.

```
(1)  input: nodes u and v
(2)  if the indeg of u and the indeg of v are not equal, or the ISE of u doesn't have v (the ESE of u
     doesn't have v i.e., u and v are from two networks) then
(3)    return
(4)  end if
(5)  if u and v are roots then
(6)    add v to the IE of u
(7)    add u to the IE of v
(8)  else
(9)    flag := false
(10)   for each parent a of u do
(11)     for each parent b of v do
(12)       if b.label = true then
(13)         continue
(14)       end if
(15)       if the IE of a has b then
(16)         flag = true
(17)         b.label = true
(18)       end if
(19)     end for
(20)     if flag = false then
(21)       return
(22)     else
(23)       flag = false
(24)     end if
(25)   end for
(26)    add v to the IE of u
(27)   add u to the IE of v
(28) end if
```

Algorithm 2: Deciding equivalence for two nodes u and v.

Lemma 22. *Let N_1, N_2, and N_3 be three networks. Then, $d_e(N_1, N_2) + d_e(N_2, N_3) \geq d_e(N_1, N_3)$.*

From Theorem 20 and Lemmas 21 and 22, we have the following main result.

Theorem 23. *The measure d_e is a metric on the space of partly reduced phylogenetic networks.*

Proof. It follows from Theorem 20 and Lemmas 21 and 22 and the fact that $\max\{0, e(v) - e(v')\} \geq 0$. □

Let $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ be two phylogenetic networks. For a node u in N_1, we refer to its semiequivalent nodes from N_1 as internal semiequivalence (equivalence) nodes and its semiequivalent (equivalence) nodes from N_2 as external semiequivalence (equivalence) nodes. When computing the distance between two networks, we first compute internal and external equivalence nodes for every node in the two networks; subsequently by formula (1) we obtain the distance between the two considered networks. The maximum of measure $d_e(N_1, N_2)$ is $(|V_1|+|V_2|)/2.0$, when any node in N_1 and in N_2 has no external equivalence nodes.

In order to show the results of the distance computed by formula (1), we give an example as follows.

Example 24. Consider the networks in Figure 1. N_1, N_2 are two different networks on $\{1, 2, 3, 4, x\}$. However, in [11], they are indistinguishable and their m-distance [11] is 0. Now, we compute the d_e-distance between them: $d_e(N_1, N_2) = 4$ (see Example 14).

4. Computational Aspects

From the definition of semiequivalent nodes, whether in the same network or in two different networks, we have that the semiequivalent nodes can be computed by means of a bottom-up technique. Similarly, the equivalent nodes can be computed by means of a top-down technique. Let $N_1 = ((V_1, E_1), f_1)$ and $N_2 = ((V_2, E_2), f_2)$ be two phylogenetic networks. For a pair of nodes u and v, whether in the same network or in different networks, the following shows the pseudocode (Algorithm 1) that decides whether they are internal semiequivalent to each other, the pseudocode (Algorithm 2) that decides whether they are internal equivalent to each other, and the pseudocode (Algorithm 3) that computes the d_e-distance for a pair of networks (where ISE is the abbreviation for the set of internal semiequivalent nodes, ESE is the abbreviation for the set of external semiequivalent nodes, IE is the abbreviation for the set of internal equivalent nodes, and EE is the abbreviation for the set of external equivalent nodes). If two nodes u and v from the same network are semiequivalent, then we add u to the ISE of v and add v to the ISE of u. Obviously, this decision costs at

```
(1)  input: networks N_1 = (V_1, E_1) and N_2 = (V_2, E_2)
(2)  output: d_e-distance
(3)  for each pair of nodes u and v in V_1 do
(4)     decide semi-equivalence and equivalence for them
(5)  end for
(6)  for each pair of nodes u and v in V_2 do
(7)     decide semi-equivalence and equivalence for them
(8)  end for
(9)  for each pair of nodes u in V_1 and v in V_2 do
(10)    decide semi-equivalence and equivalence for them
(11) end for
(12) L(N_1) = ∅; L(N_2) = ∅
(13) flag1 = false; flag2 = false
(14) for each node u in V_1 do
(15)    for each node v in L(N_1) do
(16)       if the IE of v contains u then
(17)          flag1 = true
(18)       end if
(19)    end for
(20)    if flag1 = false then
(21)       add u to L(N_1)
(22)    end if
(23) end for
(24) for each node u in V_2 do
(25)    for each node v in L(N_2) do
(26)       if the IE of v contains u then
(27)          flag2 = true
(28)       end if
(29)    end for
(30)    if flag2 = false then
(31)       add u to L(N_2)
(32)    end if
(33) end for
(34) d = 0
(35) for each node u in L(N_1) do
(36)    c = |IE| − |EE|
(37)    if c > 0 then
(38)       d = d + c
(39)    end if
(40) end for
(41) for each node u in L(N_2) do
(42)    c = |IE| − |EE|
(43)    if c > 0 then
(44)       d = d + c
(45)    end if
(46) end for
(47) return d = d/2
```

Algorithm 3: Computing the d_e-distance for $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$.

most $O(n^3)$ time, where $n = \max(|V_1|, |V_2|)$. So, it takes totally $O(n^5)$ time to find out all internal and external semiequivalent nodes for every node in the two networks. In a similar way, we have that it also takes $O(n^5)$ time to find out all internal and external equivalent nodes for every node in the two networks. Subsequently we spend $O(n)$ time computing the formula (1). In conclusion, it costs totally $O(n^5)$ time to compute the distance between two networks, where n is the maximum between their node numbers.

5. Conclusion

In [11], Nakhleh introduced a polynomial-time computable m-distance in the space of reduced phylogenetic networks. In order to enlarge the space of phylogenetic networks we can compare, we devised a polynomial-time computable d_e-distance on the space of partly reduced phylogenetic networks, which can be viewed as half of the symmetric difference of two multisets on the same set of elements. To our knowledge, the space is the largest space that has a polynomial-time computable metric. d_e-distance is also a metric on the space of reduced phylogenetic networks which is included in the space of partly reduced phylogenetic networks. In general, for two phylogenetic networks, their d_e-distance is larger than their m-distance. From [12], we have that the d_e-distance is also a metric on the space of tree-child phylogenetic networks, semibinary tree-sibling time consistent phylogenetic networks, and multilabeled phylogenetic trees. However, the d_e-distance is not a metric on the space of all rooted phylogenetic networks; for example, in the two phylogenetic networks in Figure 4, their d_e-distance is 0, but they are not isomorphic.

d_e-distance can also apply to computing the dissimilarity for other types of networks, such as spiking neural networks [18–20], which will be a direction of further research.

Competing Interests

The author declares that they have no competing interests.

Acknowledgments

This work was supported by the Natural Science Foundation of Inner Mongolia province of China (2015BS0601).

References

[1] M. Pagel, "Inferring the historical patterns of biological evolution," *Nature*, vol. 401, no. 6756, pp. 877–884, 1999.

[2] J. Wang, M. Guo, X. Liu et al., "Lnetwork: an efficient and effective method for constructing phylogenetic networks," *Bioinformatics*, vol. 29, no. 18, pp. 2269–2276, 2013.

[3] J. Wang, M. Guo, L. Xing, K. Che, X. Liu, and C. Wang, "BIMLR: a method for constructing rooted phylogenetic networks from rooted phylogenetic trees," *Gene*, vol. 527, no. 1, pp. 344–351, 2013.

[4] J. Wang, "A new algorithm to construct phylogenetic networks from trees," *Genetics and Molecular Research*, vol. 13, no. 1, pp. 1456–1464, 2014.

[5] J. Wang, M.-Z. Guo, and L. L. Xing, "FastJoin, an improved neighbor-joining algorithm," *Genetics and Molecular Research*, vol. 11, no. 3, pp. 1909–1922, 2012.

[6] L. Nakhleh, J. S. T. Warnow, C. R. Linder, B. M. Moret, and A. Tholse, "Towards the development of computational tools for evaluating phylogenetic network reconstruction methods," in *Proceedings of the 18th Pacific Symposium on Biocomputing*, Kauai, Hawaii, USA, January 2003.

[7] B. M. E. Moret, L. Nakhleh, T. Warnow et al., "Phylogenetic networks: modeling, reconstructibility, and accuracy," *IEEE/ACM*

Transactions on Computational Biology and Bioinformatics, vol. 1, no. 1, pp. 13–23, 2004.

[8] M. Baroni, C. Semple, and M. Steel, "A framework for representing reticulate evolution," *Annals of Combinatorics*, vol. 8, no. 4, pp. 391–408, 2004.

[9] G. Cardona, F. Rosselló, and G. Valiente, "Tripartitions do not always discriminate phylogenetic networks," *Mathematical Biosciences*, vol. 211, no. 2, pp. 356–370, 2008.

[10] G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente, "A distance metric for a class of tree-sibling phylogenetic networks," *Bioinformatics*, vol. 24, no. 13, pp. 1481–1488, 2008.

[11] L. Nakhleh, "A metric on the space of reduced phylogenetic networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 2, pp. 218–222, 2010.

[12] G. Cardona, M. Llabrès, F. Rossellò, and G. Valiente, "On Nakhleh's metric for reduced phylogenetic networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 4, pp. 629–638, 2009.

[13] Q. Zou, Q. Hu, M. Guo, and G. Wang, "HAlign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[14] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: a survey," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 55–64, 2016.

[15] Q. Zou, X.-B. Li, W.-R. Jiang, Z.-Y. Lin, G.-L. Li, and K. Chen, "Survey of MapReduce frame operation in bioinformatics," *Briefings in Bioinformatics*, vol. 15, no. 4, Article ID bbs088, pp. 637–647, 2014.

[16] G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente, "The comparison of tree-sibling time consistent phylogenetic networks is graph isomorphism-complete," *The Scientific World Journal*, vol. 2014, Article ID 254279, 6 pages, 2014.

[17] K. S. Booth and C. J. Colbourn, "Problems polynomially equivalent to graph isomorphism," http://cs.uwaterloo.ca/research/tr/1977/CS-77-04.pdf.

[18] S. Chowhan, U. V. Kulkarni, and G. N. Shinde, "Iris recognition using modified fuzzy hypersphere neural network with different distance measures," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 6, 2011.

[19] A. Van Schaik, "Building blocks for electronic spiking neural networks," *Neural Networks*, vol. 14, no. 6-7, pp. 617–628, 2001.

[20] B. J. Graham and D. P. M. Northmore, "A spiking neural network model of midbrain visuomotor mechanisms that avoids objects by estimating size and distance monocularly," *Neurocomputing*, vol. 70, no. 10–12, pp. 1983–1987, 2007.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 4516376, 9 pages
http://dx.doi.org/10.1155/2016/4516376

Research Article

Segmentation of MRI Brain Images with an Improved Harmony Searching Algorithm

Zhang Yang,[1] Ye Shufan,[2] Guo Li,[3] and Ding Weifeng[4]

[1]*School of Information and Engineering, Wenzhou Medical University, Wenzhou, Zhejiang 325000, China*
[2]*Zhejiang ZhongLan Environment Technology Ltd., Wenzhou, Zhejiang 325000, China*
[3]*School of Medical Imaging, Tianjin Medical University, Wenzhou, Zhejiang 300000, China*
[4]*118 Hospital of the People's Liberation Army, Wenzhou, Zhejiang 325000, China*

Correspondence should be addressed to Ye Shufan; neuyeshufan@163.com

Received 15 March 2016; Accepted 7 April 2016

Academic Editor: Yungang Xu

The harmony searching (HS) algorithm is a kind of optimization search algorithm currently applied in many practical problems. The HS algorithm constantly revises variables in the harmony database and the probability of different values that can be used to complete iteration convergence to achieve the optimal effect. Accordingly, this study proposed a modified algorithm to improve the efficiency of the algorithm. First, a rough set algorithm was employed to improve the convergence and accuracy of the HS algorithm. Then, the optimal value was obtained using the improved HS algorithm. The optimal value of convergence was employed as the initial value of the fuzzy clustering algorithm for segmenting magnetic resonance imaging (MRI) brain images. Experimental results showed that the improved HS algorithm attained better convergence and more accurate results than those of the original HS algorithm. In our study, the MRI image segmentation effect of the improved algorithm was superior to that of the original fuzzy clustering method.

1. Introduction

With the development of image processing technology, medical imaging technology has significantly improved. A wide variety of medical images are currently being produced. Several currently available imaging approaches include computed tomography, magnetic resonance imaging (MRI), and ultrasound. These techniques are extensively used in medical diagnosis, preoperative planning, treatment, and postmanagement detection. MRI is commonly used in actual clinical diagnosis. Compared with other technologies, MRI does not employ radiation on the human body. At the same time, high-resolution imaging of human soft tissue is attained, which can be achieved in any Italian dimensional imaging [1–4]. Although MRI technology is extensively used in medicine, MRI data and images can be generated under objective and subjective reasons for data transmission as well as the environment, and other instruments produce gradation unevenness and offset field effect. Limited resolutions produce similar noise effects. Therefore, improving MRI technology is important to enhance analysis. Under MRI, the skull is relatively bright white. The range of gray values in the skull and white matter usually overlaps. The skull bone and muscle exhibit gray values similar to those of brain tissue. In a segment containing white and gray matter, the skull is resolved together with the white matter. Therefore, the accurate segmentation of MRI images is important to eliminate interference. Methods for regional enlargement is suitable for achieving clear image segmentation of the target boundaries. If the target is unclear, then the image cannot be effectively extracted. Dynamic contour models generate enhanced segmentation effects but are disadvantageous because of long computing time. Meanwhile, deformable model methods are divided into two categories, namely, the parametric deformable model and variable-level set-shape model. These methods are achieved by iterative calculation,

which takes a long time. Manual determination of iteration points must first be achieved. Given the involvement of personal and subjective factors, the segmentation attained by such method is unstable. Mathematical morphological imaging of the skull for removal treatment is effective, but a suitable threshold is more difficult to achieve using such technique. Falcao et al. [5] proposed the use of a live-wire segmentation algorithm. The algorithm can provide the user effective control of the segmentation process. In the approach, the user can intervene with the results of segmentation. Another method utilizes the artificial neural network (ANN), which is composed of many processing units (nodes). The ANN can simulate the biological, particularly the massively parallel, network of the human brain learning process. Input data acquire results quickly by training under ANN theory. With such strategy, the speed of image segmentation is effectively improved. The neural network algorithm does not entail prior knowledge of the probability distribution of image gray values; consequently, segmentation results are similar to the original image [6]. The neural network method shows its unique advantages in solving a series of complex image segmentations; however, several issues arise. First, the energy function in such case falls into local minimum values for minimized images. Second, the convergence of the neural network is related to the data; thus, a suitable value for testing network inputs is needed. The fuzzy clustering method is more extensively applied for image segmentation. The fuzzy clustering algorithm has the following advantages: the algorithm avoids the issue of threshold setting and does not entail human manipulation. Furthermore, the fuzzy clustering algorithm is particularly suitable for fuzzy and uncertain images. In this study, the fuzzy c-means clustering algorithm was selected to segment MRI brain images. Scholars discovered that the effect of the initial value of the cluster centers of the fuzzy clustering algorithm is relatively large. The characteristics of a nonconvex function involve several local minima; thus, the initial value of FCM will fall into the local minima. This study utilized the rough set to compute for the initial value of the FCM.

The harmony searching (HS) algorithm was developed by Korean scholars. Geem et al. [7, 8] proposed a kind of intelligent optimization algorithm in 2001. The algorithm describes the process of musical improvisation; different musical tones are applied to a harmony vector to search for a harmony randomly. Then, the process attains an optimal harmony. Jang et al. [9] used the Nelder-Mead simplex HS algorithm, whereas Mahdavi et al. [10] adopted the adaptive HS algorithm (IHS). Omran and Mahdavi compared the performances, parameters, and noise effects related to the original HS algorithm [11], IHS, and global-best HS algorithm. H.-Q. Li and L. Li [12] employed the genetic algorithm and the HS algorithm to explore the three functions of Rastrigin, Griewank, and Sphere. Liang and coworkers [13, 14] adjusted certain parameters to improve the HS algorithm and used a hybrid GA-HS algorithm to solve the critical sliding slope problem. Cheng et al. [15] adopted the HS algorithm with several other heuristic optimization algorithms for earth slope stability analysis. Dong et al. [16] proposed the HS k-means clustering algorithm to change WEB text categorization. Bezdek [17] utilized an adaptive adjustment of parameters on the improved HS algorithm to solve the anomaly detection problems in digital images of biological tissue.

In recent years, the HS algorithm has been adopted in several applications. However, the algorithm exhibits several disadvantages. The HS algorithm operates with weak robustness, considerable randomness, lack of specific direction, and slow convergence speed; it easily falls into the local optimal solution. The problem can be attributed to the search mechanism of the HS algorithm. This study proposed an improved HS algorithm for MRI brain image segmentation to overcome the aforementioned disadvantages. We used the rough set and memory bank of the HS algorithm together with the concept of rough set upper and lower boundary correction HS algorithm of the "optimal" and "worst" harmonies. By doing so, we prevented the HS algorithm convergences from falling into the local optimum. The HS algorithm should be employed to obtain the number of optimal solutions as the initial value for the average fuzzy clustering algorithm. This strategy would overcome the random determination of the initial value of the fuzzy clustering algorithm. Experimental results showed that the proposed algorithm achieved perfect convergence, and the segmentation effect was ideal for the MRI brain images. Besides ANN and fuzzy clustering, ensemble learning [18, 19], feature ranking [20], and samples selection [21] were also employed in the biomedical research.

2. Harmony Searching Algorithm

The HS algorithm has been proposed as a new algorithm for the study of musical play. Each musician produces individual tones that can generate vector values. If the music produced is pleasant sounding, then the tone is recorded and tools are employed to generate a better harmony in the subsequent attempt. Musical harmony is analogous to the optimal solution vector, whereas the player riffs correspond to the optimization techniques in the local and global search programs. The HS algorithm uses a random search that selects probabilities and adjusts the pitch without information derived from the harmony. Compared with early heuristic optimization algorithms, the HS algorithm is conceptually straightforward, utilizing less mathematical expressions and a few parameters for the random search of theoretical values. Moreover, the algorithm can be more easily optimized for various engineering problems. These theoretical ideas can be adopted to formulate the solution vector $X^j = (X_1^j, X_2^j, \ldots, X_n^j)$, which refers to the evaluation function for $f(X^j)$. The HS algorithm is mainly divided into the following steps.

Step 1 (initialization parameter). The HS algorithm includes a series of important parameters, such as the number of iterations k; the harmony memory data base HM; the harmony memory probability values $\mathrm{PAR}_{\max}$ and $\mathrm{PAR}_{\min}$; the fine-tuning probability BW; the harmony memory size HMS;

the dimension parameter optimization problem N; and the upper and lower boundaries x^{U} and x^{L}, respectively.

Step 2 (harmony memory initialization). A harmony database is adopted to store the HMS of random harmonic vectors. The random harmonic vectors by weight of each dimension on the upper and lower boundaries x^{U} and x^{L}, respectively, are expressed as follows:

$$x_i = x^{\mathrm{L}} + \mathrm{rand}\left(x^{\mathrm{U}} - x^{\mathrm{L}}\right) \quad i = 1, 2, \ldots N. \tag{1}$$

The HM matrix expression is as follows:

$$\mathrm{HM} = \begin{vmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \mid f\left(\overrightarrow{x^1}\right) \\ x_1^2 & x_2^2 & \cdots & x_N^1 \mid f\left(\overrightarrow{x^2}\right) \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}} & x_1^{\mathrm{HMS}} & \cdots & x_1^{\mathrm{HMS}} \mid f\left(\overrightarrow{x^{\mathrm{HMS}}}\right) \end{vmatrix}. \tag{2}$$

Step 3 (new harmony generation). In accordance with the change in objective function value, the adaptive setting of the harmony memory considers the probability HMCR. By the maximum and minimum sounds of the initial setup, probability dynamic adjustment PAR is achieved. After parameter adjustment, learning by the differences in operation, pitch adjustment, and random mutation process, new harmonic solution vectors are created.

The process involved is as follows:

For $i \in [1, N]$ % *Beginning,*

if rand( ) <= HMCR *then*

$x_i' = x_i/(j = ceil(\mathrm{rand}( ) * \mathrm{HMS}))$ % *memory consideration*

if rand( ) <= PAR *then*

$x_i' = x_i \pm \mathrm{rand}( ) * \mathrm{BW}$ % *pitch adjustment*

else

$x_i' = x_i^{\mathrm{L}} + \mathrm{rand}( ) * (x_i^{\mathrm{L}} - x_i^{\mathrm{U}})$ % *random selection*

end if

Step 4 (memory bank updating). The new harmony is regarded as the worst harmony in the database, and the most optimal update for the worst harmony is utilized in the database.

Step 5 (termination conditions). The current number of iterations is determined to achieve the maximum number of iterations. Once the number of maximum iterations is achieved, the terminate iteration cycle is commenced through Steps 3 to 5.

3. Harmony Searching Algorithm Improvement

Compared with other optimization algorithms, the HS algorithm is superior on the basis of the following reasons. (1) The HS algorithm requires minimal mathematical criteria and does not entail variable initialization. (2) The entire search process of the HS algorithm assumes a completely random pattern without considerable manual intervention. (3) The HS algorithm considers the entire available acoustic memory information to create a new harmony vector. Given these advantages, the HS algorithm has been the focus of attention of many foreign scholars since 2001.

The randomness of the algorithm results in low precision. The HS algorithm is mainly adopted to improve the accuracy of the optimization problem. The HS algorithm is a strong randomness heuristic algorithm, has a simple structure, is easily operated, and involves only a few parameters and other characteristics. However, the HS algorithm also adopts sensitive parameters and generates slow convergence defects, thereby entailing further research on enhancements. The HS algorithm uses a few paramount parameters that directly affect the algorithm. These parameters include the harmony memory probability values $\mathrm{PAR}_{\max}$ and $\mathrm{PAR}_{\min}$ and the fine-tuning probability BW. In this study, the accuracy of the HS algorithms is improved to prevent attaining a premature local optimum. In this regard, the following enhancements were applied.

3.1. Construction of a New Harmony HM Database. In the original harmony algorithm, harmony memory acquisition is random, thereby entailing a relatively large stochastic algorithm. This effect reduces the accuracy of the algorithm. In this study, a rough set was employed on the upper and lower boundaries to establish a new harmony HM database. Rough set theory was adopted to reduce the randomness of the harmony memory database and improve the latter's accuracy.

Step 1. The relationship $1 \le i \le k$ was applied, where k is the clustering center, to establish the initial average Z_i.

Step 2. With the data points x_i, $1 \le i \le n$, the limits of the upper and lower boundaries, $\overline{B}U_{i'}$ and $\underline{B}U_{i'}$, respectively, were almost reached. $\overline{B}U_{i'}$ and $\underline{B}U_{i'}$ are the limits of the clustering center $U_{i'}$. $U_{i'}$ was adopted to denote the distance between two points $d_{i'j} - d_{ij}$.

Step 3. If d_{ij} corresponds to the extreme minimum value, then $d_{i'j}$ must be close to d_{ij}. If $d_{i'j} - d_{ij}$ is less than a given threshold, then $x_{ij} \in \underline{B}U_{i'}$; otherwise, $x_{ij} \in \overline{B}U_{i'}$.

Step 4. A lower limit is established on the matrix, as follows:

$$Z_i = \begin{cases} \sum x_{j\in(\overline{B}U_{i'}-\underline{B}U_{i'})} x_j & \text{if } \underline{B}U_{i'} = \phi,\ \overline{B}U_{i'} - \underline{B}U_{i'} \neq \phi \\ w_{\text{low}} \dfrac{\sum x_{j\in(\overline{B}U_{i'}-\underline{B}U_{i'})} x_j}{\left|\underline{B}U_i\right|} + w_{\text{up}} \dfrac{\sum x_{j\in(\overline{B}U_{i'}-\underline{B}U_{i'})} x_j}{\left|\overline{B}U_{i'} - \underline{B}U_{i'}\right|} & \text{if } \underline{B}U_{i'} \neq \phi,\ \overline{B}U_{i'} - \underline{B}U_{i'} \neq \phi \\ \dfrac{\sum x_{j\in\underline{B}U_{i'}} x_j}{\left|\underline{B}U_i\right|} & \text{otherwise.} \end{cases} \tag{3}$$

According to the previously presented steps, alternate data were used for preliminary screening to establish the harmony algorithm. However, more accurate data were required by the algorithm. Thus, the K-nearest neighbor (KNN) algorithm was employed to attain the appropriate harmony memory matrix. K denotes a given clustering center based on prior knowledge.

The KNN method [22] was originally proposed in 1968 by Cover and Hart. The KNN is a theoretically more mature classification algorithm. The core idea of the KNN algorithm is simple: if a sample feature vector space k most similar (the nearest feature vector space) to the sample belongs to the major category, then the sample likely belongs to such category. The KNN method for the decision-making category is based solely on the nearest category or several categories of samples to which samples are designated to. The traditional KNN algorithm has been referred to as an example-based learning classification algorithm. By comparing each training sample, users find the text to be classified with the most similar K text. Finally, the text that contains the greatest number of similar categories is selected and classified as category text. The related mathematical expression is as follows:

$$p\left(d_i, C_j\right) = \sum_{d_j \in \text{KNN}} \text{sim}\left(d_i, d_j\right) y\left(d_i, C_j\right), \tag{4}$$

where d_i is the feature vector, $\text{sim}(d_i, d_j)$ corresponds to similarity, and $y(d_i, C_j)$ denotes the classification properties. If d_i belongs to C_j, then the value of the function is 1; otherwise, the value is 0. Herein, we used this kind of thinking process for classification.

In distributing the matrix Z_i sample items across the class space, we applied Euclidean distance as a distribution rule as follows:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{n} \left(v_k\left(x_i\right) - v_k\left(x_i\right)\right)^2}. \tag{5}$$

If $Z_i = \{z_1, z_2, \ldots, z_n\}$, then each area involves a k clustering center. The aforementioned methods were adopted to establish a suitable search memory database HM as follows:

$$\text{HM} = K_{1i} * K_{2i}, \ldots, K_{Ni}. \tag{6}$$

Step 5. When all the optional data maximum and minimum values were less than the threshold T, the loop was terminated. Otherwise, Steps 2 to 4 were repeated to establish the appropriate database matrix of harmony. Herein, we considered $w_{\text{up}} = 1 - w_{\text{low}}$, $0.5 < w_{\text{low}} < 1$.

Matrix HM_{Z_i} was established by using the new harmony matrix of rough set theory. By using rough set theory to establish a harmony matrix principle instead of a random matrix, we avoided the poor robustness and randomness of the HS algorithm.

3.2. Probability PAR Adjustment. A study on the HS algorithm revealed that probability PAR tuning and volume BW are set randomly or by experience. In such case, no change in the convergence process is achieved. In fact, the effect of these two parameters on the convergence of the algorithm is relatively large, particularly in the latter part of the run. The original HS algorithm is not concerned with this aspect; it is not conducive for a fast algorithm that converges to the global optimum. In this study, the PAR and BW parameters in the original HS algorithm were improved to avoid falling into the local optimum.

In the HS algorithm, adjusting the probability PAR is also an important component. In the literature [10], the value of a small PAR has been shown to enhance the local search ability of the algorithm. By contrast, the value of a larger PAR is beneficial for adjusting the search area. The expression is shown as follows:

$$\text{PAR}_T = \text{PAR}_{\min} + \frac{\text{PAR}_{\max} - \text{PAR}_{\min}}{T} * t, \tag{7}$$

where T is the iteration number and t is the current number of iterations. In this study, the global search algorithm was improved by introducing a feedback mechanism and moving a step length. The number of iterations T was also updated. To update the probability of the harmony memory database and step length, we adopted the following expression:

$$T = \frac{\sum_{i=1}^{c} \left(X_{\text{best}}^i - X_{\text{worst}}^i\right)}{\sqrt{\sum_{i=1}^{c} \left(X_{\text{best}}^i - X_{\text{worst}}^i\right)^2}}. \tag{8}$$

The t times moving steps were expressed as follows:

$$\text{BW}(T) = \begin{cases} \text{BW}_{\min} + \dfrac{\left(\text{BW}_{\max} - \text{BW}_{\min}\right) * T(t)}{T_{\max}} & \text{if } T < \dfrac{T_{\max}}{2} \\ \text{BW}_{\max} & \text{if } T \geq \dfrac{T_{\max}}{2}. \end{cases} \tag{9}$$

By improving the HS algorithm using dynamic tone control, adjustable probability PAR values and bandwidth BW were attained, overcoming the shortcomings in probability PAR value and bandwidth generated by the fixed tone control in the basic HS algorithm. Compared with other algorithms, whether on test function or vector search solutions, the enhanced HS algorithm exhibited a better performance.

3.3. Termination Conditions. At the maximum or minimum harmony database values less than the threshold T, the loop was terminated. Otherwise, the original HS algorithm was repeated from Steps 2 to 4.

4. Fuzzy Clustering Segmentation

The FCM clustering algorithm was proposed using fuzzy set theory. The FCM uses fuzzy set theory for classification. Data under a certain degree of categorization is divided into various types, and cluster centers are calculated in accordance with all the updated data objects of each category. This ambiguity makes the classification process of the FCM algorithm better reflect the actual data distribution, particularly for the treatment of overlap between all categories.

FCM clustering image segmentation treats pixels in an image as a cluster sample and the entire diagram as a sample set; each pixel feature vector is extracted from the image and regarded as the sample; then the pixels in the feature space are clustered. In essence, the pixels with similar characteristics are grouped in an aggregate class, whereas the pixels with dissimilar features are distributed into different classes. Finally, each pixel is completely tagged to image segmentation.

In the fuzzy means clustering algorithm (FCM), the initial value setting c is a more important direct effect of segmentation speed, accuracy, and effectiveness. Before starting, the cluster number must be given first. However, in the absence of human intervention and prior knowledge of the image, such as in an automated system, determining the cluster number is a difficult task. Therefore, c values based on image segmentation problems are difficult to determine under fuzzy clustering. In traditional FCM, the initial value c is random. Thus, the randomness of the algorithm is high and a local optimal solution is attained.

In this study, the initial value is regarded as the number of optimal solutions obtained by the HS algorithm; the algorithm can achieve a favorable result by avoiding the local optimal solution. The MRI brain image segmentation effect attained by the improved algorithm is better than that achieved through the traditional FCM. A previous study [17] promoted the objective function of the FCM clustering algorithm; the related expression is as follows:

$$J\left(U \in M_{hc}, V\right) = \sum_{j=1}^{c} \sum_{k=1}^{N} u_{ji} d_{ij}^2, \tag{10}$$

where $U = [u_{ki}]_{c*n} \in M_{hc}$, m is the fuzzy index, and d_{ij} is the distance between the clustering center and clustering objects. In this study, the Euclidean metric distance was adopted to compute the gray difference between any point and the cluster center. The Euclidean metric distance can be calculated with minimum steps.

The minimum value refers to the direction of clustering in $\sum_{k=1}^{c} u_{ki} = 1$ under the condition of the constraint $J_m(U, V)$. By using the Lagrangian approximation solution, the degree of membership and cluster center under the extreme value are calculated as follows:

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{j=1}^{N} \left(u_{ij}\right)^m \left(d_{ji}\right)^2 + \sum_{j=1}^{N} \lambda_i \left(\sum_{i=1}^{c} u_{ji} - 1 \right), \tag{11}$$

$$u_j(x_i) = \frac{\left(\left(1/\left\|x_i - m_j\right\|\right)^2\right)^{1/b-1}}{\sum_{k=1}^{c} \left(\left(1/\left\|x_i - m_j\right\|\right)^2\right)^{1/b-1}}, \quad i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, c, \tag{12}$$

$$v_i = \frac{\sum_{k=1}^{N} u_{ki}^m x_k}{\sum_{k=1}^{N} u_{ki}^m}, \quad i = 1, 2, \ldots, c. \tag{13}$$

Equations (9) and (11) were utilized by continuous iterative optimized clustering. Each iteration was adopted to calculate the membership degree matrix and cluster center until convergence was reached.

The detailed steps for FCM calculation are as follows:

(1) The optimal value of c was obtained using the improved HS algorithm and rough set theory as the initial value c for the FCM algorithm.

(2) The clustering center vector $V = [v_1, v_2, \ldots, v_c]$ was initialized.

(3) On the basis of (10), the membership degree matrix $U^{(t)} = [u_{ki}^{(t)}]_{c*n}$ was updated, where t denotes the iteration number.

(4) Equation (11) was employed to update the clustering center $V' = [v_1', v_2', \ldots, v_c']$.

(5) The number of iterations T or error parameter ε when $t < T$ or $|V - V'| > \varepsilon$ was determined. Then, Steps 3 to 5 were repeated until the loop was terminated.

5. Experiment

In a simulation experiment, the improved harmonic search algorithm and the original harmony algorithm were used to obtain the optimal, worst, and average values for MRI brain images 1–4 (MRI1–4).

In the data index, all values of the improved HS algorithm were superior to those of the original HS algorithm. The improved HS algorithm also obtained a better optimal solution. The different values were computed using Euclidean distance. The quantitative units were expressed to 10^5.

In the experimental data (Table 1), a smaller optimal value indicates a nearer distance to the clustering center and a more accurate selection of the cluster center. As the average value approaches the optimal value, the more optimal condition

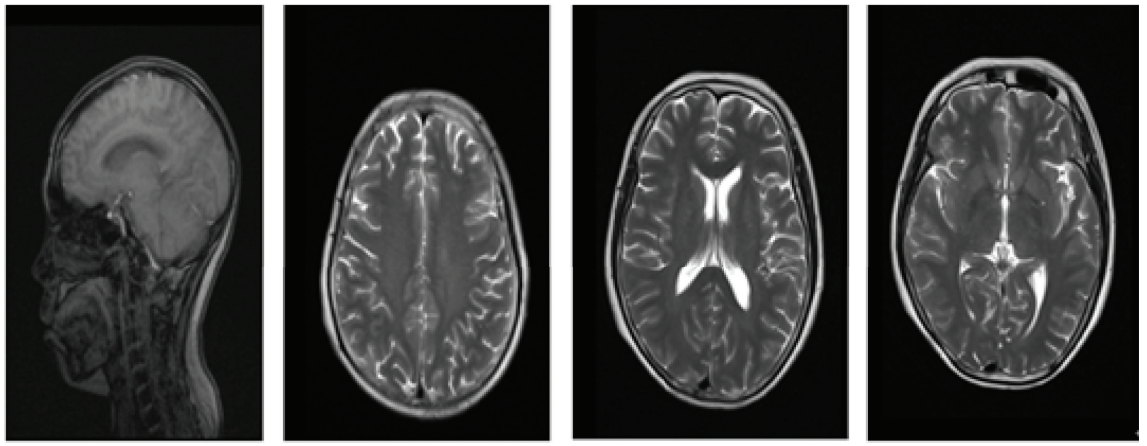


Figure 1: Original MRI images.

Table 1: Experimental results and comparison of data.

Image	Algorithm	Best value	Worst value	Average value
MRI1	Improved HS	2.17848	2.40321	2.18952
	HS	2.37648	3.17136	2.48956
MRI2	Improved HS	4.68368	5.03184	4.69925
	HS	4.78272	5.29190	4.89628
MRI3	Improved HS	4.24348	6.88194	4.25740
	HS	4.37389	9.16356	5.06598
MRI4	Improved HS	3.35994	4.70662	3.45236
	HS	4.5226	5.00091	4.63587

of the cluster center is achieved. More precise cluster centers attain better segmentation effects.

The coefficient segmentation function V_{pc} and entropy segmentation function V_{pe} were utilized to qualitative analyze the experimental results [23, 24]. The related expressions are as follows:

$$V_{pc} = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2,$$
$$V_{pe} = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\log u_{ij}. \quad (14)$$

In the simulation experiments, the improved search algorithm and fuzzy clustering segmentation were compared with the original fuzzy clustering (FCM) segmentation algorithm. The experimental results are shown in Table 2.

The value of the coefficient segmentation function V_{pc} of the improved algorithm was greater than that of the FCM algorithm (Table 2). Conversely, V_{pe} values were lower in the improved algorithm than in the FCM algorithm. For image segmentation, a high V_{pc} or a low V_{pe} indicates perfect segmentation effects.

Consequently, the segmentation effect of the improved algorithm is better than that of the FCM algorithm. The data of the FCM algorithm shown in Table 2 reveal that the values of V_{pc} and V_{pe} are closed. This result can be explained by the

Table 2: Experimental results and comparison of data.

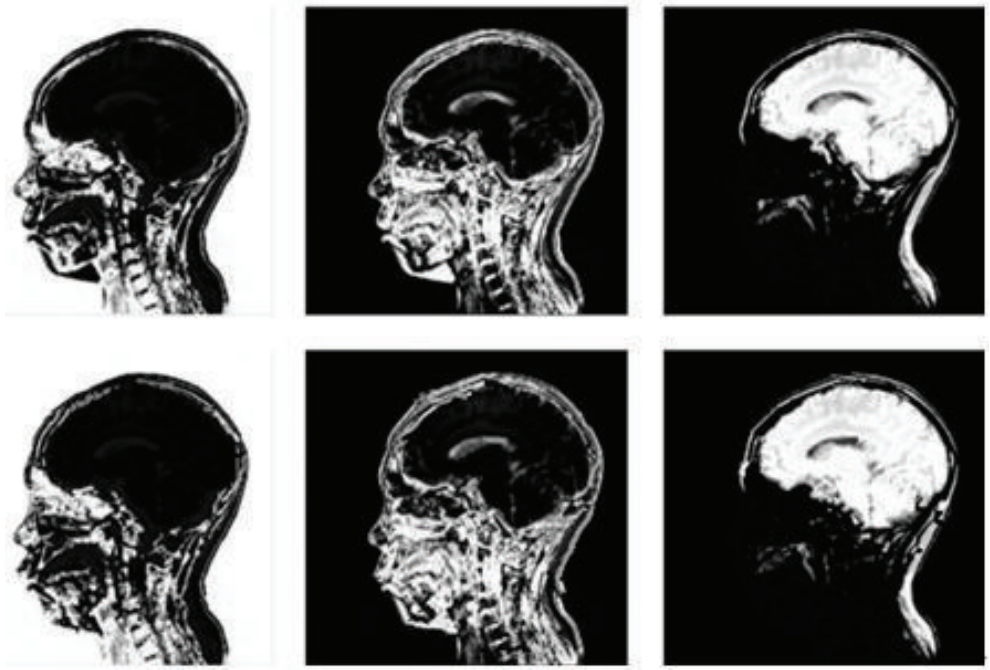
Image	Algorithm	V_{pc}	V_{pe}	Iterations
MRI1	Improved algorithm	0.873789	0.231145	27
	FCM	0.689989	0.656345	20
MRI2	Improved algorithm	0.881977	0.314585	29
	FCM	0.699575	0.642841	23
MRI3	Improved algorithm	0.903779	0.217534	29
	FCM	0.668798	0.656146	21
MRI4	Improved algorithm	0.887907	0.334695	29
	FCM	0.657174	0.632898	23

fuzzy clustering algorithm in images MRI1 and MRI2, which involved 20 and 23 iteration times into the local optimal solution, respectively. Moreover, in 21 and 23 iteration times, MRI3 and MRI4 fell into the local optimal solution. Thus, the existence of the local optimal solution rendered the FCM algorithm not ideal for MRI image segmentation.

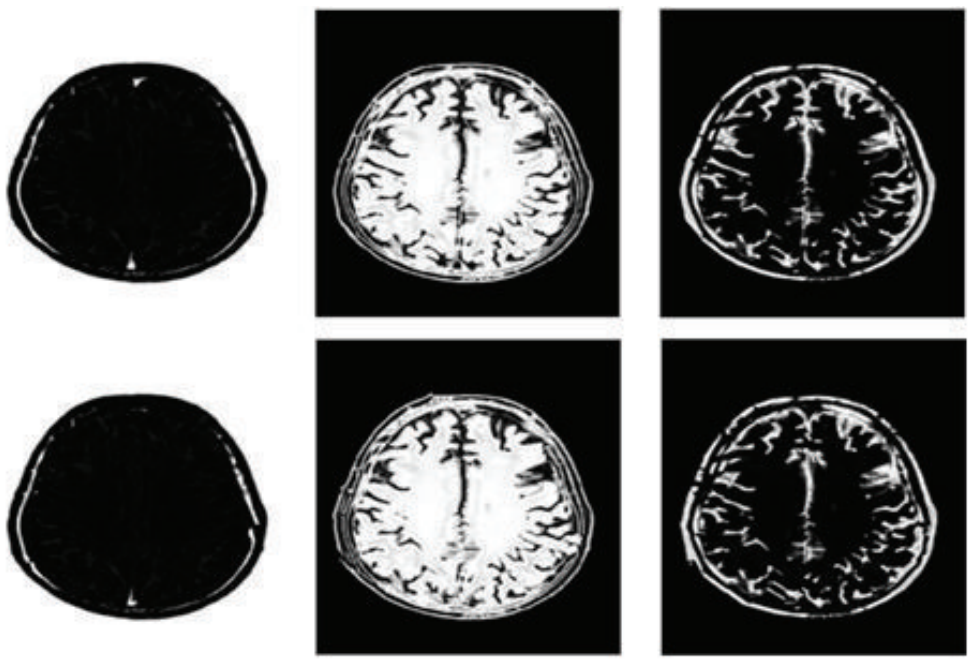
By using the improved algorithm, MRI1–4 brain image segmentation obtained different initial values of c. The differences in initial value also changed the partitioning membership.

The final segmentation results for MRI1 and MRI2 are shown in Figure 2. When the improved algorithm was used to determine the partition $c = 3$, the original FCM algorithm was set to $c = 3$. Meanwhile, the final segmentation results for MRI3 and MRI4 are shown in Figure 3. When the improved algorithm was used to determine the partition $c = 4$, the original FCM algorithm was also set to $c = 4$.

Figure 1 displays the experimental data for images MRI1–4. Meanwhile, Figure 2 shows the segmentation results for images MRI1 and MRI2. The membership is associated with the initial value $c = 3$. The first and second rows display the segmentation effects. The first row shows the experimental



(a) Segmentation results for MRI1

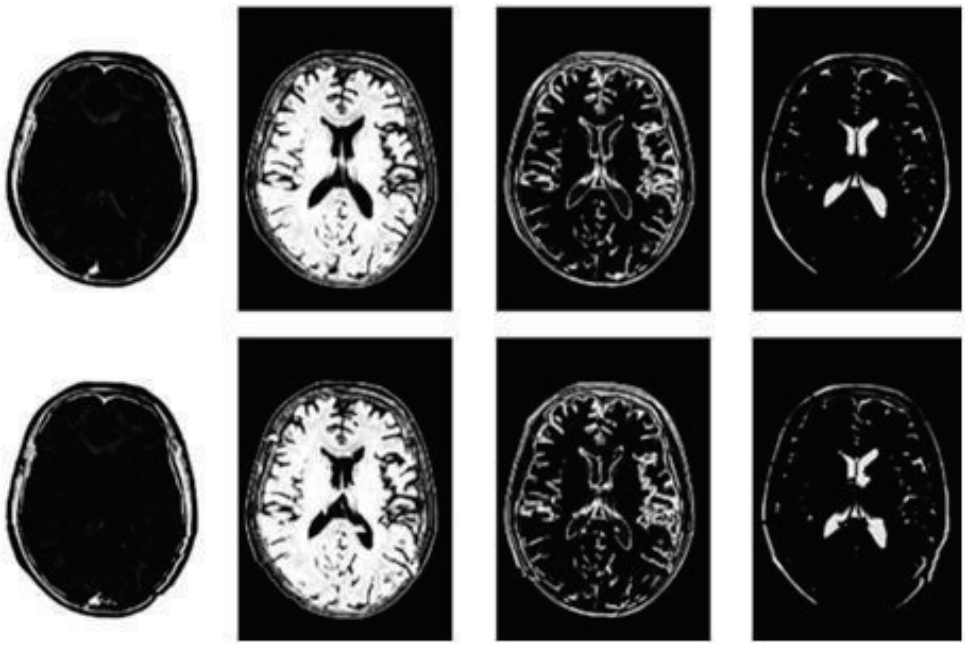


(b) Segmentation results for MRI2

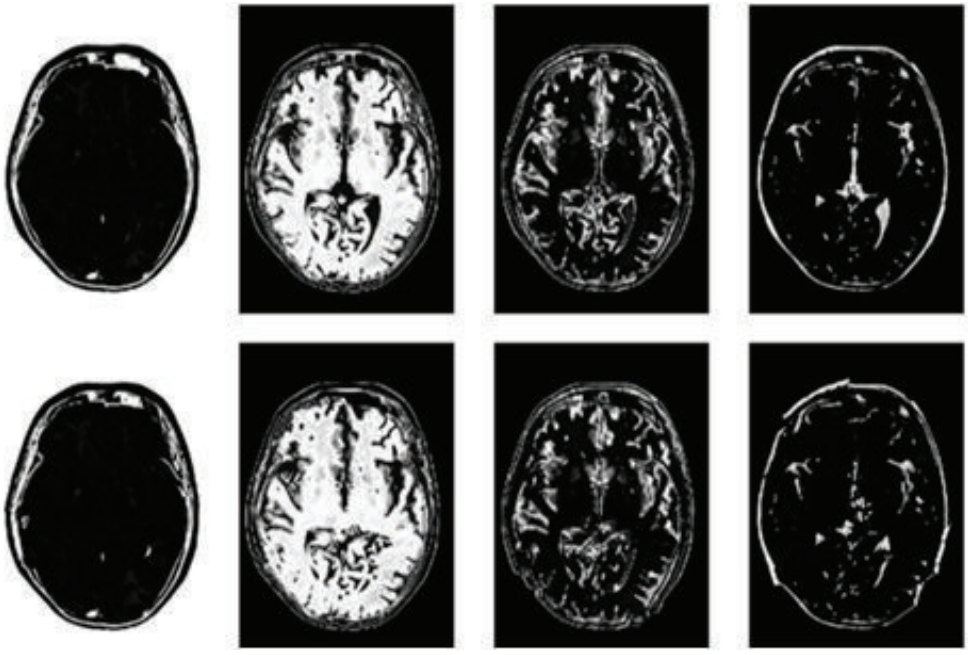
Figure 2: Segmentation results for the MRI1 and MRI2 brain images.

results for the improved HS algorithm and FCM. The second row shows the experimental results obtained using the original FCM algorithm. The segmentation results reveal that the fuzzy clustering method generated an oversegmentation phenomenon. For the actual MRI brain image segmentation effect, the algorithm proposed in this study performed better than the original FCM algorithm.

In Figure 3, the segmentation effect was affected by membership; the membership degree was associated with the initial $c = 4$. The first row shows the experimental results of the improved HS algorithm and FCM. The second row shows the experimental results obtained using the original FCM algorithm. The segmentation results reveal that the proposed MRI brain image segmentation effect obtained using the



(a) Segmentation results for MRI3



(b) Segmentation results for MRI4

Figure 3: Segmentation results for MRI3 and MRI4.

improved algorithm is better than that of the fuzzy clustering algorithm. In the fuzzy clustering algorithm, the initial value of uncertainty generated a local optimum algorithm, which affected the segmentation.

6. Conclusion

In this study, MRI brain image segmentation was achieved using the HS algorithm and the fuzzy clustering algorithm. The HS algorithm is more extensively used. However, given its drawbacks, the algorithm easily falls into the local optima. Thus, this study proposed an improved HS algorithm for MRI brain segmentation. Rough set theory was adopted to achieve an improved HS algorithm of an optimal harmonic database and important probability parameters for promoting harmony contraction convergence. Then, brain images were segmented using the fuzzy clustering algorithm. The initial value in the fuzzy clustering algorithm was random, which

affected the segmentation. Therefore, the optimal harmony value obtained by the improved HS algorithm was used as the initial value of the fuzzy clustering algorithm. The uncertainty in the initial value of the fuzzy clustering algorithm was avoided, thereby preventing the algorithm from falling into the local optimum. The simulation experiments showed that the proposed method produces better segmentation effects than those of the original fuzzy clustering algorithm.

Competing Interests

The authors declare that they have no competing interests related to this work.

Acknowledgments

This work is supported by Scientific Research Task in the Department of Education of Zhejiang (Y201328002) and Talent Starting Task of Wenzhou Medical University (QTJ11008).

References

[1] L. P. Clarke, R. P. Velthuizen, M. A. Camacho et al., "MRI segmentation: methods and applications," *Magnetic Resonance Imaging*, vol. 13, no. 3, pp. 343–368, 1995.

[2] Z. Liang, "Tissue classification and segmentation of MR images," *IEEE Engineering in Medicine and Biology Magazine*, vol. 12, no. 1, pp. 81–85, 1993.

[3] M. Vaidyanathan, L. P. Clarke, R. P. Velthuizen et al., "Comparison of supervised MRI segmentation methods for tumor volume determination during therapy," *Magnetic Resonance Imaging*, vol. 13, no. 5, pp. 719–728, 1995.

[4] L. P. Clarke, R. P. Velthuizen, S. Phuphanich, J. D. Schellenberg, J. A. Arrington, and M. Silbiger, "MRI: stability of three supervised segmentation techniques," *Magnetic Resonance Imaging*, vol. 11, no. 1, pp. 95–106, 1993.

[5] A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. D. A. Lotufo, "User-steered image segmentation paradigms: live wire and live lane," *Graphical Models and Image Processing*, vol. 60, no. 4, pp. 233–260, 1998.

[6] G. Lera and M. Pinzolas, "Neighborhood based Levenberg-Marquardt algorithm for neural Network training," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1200–1203, 2002.

[7] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[8] G. Liqun, G. Yanfeng, G. Yanfeng, and K. Zhi, "Adaptive harmonic particle swarm algorithm," *Control and Decision*, vol. 25, no. 7, pp. 1101–1104, 2010.

[9] W. S. Jang, H. I. Kang, and B. H. Lee, "Hybrid simplex-harmony search method for optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 4157–4164, IEEE, Hong Kong, June 2008.

[10] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[11] M. G. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.

[12] H.-Q. Li and L. Li, "A novel hybrid real-valued genetic algorithm for optimization problems," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '07)*, pp. 91–95, IEEE, Harbin, China, December 2007.

[13] L. Liang, C. Shichun, and Lin, "Improved harmony search algorithm and its application in slope stability analysis," *China Civil Engineering Journal*, vol. 39, no. 5, pp. 107–111, 2006.

[14] L. Li, Y. J. Wang, and Q. S. Wang, "New procedure for simulating arbitrary slip surface f soil slope in stability analysis," *Journal of Hydraulic Engineering*, vol. 16, no. 4, pp. 535–541, 2008.

[15] Y. M. Cheng, L. Li, T. Lansivaara, S. C. Chi, and Y. J. Sun, "An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis," *Engineering Optimization*, vol. 40, no. 2, pp. 95–115, 2008.

[16] H. Dong, Y. Bo, and M. Gao, "Improved harmony search for detection with photon density wave," in *Proceedings of the International Symposium on Photoelectronic Detection and Imaging: Related Technologies and Applications*, vol. 6625 of *Proceedings of SPIE*, pp. 23–26, Beijing, China, February 2008.

[17] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.

[18] Q. Zou, J. Guo, Y. Ju, M. Wu, X. Zeng, and Z. Hong, "Improving tRNAscan-SE annotation results via ensemble classifiers," *Molecular Informatics*, vol. 34, no. 11-12, pp. 761–770, 2015.

[19] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: ensemble classifiers with a clustering and dynamic selection strategy," *Neurocomputing*, vol. 123, pp. 424–435, 2014.

[20] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[21] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identificationof human microRNAs by incorporatinga high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.

[22] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1968.

[23] X. L. Xie and G. A. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[24] J. C. Bezdek, "Mathematical models for systematic and taxonomy," in *Proceedings of the Right International Conference on Numerical Taxonomy*, pp. 143–166, San Francisco, Calif, USA, 1975.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 3832176, 10 pages
http://dx.doi.org/10.1155/2016/3832176

Research Article

A Novel Peptide Binding Prediction Approach for HLA-DR Molecule Based on Sequence and Structural Information

Zhao Li,[1] Yilei Zhao,[1] Gaofeng Pan,[1] Jijun Tang,[1,2] and Fei Guo[1]

[1]*School of Computer Science and Technology, Tianjin University, 92 Weijin Road, Nankai District, Tianjin 300072, China*
[2]*School of Computational Science and Engineering, University of South Carolina, Columbia, SC, USA*

Correspondence should be addressed to Fei Guo; fguo@tju.edu.cn

Received 10 March 2016; Accepted 4 May 2016

Academic Editor: Yungang Xu

MHC molecule plays a key role in immunology, and the molecule binding reaction with peptide is an important prerequisite for T cell immunity induced. MHC II molecules do not have conserved residues, so they appear as open grooves. As a consequence, this will increase the difficulty in predicting MHC II molecules binding peptides. In this paper, we aim to propose a novel prediction method for MHC II molecules binding peptides. First, we calculate sequence similarity and structural similarity between different MHC II molecules. Then, we reorder pseudosequences according to descending similarity values and use a weight calculation formula to calculate new pocket profiles. Finally, we use three scoring functions to predict binding cores and evaluate the accuracy of prediction to judge performance of each scoring function. In the experiment, we set a parameter α in the weight formula. By changing α value, we can observe different performances of each scoring function. We compare our method with the best function to some popular prediction methods and ultimately find that our method outperforms them in identifying binding cores of HLA-DR molecules.

1. Introduction

Histocompatibility refers to the degree of antigenic similarity between the tissues of different individuals, which determines the acceptance or rejection of allografts. Transplantation antigen or histocompatibility antigen is the cause of rejection of allografts [1, 2]. MHC (Major Histocompatibility Complex) is present on the chromosome encoding a major histocompatibility antigen, mutual recognition between control cells, and the regulation of immune response.

MHC molecule plays a key role in immunology, and the molecule binding reaction with peptide is an important prerequisite for T cell immunity induced [2, 3]. By detecting a wide variety of microbial pathogens, the immune system protects host against diseases. Because of this, the binding prediction of MHC molecules with peptides has always been a hot topic in bioinformatics. Many researches in this field not only help us to understand the process of immune but also develop the work of vaccine design assisted by computers.

MHC genes produce two different types of molecules, which are MHC I molecules and MHC II molecules [1, 2]. MHC I molecules contain two separate polypeptide chains: the MHC α chain encoded by MHC genes and the MHC β chain encoded by non-MHC genes [4, 5]. MHC I class molecules are expressed in almost all eukaryotic cell surfaces, recognized by CD8+ cells. MHC II class molecules consist of two non-covalently linked polypeptide chains, namely, α chain and β chain. MHC II class molecules are expressed on antigen-presenting cells in general. Foreign MHC II antigens only capture and present on the surface of antigen-presenting cells (APC) TH cell [6]. After that, APC secretes large amounts of cytoplasm, activating cell invasion defensed behavior. Only the binding of antigen peptides and MHC II class molecules can activate CD4+ TH cells (helper T cells) [7]. Then, the activated TH cells would differentiate into effector cells and activate the immune response.

The structures of MHC I molecules and MHC II molecules slightly differ in the binding grooves [5]. Close grooves form on the binding of MHC I molecules and antigenic peptides. On the other hand, MHC II molecules do not have conserved residues, so they appear as open grooves. As a consequence, this will increase the difficulty in

predicting MHC II molecules binding peptides [7]. In this paper, we aim to solve more difficult problem of predicting MHC II binding peptides.

The pioneering and most popular pan-specific approach for MHC II binding prediction is the TEPITOPE method [8], and basic idea is the HLA-DR allele having identical pseudosequence. The same pocket will share the same quantitative profile. By using multiple instance learning, the MHCIIMulti method [9] can predict more than 500 HLA-DR molecules. Transforming each DRB allele into a pseudosequence with 21 amino acids and using the SMM-align method to identify binding cores, the NetMHCIIpan method [5] gets an accurate prediction by using an artificial neural network algorithm [10, 11]. Combining NN-align and NetMHCpan with NetMHCIIpan [9, 12], the MULTIPRED2 method [13–15] can get a perfect prediction for 1077 HLA-I and HLA-II alleles and 26 HLA supertypes.

In this paper, we propose a novel prediction method for predicting MHC II molecules binding peptides. First, we calculate sequence similarity and structural similarity between different MHC molecules [13, 16]. Then, we reorder pseudosequences according to descending similarity values and use a weight calculation formula to calculate new pocket profiles. Finally, we use three scoring functions to predict binding cores and evaluate the accuracy of prediction to judge performance of each scoring function [17, 18]. In the experiments, we set a parameter α in the weight formula. By changing α value, we can observe different performances of each of the scoring functions. We compare our method with the best function to some popular prediction methods and ultimately find that our method outperforms them in identifying binding cores of HLA-DR molecule [19]. The work would suggest a novel computational strategy for special protein identification instead of traditional machine learning based methods [20, 21].

2. Materials and Methods

2.1. Data Sets. We find 39 MHC molecules and peptides binding complexes from Protein Data Bank (http://www.rcsb.org/pdb/search/), which constitutes the data set used in this paper. In this data set, lengths are between 11 and 23, and we can find polypeptide-binding sites, namely, binding cores. Table 1 lists the details of these 39 MHC molecules and peptide binding complexes [14, 22, 23].

In Table 1, the first column is PDB ID of 39 complexes from PDB; the second column is the name of corresponding alleles from 39 complexes; the third column is the corresponding polypeptide sequences, in which the enlarged nine positions are the binding cores.

Table 1: Details of 39 MHC molecules and peptide binding complexes.

PDB ID	DRB allele	Peptide sequence
1AQD	DRB1*0101	VGSDWRFLRGYHQYA
1PYW	DRB1*0101	XFVKQNAAALX
1KLG	DRB1*0101	GELIGILNAAKVPAD
1KLU	DRB1*0101	GELIGTLNAAKVPAD
2FSE	DRB1*0101	AGFKGEQGPKGEPG
1SJH	DRB1*0101	PEVIPMFSALSEG
1SJE	DRB1*0101	PEVIPMFSALSEGATP
1T5W	DRB1*0101	AAYSDQATPLLLSPR
1T5X	DRB1*0101	AAYSDQATPLLLSPR
2IAN	DRB1*0101	GELIGTLNAAKVPAD
2IAM	DRB1*0101	GELIGILNAAKVPAD
2IPK	DRB1*0101	XPKWVKQNTLKLAT
1FYT	DRB1*0101	PKYVKQNTLKLAT
1R5I	DRB1*0101	PKYVKQNTLKLAT
1HXY	DRB1*0101	PKYVKQNTLKLAT
1JWM	DRB1*0101	PKYVKQNTLKLAT
1JWS	DRB1*0101	PKYVKQNTLKLAT
1JWU	DRB1*0101	PKYVKQNTLKLAT
1LO5	DRB1*0101	PKYVKQNTLKLAT
2ICW	DRB1*0101	PKYVKQNTLKLAT
2OJE	DRB1*0101	PKYVKQNTLKLAT
2G9H	DRB1*0101	PKYVKQNTLKLAT
1A6A	DRB1*0301	PVSKMRMATPLLMQA
1J8H	DRB1*0401	PKYVKQNTLKLAT
2SEB	DRB1*0401	AYMRADAAAGGA
1BX2	DRB1*1501	ENPVVHFFKNIVTPR
1YMM	DRB1*1501	ENPVVHFFKNIVTPRGGSGGGGG
1FV1	DRB5*0101	NPVVHFFKNIVTPRTPPPSQ
1H15	DRB5*0101	GGVYHFVKKHVHES
1ZGL	DRB5*0101	VHFFKNIVTPRTPGG
4E41	DRB1*0101	GELIGILNAAKVPAD
1DLH	DRB1*0101	PKYVKQNTLKLAT
1KG0	DRB1*0101	PKYVKQNTLKLAT
3L6F	DRB1*0101	APPAYEKLSAEQSPP
3PDO	DRB1*0101	KPVSKMRMATPLLMQALPM
3PGD	DRB1*0101	KMRMATPLLMQALPM
3S4S	DRB1*0101	PKYVKQNTLKLAT
3S5L	DRB1*0101	PKYVKQNTLKLAT
1HQR	DRB5*0101	VHFFKNIVTPRTP

2.2. Methods. There are thousands of allele variants in nature [2, 4]. It is absolutely impossible to measure the binding specificity one by one. Motivated by this perspective, we propose a new computational method to predict the binding specificity of peptides without any biochemical experiment, which combines the sequence and structural information of these known specificity-binding MHC molecules, as showed in Figure 1. We evaluate the method on all general HLA-DRB data sets, and results indicate that our method is close to the state-of-the-art technology and our approach can predict all sequence-known MHC molecules and cost little time, extending the prediction space compared with other time-consuming approaches.

2.3. Crucial Pockets relative to Binding Specificities of HLA-DR Molecules. We mainly use Position Specific Scoring Matrix (PSSM) [13, 24] in our approach, which is a popular technology in the problem of MHC binding. Roughly speaking,

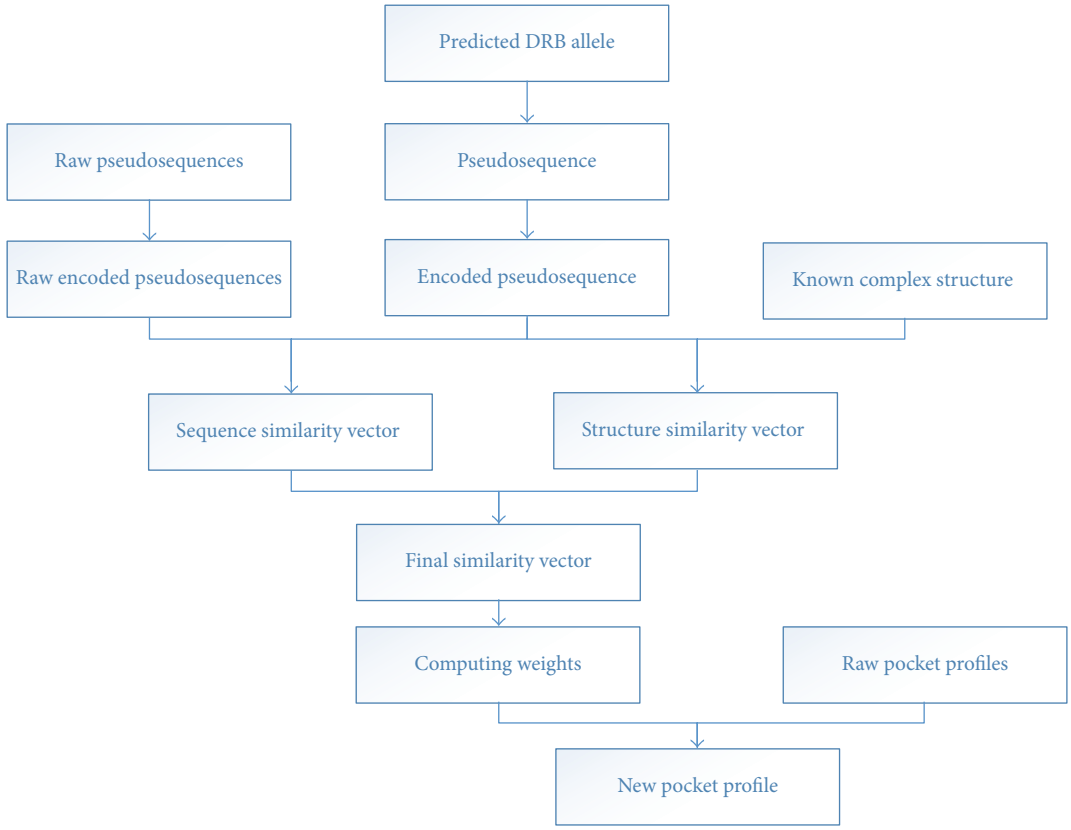


Figure 1: The architecture of our approach to MHC II and peptide binding problem.

there are nine amino acids in MHC binding cores, and each position is a specific pocket as showed in Table 2. We use PSSM to quantify the binding affinity between twenty basic amino acids with these nine pockets.

There are five anchor sites (1, 4, 6, 7, and 9) at the binding core for MHC II molecules, which determine the binding strength of peptides with MHC II molecules. Because site 1 of MHC II is consistent with different MHC II molecules and peptides, it is important to identify the precise quantification of its binding core in site 1, yet we use weights of four anchor sites (4, 6, 7, and 9) to define profiles. For other sites, the same approach, such as TEPITOPE, is to specify their quantitative profiles.

2.4. Computing Similarity between Different MHC Molecules

2.4.1. Sequence-Based Similarity. Sequence-based similarity can be calculated by alignment results. Here, pocket pseudosequences and associated profiles refer to raw pocket pseudosequences and raw pocket profiles, respectively. These raw pseudosequences are composed of several amino acids, whose associated residue indices are shown in Table 3. Eleven representative HLA-DR alleles are adopted to specify different profiles for anchor pockets 4, 6, 7, and 9. These eleven alleles are DRB1*0101, DRB1*0301, DRB1*0401, DRB1*0402, DRB1*0404, DRB1*0701, DRB1*0801, DRB1*1101, DRB1*1302, DRB1*1501, and DRB5*0101. If two alleles have identical pseudosequences in the same pocket, they will have identical profiles. For a given pocket, we collect all the different raw pocket pseudosequences into one set R^x, $R^x = \{r_1, r_2, \ldots, r_m\}$, and $|r_i| = n$, where $i = 1, 2, \ldots, m$, $x \in \{4, 6, 7, 9\}$, m is the number of unique pseudosequences, and n is the number of amino acids contained in a pseudosequence. Meanwhile, we collect all different raw profiles into one set P^x, $P^x = \{p_1, p_2, \ldots, p_m\}$, and $|p_i| = 20$, where $i = 1, 2, \ldots, m$. There is a one-to-one correspondence between p_i and r_i. We use BLOSUM to calculate the sequence similarity between different MHC molecules, defined as BLOSUM $= (S_q - S_i)$. Then, we can get encoded pseudosequence, which is a $20n$-dimensional real vector $V^x = \{V_1, V_2, \ldots, V_m\}$. We use Radial Basis Function (RBF) to measure the similarity between encoded predicted pseudosequences V_a and a raw encoded pseudosequence:

$$K_{\text{seq}}(V_a, V_i) = \text{BLOSUM}(V_a, V_i), \quad V_i \subseteq V^x. \tag{1}$$

2.4.2. Structure-Based Similarity. Using MHC II HLA-peptide complex structure from Protein Data Bank (PDB), we can get the residues 3D-coordinate of the pocket in each MHC molecule, $h(p_x, p_y, p_z)$. We define vector $H^x = \{h_1, h_2, \ldots, h_n\}$, where n is the number of amino acids in the pseudocontained sequence; meanwhile, we collect a set S^x, $S^x = \{H_1, H_2, \ldots, H_m\}$, m is the number of different pseudosequences, and there is also one-to-one correspondence between H_i and r_i.

Table 2: 30 HLA-complexes binding pockets.

PDB ID	Pocket 1	Pocket 2	Pocket 3	Pocket 4	Pocket 5	Pocket 6	Pocket 7	Pocket 8	Pocket 9
1AQD	82N 85V 86G	77T 78Y 81H 82N	78Y	13F 74A 78Y	13F 71R	11L	47Y 61W 67L 70Q 71R	60Y 61W	9W 57D 61W
1PYW	82N 85V 86G 89F	77T 78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	11L 28E 61W 71R	60Y 61W	57D 61W
1KLG	82N 85V	78Y 81H 82N	78Y	13F 71R 78Y	13F 71R	11L	61W	60Y 61W	57D 61W
2FSE	82N 85V 86G 89F	77T 78Y 82N		13F 28E 70Q 71R 74A 78Y	13F 71R	71R	28E 47Y 61W 67L 71R	61W	57D
1KLU	82N 85V	78Y 81H 82N		13F 71R 78Y	13F 71R	11L	61W	60Y 61W	57D 61W
1SJH	82N	78Y 81H 82N		13F 26L 70Q 71R 74A 78Y	71R	11L	61W	60Y 61W	57D 61W
1SJE	82N	78Y 81H 82N	78Y	13F 26L 70Q 71R 74A 78Y	71R	11L	61W	60Y 61W	57D 60Y 61W
1T5W	82N 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	61W 71R	60Y 61W	9W 57D 61W
1T5X	82N 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	71R	11L	61W 71R	61W	57D 61W
2IAN	82N 85V	78Y 81H 82N	78Y	13F 70Q 74A 78Y	13F 70Q 71R	11L	61W 71R	61W	57D 61W
2IPK	82N 85V 86G 89F	77T 78Y 81H 82N		13F 70Q 71R 74A 78Y	71R	11L	47Y 61W 67L 71R	60Y 61W	9W 57D 61W
1FYT	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	28E 47Y 61W 67L 71R	60Y 61W	9W 57D 61W
1R5I	82N 85V 86G 89F	77T 78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	70Q 71R	11L	47Y 61W 67L 71R	61W	9W 57D 61W
1HXY	82N 85V 86G 89F	78Y 81H 82N		13F 70Q 71R 74A 78Y	71R	11L	28E 47Y 61W 67L 71R	60Y 61W	9W 57D 61W
1JWM	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	71R	11L	28E 47Y 61W 67L 71R	61W	57D 61W
1JWS	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	47Y 61W 67L 71R	61W	9W 57D 61W
1JWU	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	28E 47Y 61W 67L 71R	61W	9W 57D 61W
1LO5	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 78Y	13F 71R	11L	47Y 61W 67L 71R	61W	9W 57D 60Y 61W
2ICW	82N 85V 86G 89F	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	13F 71R	11L	28E 47Y 61W 67L 71R	61W	9W 57D 61W
2OJE	82N 85V 86G	77T 78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	70Q 71R	11L	28E 47Y 61W 67L 71R	61W	9W 57D 61W
2G9H	82N 85V 86G 89F	77T 78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	71R	11L 13F	28E 47Y 61W 67L 71R	60Y 61W	9W 57D 61W
2IAM	82N	78Y 81H 82N	78Y	13F 70Q 71R 74A 78Y	70Q 71R	11L	61W 67L 71R	60Y 61W	57D 61W
1A6A	82N 85V 86V	77T 78Y 81H 82N	78Y	13S 26Y 74R 78Y	71K 74R	11S 30Y	30Y 47F 61W 67L 71K	61W	9E 30Y 57D 61W

Table 2: Continued.

PDB ID	Pocket 1	Pocket 2	Pocket 3	Pocket 4	Pocket 5	Pocket 6	Pocket 7	Pocket 8	Pocket 9
1J8H	82N 85V 86G 89F	77T 78Y 81H 82N	78Y	13H 26F 28D 70Q 74A 78Y	13H 70Q 71K	11V 13H 30Y	30Y 47Y 61W 67L	60Y 61W	37Y 57D 61W
2SEB	82N	77T 78Y 81H 82N		13H 26F 71K 78Y	13H 71K	30Y	30Y 47Y 61W	60Y 61W	61W
1BX2	82N 85V	77T 78Y 81H 82N	78Y	13H 26F 28D 70Q 74A 78Y	70Q	13R			57D 60Y 61W
1YMM	82N	77T 78Y 81H 82N	78Y	13R 26F 28D 70Q 74A 78Y	70Q	13R	61W 67I	61W	57D 61W
1FV1	82N 85V 86G 89F	78Y 81H 82N	78Y	13Y 71R 78Y	71R	13Y	61W 67L 71K	61W	57D
1H15	82N 89F	77T 78Y 81H 82N	78Y	13Y 71R 78Y	71R	11D 13Y 30D	61W		57D 60Y
1ZGL	82N 85V 89F	77T 78Y 81H 82N		13Y 26F 71R 78Y	13Y	13Y 28H 61W 71R	61W		57D 60Y 61W

Table 3: Important positions at the binding core for MHC II molecules.

Pocket	Important positions
Pocket 1	82 85 86 89
Pocket 2	77 78 81 82
Pocket 3	78
Pocket 4	11 13 26 28 70 71 74 78
Pocket 5	11 13 28 70 71 74
Pocket 6	11 13 28 70 71 74
Pocket 7	11 28 30 47 61 67 70 71
Pocket 8	60 61
Pocket 9	9 30 37 57 60 61

Next, we need to estimate the similarity of three-dimensional structures between a measured MHC molecule and five MHC molecules with known pseudosequence PSSM. Rigid transformation is to compare three-dimensional substructures of two proteins [25, 26].

Intuitively, we fix one of the structures, A, move (translation and rotation) the other structure, B, and find the best movement in three-dimensional space, with two atoms to the nearest structure. We calculate the Euclidean distance between two structures, defined as RMSD $= |C_q - C_i|$. We can get encoded pseudosequence $V^x = \{V_1, V_2, \ldots, V_m\}$ and calculate the similarity between 3D structures of encoded predicted pseudosequences V_a and a raw encoded pseudosequence:

$$K_{\text{spa}}(V_a, V_i) = \text{RMSD}(V_a, V_i), \quad V_i \subseteq V^x. \tag{2}$$

2.4.3. Overall Similarity. After that, we have obtained sequence similarity and structural similarity. We calculate final similarity score functions according to the following three formulas:

$$K_1(V_a, V_i) = \sqrt{\frac{K_{\text{seq}}(V_a, V_i)^2 + K_{\text{spa}}(V_a, V_i)^2}{2}},$$
$$K_2(V_a, V_i) = \frac{K_{\text{seq}}(V_a, V_i) + K_{\text{spa}}(V_a, V_i)}{2}, \tag{3}$$
$$K_3(V_a, V_i) = \sqrt{K_{\text{seq}}(V_a, V_i) + K_{\text{spa}}(V_a, V_i)}.$$

2.5. Weights Calculation for New Pocket Profiles. We reorder all pseudosequences according to descending similarity values and use a weight calculation formula to calculate new pocket profiles. A new pocket profile is generated as a weighted average over m raw pocket profiles in P^x. Next, we use the gamma distribution to generate the weights. The gamma PDF distribution is defined as follows:

$$g(x; k, \theta) = \frac{1}{\theta^k} \frac{1}{\gamma(k)} x^{k-1} e^{-x/\theta}, \tag{4}$$

where $x > 0$ and $k, \theta > 0$, and $\gamma(k)$ denotes the gamma function.

The weight distribution is generated to discretize the gamma PDF as follows:

$$G(X = i) = \frac{1}{\theta^k} \frac{1}{\gamma(k)} i^{k-1} e^{-i/\theta}, \quad i = 1, 2, \ldots, m, \tag{5}$$

where m is the dimension of the weights and k and θ are the shape and scale parameters, respectively. The gamma distribution generates the weight vector to give a higher weight for more similarity pseudosequences.

After normalizing, the weight vector is defined as follows:

$$P(X = i) = \frac{[G(X = i)]^\alpha}{\sum_{k=1}^{m} [G(X = k)]^\alpha}, \quad i = 1, 2, \ldots, m. \tag{6}$$

Given a predicted DRB allele a, let $K_a = (K_{a1}, K_{a2}, \ldots, K_{am})$, where $K_{ai} = K(V_a, V_i)$, $V_i \in V^x$, and α is a positive number and enhances the weight vector to protect the outstanding contribution of most similarity pseudosequences. Associated raw pocket profiles are $P_x = \{P_1, P_2, \ldots, P_m\}$. Elements of K_a are sorted in descending order, and the reordered vector of K_a is denoted as $\widetilde{K_a} = (\widetilde{K_{a1}}, \widetilde{K_{a2}}, \ldots, \widetilde{K_{am}})$. The corresponding weight vector is denoted as $W = (\omega_1, \omega_2, \ldots, \omega_m)$. We denote pocket profiles associated with the reordered vector $\widetilde{K_a}$ as $\widetilde{P}^x$, $\widetilde{P}^x = \{\widetilde{P}_1, \widetilde{P}_2, \ldots, \widetilde{P}_m\}$. We define the pocket profile for allele a as follows:

$$\widetilde{P}_a^x = \omega_1 \widetilde{P}_1 + \omega_2 \widetilde{P}_2 + \cdots + \omega_m \widetilde{P}_m, \tag{7}$$

where $x \in \{4, 6, 7, 9\}$.

3. Result

First, we design an experiment to choose appropriate scoring function to combine sequence similarity and structural similarity. Then, we compare with other state-of-the-art technologies, which are TEPITOPE, MultiRTA, NetMHCIIpan-2.0, and NetMHCIIpan-1.0. The result indicates that our approach can obtain better prediction and effectively extend current prediction methods. Finally, we test on more data sets.

3.1. Evaluation of Different Scoring Functions. Here, we use 30 of 39 MHC molecules and peptide complexes as test set and get the appropriate scoring functions as showed above. The value of the parameter α is set to 1, 2, 3, 4, 5, 10, 15, and 20, followed by results shown in Figure 2. We find that no significant changes can be found by $K_1(V_a, V_i)$; for $K_2(V_a, V_i)$ and $K_3(V_a, V_i)$, when $\alpha = 1$ prediction error number is 10 and 9 and when $\alpha = 3$ prediction errors reduced to 8, we set the value of α to 3. Comparing these three functions, the least numbers of errors by three functions are 4, 8, and 8. Details are shown in Tables S1, S2, and S3, in the Supplementary Material available online at http://dx.doi.org/10.1155/2016/3832176.

3.2. Compared with Conventional Well-Known Methods. From the above experimental results, $K_1(V_a, V_i)$ obtains the most accurate prediction, so we will select $K_1(V_a, V_i)$ with $\alpha = 3$ as our final approach. We compare our current

Table 4: Comparison of our binding prediction with other approaches. The 5th column is the result of our method, and 6th to 8th columns are results of TEPITOPE, MultiRTA, and NetMHCIIpan. The bold cell means one error.

PDB ID	Allele	Peptide	Core	Ours	TEPITOPE	MultiRTA	NetMHCIIpan-2.0
1AQD	DRB1*0101	VGSDWRFLRGYHQYA	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ	WRFLRGYHQ
1PYW	DRB1*0101	XFVKQNAAALX	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL	FVKQNAAAL
1KLG	DRB1*0101	GELIGILNAAKVPAD	IGILNAAKV	IGILNAAKV	IGILNAAKV	IGILNAAKV	**LIGILNAAK**
2FSE	DRB1*0101	GELIGTLNAAKVPAD	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV
1KLU	DRB1*0101	AGFKGEQGPKGEPG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG	FKGEQGPKG
1SJH	DRB1*0101	PEVIPMFSALSEG	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS
1SJE	DRB1*0101	PEVIPMFSALSEGATP	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS	VIPMFSALS
1T5W	DRB1*0101	AAYSDQATPLLLSPR	YSDQATPLL	**SDQATPLLL**	YSDQATPLL	**SDQATPLLL**	YSDQATPLL
1T5X	DRB1*0101	AAYSDQATPLLLSPR	YSDQATPLL	**SDQATPLLL**	YSDQATPLL	**SDQATPLLL**	YSDQATPLL
2IAN	DRB1*0101	GELIGTLNAAKVPAD	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV	IGTLNAAKV
2IPK	DRB1*0101	GELIGILNAAKVPAD	IGILNAAKV	IGILNAAKV	IGILNAAKV	IGILNAAKV	**LIGILNAAK**
1FYT	DRB1*0101	XPKWVKQNTLKLAT	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL	WVKQNTLKL
1R5I	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1HXY	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1JWM	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1JWS	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1JWU	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1LO5	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
2ICW	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
2OJE	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
2G9H	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
2IAM	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
1A6A	DRB1*0301	PVSKMRMATPLLMQA	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM	MRMATPLLM
1J8H	DRB1*0401	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL	YVKQNTLKL
2SEB	DRB1*0401	AYMRADAAAGGA	MRADAAAGG	MRADAAAGG	MRADAAAGG	MRADAAAGG	**YMRADAAAG**
1BX2	DRB1*1501	ENPVVHFFKNIVTPR	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	**VVHFFKNIV**
1YMM	DRB1*1501	ENPVVHFFKNIVTPRGGSGGGGG	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT	VHFFKNIVT
1FV1	DRB5*0101	NPVVHFFKNIVTPRTPPPSQ	FKNIVTPRT	**KNIVTPRTP**	FKNIVTPRT	**VHFFKNIVT**	**FFKNIVTPR**
1H15	DRB5*0101	GGVYHFVKKHVHES	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH	YHFVKKHVH
1ZGL	DRB5*0101	VHFFKNIVTPRTPGG	FKNIVTPRT	**KNIVTPRTP**	FKNIVTPRT	**VHFFKNIVT**	**FFKNIVTPR**
Results				4 errors	0 errors	4 errors	6 errors

Table 5: Other prediction results of nine MHC molecules. This table shows the prediction result of our method on 9 MHC molecules. The 5th column is the result. There is only one error result, which is shown using bold font.

PDB ID	Allele	Peptide	Core	Ours
4E41	DRB1*0101	GELIGILNAAKVPAD	IGILNAAKV	IGILNAAKV
1DLH	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL
1KG0	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL
3L6F	DRB1*0101	APPAYEKLSAEQSPP	YEKLSAEQS	YEKLSAEQS
3PDO	DRB1*0101	KPVSKMRMATPLLMQALPM	MRMATPLLM	**KMRMATPLL**
3PGD	DRB1*0101	KMRMATPLLMQALPM	MRMATPLLM	MRMATPLLM
3S4S	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL
3S5L	DRB1*0101	PKYVKQNTLKLAT	YVKQNTLKL	YVKQNTLKL
1HQR	DRB5*0101	VHFFKNIVTPRTP	FKNIVTPRT	FKNIVTPRT
Results				1 error

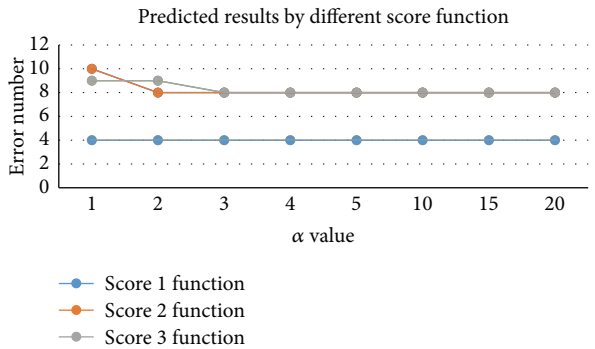


Figure 2: Predicted results by different score functions. x-axis represents different α values, and the y-axis refers to predicted results of different score functions.

prediction results with conventional well-known methods TEPITOPE [23], MultiRTA [13], NetMHCIIpan-2.0 [12], and NetMHCIIpan-1.0 [12], and these results are shown in Table 4.

TEPITOPE is a relatively early method and is one of the most popular methods for predicting MHC II binding molecules. The basic idea is that if two HLA-DR alleles have the same pseudorandom sequence in the same pocket, they share the same number of profiles. Through multiple instances, MHCIIMulti has predicted over 500 HLA-DR molecules. NetMHCIIpan firstly converts each of the DRB alleles into a pseudorandom sequence of 21 amino acids, then uses the SMM-align method to identify binding residues in the peptide chain and the core side, and finally uses artificial neural network to train the model. MultiRTA makes prediction on HLA-DR and HLA-DP molecules. By thermodynamic method, it calculates a peptide chain and all other residues to predict the average binding affinity of binding strength and the introduction of standardization constraints to avoid overfitting. MULTIPRED2 can predict 1077 HLA-I and HLA-II genes and 26 HLA supertypes. Details are as shown in Figure 3. Our method obtains 4 errors; however, TEPITOPE, MultiRTA, NetMHCIIpan-2.0, and NetMHCIIpan-1.0 get the numbers of errors as 0, 4, 6, and 3, respectively. Because now we only find five MHC II molecules with three-dimensional structural information, we use the scoring matrix with only 5 MHC II molecules. If the three-dimensional structural information of MHC II molecules can be extended to all of the 11 MHC II molecules, our predictions will be more accurate. From the current view, our approach has reached a higher level of prediction.

3.3. Other Prediction Results. When compared with other methods on the above experiments, we only use 30 of 39 MHC molecules and peptide complexes as test set. In this section, we test on the remaining nine MHC molecules. In this experiment, we choose $K_1(V_a, V_i)$ and set the parameter $\alpha = 3$. As seen in Table 5, eight of nine predictions are accurate. Therefore, our approach produces a considerably great performance.

4. Conclusion

In this paper, we try to solve the problem of predicting MHC II binding peptides with a novel metric and strategy. Sequence similarity and structural similarity between different MHC molecules are calculated to reorder pseudosequences according to descending similarity, and then a weight calculation formula is used to calculate new pocket profiles. Finally, we use three scoring functions to predict binding cores and evaluate the accuracy of prediction to judge performance of each scoring function. In the experiment, we set a parameter α in the weight formula. By changing α value, we can observe different performances of each scoring function. Then, we compare our method with the best function to some popular prediction methods and ultimately find that our method outperforms them in identifying binding cores of HLA-DR molecules.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

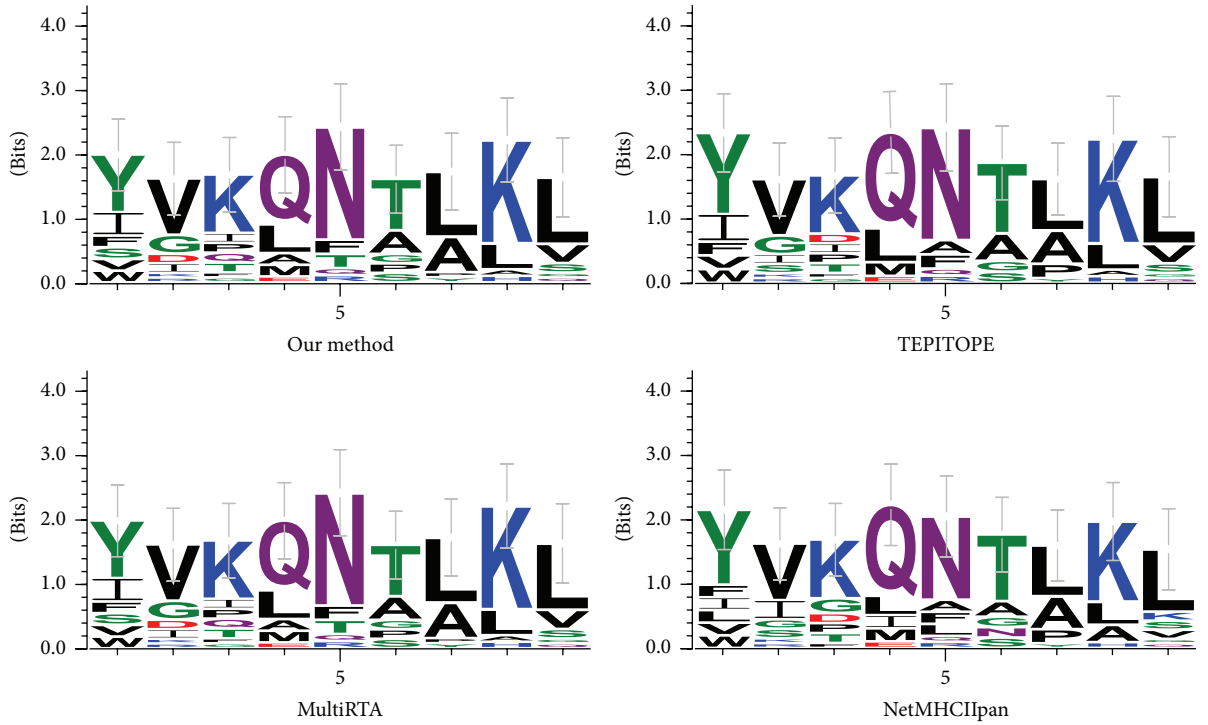


Figure 3: Comparison of different methods by sequence logos of peptides on HLA-DRB1*0101.

Acknowledgments

This work is supported by a grant from the National Science Foundation of China (NSFC 61402326) and Peiyang Scholar Program of Tianjin University (no. 2016XRG-0009).

References

[1] R. M. Zinkernagel and P. C. Doherty, "Restriction of in vitro T cell-mediated cytotoxicity in lymphocytic choriomeningitis within a syngeneic or semiallogeneic system," *Nature*, vol. 248, no. 5450, pp. 701–702, 1974.

[2] P. I. Terasaki, "A brief history of HLA," *Immunologic research*, vol. 38, no. 1–3, pp. 139–148, 2007.

[3] K. Maenaka and E. Y. Jones, "MHC superfamily structure and the immune system," *Current Opinion in Structural Biology*, vol. 9, no. 6, pp. 745–753, 1999.

[4] J. Robinson, M. J. Waller, S. C. Fail et al., "The IMGT/HLA database," *Nucleic Acids Research*, vol. 37, supplement 1, pp. D1013–D1017, 2009.

[5] M. Nielsen, C. Lundegaard, T. Blicher et al., "Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan," *PLoS Computational Biology*, vol. 4, no. 7, article e1000107, 2008.

[6] H.-G. Rammensee, J. Bachmann, N. P. N. Emmerich, O. A. Bachor, and S. Stevanović, "SYFPEITHI: database for MHC ligands and peptide motifs," *Immunogenetics*, vol. 50, no. 3, pp. 213–219, 1999.

[7] R. N. Germain, "MHC-dependent antigen processing and peptide presentation: providing ligands for T lymphocyte activation," *Cell*, vol. 76, no. 2, pp. 287–299, 1994.

[8] T. Sturniolo, E. Bono, J. Ding et al., "Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices," *Nature Biotechnology*, vol. 17, no. 6, pp. 555–561, 1999.

[9] N. Pfeifer and O. Kohlbacher, "Multiple instance learning allows MHC class II epitope predictions across alleles," *Algorithms in Bioinformatics*, vol. 5251, pp. 210–221, 2008.

[10] T. J. Kindt, R. A. Goldsby, B. A. Osborne, and J. Kuby, *Kuby Immunology*, WH Freeman & Company, New York, NY, USA, 2007.

[11] A. Sette, L. Adorini, E. Appella et al., "Structural requirements for the interaction between peptide antigens and I-Ed molecules," *Journal of Immunology*, vol. 143, no. 10, pp. 3289–3294, 1989.

[12] M. Nielsen, C. Lundegaard, T. Blicher et al., "Quantitative predictions of peptide binding to any HLA-DR molecule of known sequence: NetMHCIIpan," *PLoS Computational Biology*, vol. 4, no. 7, Article ID e1000107, 2008.

[13] A. J. Bordner and H. D. Mittelmann, "MultiRTA: a simple yet reliable method for predicting peptide binding affinities for multiple class II MHC allotypes," *BMC Bioinformatics*, vol. 11, article 482, 2010.

[14] P. A. Reche, H. Zhang, J.-P. Glutting, and E. L. Reinherz, "EPIMHC: a curated database of MHC-binding peptides for customized computational vaccinology," *Bioinformatics*, vol. 21, no. 9, pp. 2140–2141, 2005.

[15] P. A. Reche, J.-P. Glutting, H. Zhang, and E. L. Reinherz, "Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles," *Immunogenetics*, vol. 56, no. 6, pp. 405–419, 2004.

[16] A. Sette, L. Adorini, E. Appella et al., "Structural requirements for the interaction between peptide antigens and I-Ed

molecules," *The Journal of Immunology*, vol. 143, no. 10, pp. 3289–3294, 1989.

[17] M. Nielsen, S. Justesen, O. Lund, C. Lundegaard, and S. Buus, "NetMHCIIpan-2.0—improved pan-specific HLA-DR predictions using a novel concurrent alignment and weight optimization training procedure," *Immunome Research*, vol. 6, no. 1, article 9, 2010.

[18] M. Nielsen and O. Lund, "NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction," *BMC Bioinformatics*, vol. 10, article 296, 2009.

[19] G. L. Zhang, D. S. DeLuca, D. B. Keskin et al., "MULTIPRED2: a computational system for large-scale identification of peptides predicted to bind to HLA supertypes and alleles," *Journal of Immunological Methods*, vol. 374, no. 1-2, pp. 53–61, 2011.

[20] X.-Y. Cheng, W.-J. Huang, S.-C. Hu et al., "A global characterization and identification of multifunctional enzymes," *PLoS ONE*, vol. 7, no. 6, Article ID e38979, 2012.

[21] Q. Zou, Z. Wang, X. Guan, B. Liu, Y. Wu, and Z. Lin, "An approach for identifying cytokines based on a novel ensemble classifier," *BioMed Research International*, vol. 2013, Article ID 686090, 11 pages, 2013.

[22] W. Shen, S. Zhang, and H. Wong, "An effective and effecient peptide binding prediction approach for a broad set of HLA-DR molecules based on ordered weighted averaging of binding pocket profiles," *Proteome Science*, vol. 11, p. S15, 2013.

[23] L. Zhang, Y. Chen, H.-S. Wong, S. Zhou, H. Mamitsuka, and S. Zhu, "TEPITOPEpan: extending TEPITOPE for peptide binding prediction covering over 700 HLA-DR molecules," *PLoS ONE*, vol. 7, no. 2, Article ID e30483, 2012.

[24] P. A. Reche and E. L. Reinherz, "Prediction of peptide-MHC binding using profiles," *Methods in Molecular Biology*, vol. 409, pp. 185–200, 2007.

[25] F. Guo, S. C. Li, L. Wang, and D. Zhu, "Protein-protein binding site identification by enumerating the configurations," *BMC Bioinformatics*, vol. 13, article 158, 2012.

[26] F. Guo, S. C. Li, and L. Wang, "Protein-protein binding sites prediction by 3D structural similarities," *Journal of Chemical Information and Modeling*, vol. 51, no. 12, pp. 3287–3294, 2011.

Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 9246280, 8 pages
http://dx.doi.org/10.1155/2016/9246280

Research Article

Analysis and Classification of Stride Patterns Associated with Children Development Using Gait Signal Dynamics Parameters and Ensemble Learning Algorithms

Meihong Wu,[1] Lifang Liao,[1] Xin Luo,[1] Xiaoquan Ye,[1] Yuchen Yao,[1] Pinnan Chen,[1] Lei Shi,[2] Hui Huang,[3] and Yunfeng Wu[1]

[1]*School of Information Science and Technology, Xiamen University, 422 Si Ming South Road, Xiamen, Fujian 361005, China*
[2]*Department of Orthopedics, Zhongshan Hospital, Xiamen University, 201 Hubin South Road, Xiamen, Fujian 361004, China*
[3]*Department of Rehabilitation, Zhongshan Hospital, Xiamen University, 201 Hubin South Road, Xiamen, Fujian 361004, China*

Correspondence should be addressed to Yunfeng Wu; y.wu@ieee.org

Received 23 January 2016; Accepted 11 February 2016

Academic Editor: Yungang Xu

Measuring stride variability and dynamics in children is useful for the quantitative study of gait maturation and neuromotor development in childhood and adolescence. In this paper, we computed the sample entropy (SampEn) and average stride interval (ASI) parameters to quantify the stride series of 50 gender-matched children participants in three age groups. We also normalized the SampEn and ASI values by leg length and body mass for each participant, respectively. Results show that the original and normalized SampEn values consistently decrease over the significance level of the Mann-Whitney U test ($p < 0.01$) in children of 3–14 years old, which indicates the stride irregularity has been significantly ameliorated with the body growth. The original and normalized ASI values are also significantly changing when comparing between any two groups of young (aged 3–5 years), middle (aged 6–8 years), and elder (aged 10–14 years) children. Such results suggest that healthy children may better modulate their gait cadence rhythm with the development of their musculoskeletal and neurological systems. In addition, the AdaBoost.M2 and Bagging algorithms were used to effectively distinguish the children's gait patterns. These ensemble learning algorithms both provided excellent gait classification results in terms of overall accuracy (≥90%), recall (≥0.8), and precision (≥0.8077).

1. Introduction

An infant commonly begins to crawl after 9 months and then learns how to walk with voluntary postural control at about one year after birth [1, 2]. During the physical growth in adolescence, the locomotor control and postural coordination of children become mature, in correspondence with the development of the central nervous system and musculoskeletal system [3]. According to Hillman et al. [4], temporal and spatial parameters of children's gait become relatively mature until 4 years old. The study of Chester et al. [5] suggested that adult-like kinetic patterns for the hip and knee are almost achieved in children by 5 years of age, whereas the ankle joint patterns remain premature until 9 years old. Menkveld et al. [6] analyzed the temporal gait parameters of a few children subjects from 7 to 16 years of age and reported that the stride patterns are more stable, but the gait modulation function still continues to improve in adolescence.

Because human locomotion functions are regulated by the neuromotor and muscular functions, immature neurological control or inconsistent muscle contractions would result in erratic body movement behaviors with irregular rhythm [7, 8]. The immature motor control of young children causes higher degree of variability in stride time (the duration from initial contact of one foot to the succeeding contact of the same foot) [4], such that the stride series would present large fluctuations or dynamic complexity. Recent studies [9, 10] emphasized how to measure the gait unsteadiness and subtle fluctuations in the course of motor skill development. Hausdorff et al. [9] used the coefficient of variation parameter

and fractal analysis tools to compute the fluctuation magnitude and fractal properties of stride series of young children. Their results suggested that the stride variability significantly decreases and also exhibits long-range fractal correlations in adolescents [9].

Recently, signal irregularity analysis of physiological systems based on entropy and novel statistical measures has received extensive attractions in the research community [11, 12]. Huang et al. [13] and Wei et al. [14] measured the sample entropy (SampEn) parameter of electroencephalogram (EEG) intrinsic mode functions decomposed by the multivariate empirical mode decomposition and applied an artificial neural network trained by the back-propagation algorithm to detect the particular signal patterns related to anesthesia. Sharma et al. [15] extracted the Shannon entropy, Renyi entropy, approximate entropy, and SampEn features from the EEG signal components derived from the empirical mode decomposition algorithm and then employed the least-squares support vector machine to discriminate the focal EEG signals. In our previous studies [16, 17], we applied the nonparametric statistical methods to establish the probability density models of stride series for the adolescents at different ages.

As reported by Shumway-Cook and Woollacott [18], stride dynamics analysis may provide important indices related to the development of neuromuscular control in children. Analysis of gait patterns of young children can assist physiologists to better understand the course of gait maturation. Further quantitative studies require more advanced computational and mathematical tools to characterize the progress of gait development. The motivation of our study is to compute the SampEn and average stride interval (ASI) features to quantify the changes of gait dynamics in the stride time series of children associated with the adolescent development. The AdaBoost.M2 and Bagging ensemble learning algorithms were used to effectively perform the gait pattern classifications for the children participants in different age groups.

2. Material and Methods

2.1. Gait Data Description. The gait data set was obtained from a PhysioNet database provided by Hausdorff et al. [9], for public research of gait maturation. A total of 50 healthy children (equal number of boys and girls) aged from 3 to 14 years were recruited from the local community in Boston, MA, USA, to participate in the gait data acquisition experiments [9]. None of these children was prematurely born or suffering from any of musculoskeletal, neurological, or cardiovascular disease. In order to investigate the gait development of children with aging, the children participants were categorized into three age groups: young children of 3–5 years old (14 subjects: 6 boys and 8 girls), middle children of 6–8 years old (21 subjects: 10 boys and 11 girls), and elder children of 10–14 years old (15 subjects: 9 boys and 6 girls). Statistics of body mass and leg length of the children participants are listed in Table 1. The children's parents provided their informed and written consent letters as approved by Harvard Medical School and completed the questionnaire sheets to declare the medical history of their kids [9].

Table 1: Statistics of body mass and leg length of the children in the young, middle, and elder age groups, respectively. Values are expressed as mean ± standard deviation.

Age groups	Body mass (kg)	Leg length (m)
Young (3–5 years old)	18.01 ± 2.98	0.55 ± 0.04
Middle (6–8 years old)	25.31 ± 4.02	0.65 ± 0.05
Elder (10–14 years old)	42.61 ± 9.21	0.79 ± 0.07

Each participant was asked to walk with his or her comfortable pace for 8 min, around a 400 m running track outdoors [9]. An investigator followed up each child during the gait data acquisition experiments. The contact force of the body on level ground was measured by two ultrathin pressure-sensitive sensors, which were placed in the right shoe of each child (one underneath the ball of the foot and the other underneath the heel) [19].

The voltage signals of force underneath the right foot were amplified with a portable signal acquisition board (dimensions: $5.5 \times 2 \times 9$ cm; weight: 100 g) worn on the ankle cuff of each child. The signal data were sampled at 300 Hz and digitized by a built-in analog-to-digital converter with a resolution of 12 bits per sample. The series of gait cycle durations (the time from heel strike to heel strike of the same foot) or stride intervals (in seconds) were estimated with the algorithm proposed by Hausdorff et al. [19].

Because the gait speed and other phase parameters are often altered by the accelerating or decelerating movements when the subject starts or stops walking, it is necessary to eliminate the start-up or ending effects of walking posture in the gait data. In the present work, the data samples of the stride interval series recorded in the first 60 s and the last 5 s were removed, respectively, which was the same as implemented in the previous related studies [9, 17]. The stride outliers whose amplitude values were larger or smaller than three times standard deviations of the median of each stride interval series were detected and removed by a median filter [17, 20].

2.2. Gait Signal Dynamics Quantification

2.2.1. Sample Entropy (SampEn). SampEn has been widely used to measure the degree of regularity in complex physiological signals, by calculating the negative natural logarithm of the estimated conditional probability of self-similarity signal segments (epochs). A lower value of SampEn indicates more similar epochs occurring in the time series. Considering a gait rhythm time series $\{x(l)\}$ of length L, we may define a template that contains a series of k consecutive signal elements as $\mathbf{x}_m^k = [x(m), x(m+1), \ldots, x(m+k-1)]$, where k is commonly known as the embedding dimension. The similar elements included in two templates are measured by the absolute maximum difference as

$$d\left[\mathbf{x}_m^k, \mathbf{x}_n^k\right] = \max_{0 \le q \le k-1} \left|x\left(m+q\right) - x\left(n+q\right)\right|. \quad (1)$$

Let $B_m(\theta)$ denote the total number of n, $n = 1, 2, \ldots, L - k + 1$ ($n \neq m$), which meets the requirement $d[\mathbf{x}_m^k, \mathbf{x}_n^k] \leq \theta$, where θ denotes the tolerance threshold for accepting the similar templates. The probability of the similar templates within the tolerance level θ is then defined as

$$B_m^k(\theta) = \frac{B_m(\theta)}{L - k + 1}. \tag{2}$$

Then, we can compute the average number of the total similar templates as

$$B^k(\theta) = \frac{1}{L - k + 1} \sum_{m=1}^{L-k+1} B_m^k(\theta). \tag{3}$$

Similarly, by increasing the embedding dimension up to $k + 1$, we may compute the corresponding probability of the similar templates, $A_m^{k+1}(\theta)$, as

$$A_m^{k+1}(\theta) = \frac{A_m(\theta)}{L - k}, \tag{4}$$

where $A_m(\theta)$ satisfies $d[\mathbf{x}_m^{k+1}, \mathbf{x}_n^{k+1}] \leq \theta$, for $n = 1, 2, \ldots, L - k$ ($n \neq m$). The average of all matching similar templates with the embedding dimension $k + 1$ is computed as

$$A^{k+1}(\theta) = \frac{1}{L - k} \sum_{m=1}^{L-k} A_m^{k+1}(\theta). \tag{5}$$

Finally, the SampEn is defined as

$$\text{SampEn}(k, \theta, L) = -\ln\left[\frac{A^{k+1}(\theta)}{B^k(\theta)}\right]. \tag{6}$$

In the present study, the SampEn method was used to probe the self-similarity gait signal epochs by estimating the similar-matching templates in stride series. The length of stride series $L = 350$ is identical for every single child. The SampEn embedding dimension is set to be $k = 2$. The optimal tolerance parameter of the SampEn model, $\theta = 0.05$, was derived with the lowest p value results of the Mann-Whitney U test (significance level: $p < 0.01$). Thus, the SampEn(2, 0.05, 350) model was selected to quantify the gait regularity in the children's stride series.

2.2.2. Average Stride Interval (ASI). ASI is referred to as the mean of stride interval during a period of gait monitoring [17]. In the present work, we computed the ASI value based on the probability density function (PDF) of stride interval, as a dominant gait feature to represent the average duration of a stride for each child participant. The PDF of stride interval provides a continuous probability distribution estimate for a number of stride observations. For a given stride time series $\{x(l)\}$, $l = 1, 2, \ldots, L$, the PDF of stride interval, $\widehat{p}(g)$, can be established by using the Parzen-window method [17, 20, 21] as

$$\widehat{p}(x) = \frac{1}{L} \sum_{l=1}^{L} \kappa[x - x(l)], \tag{7}$$

where $\kappa(\cdot)$ denotes a nonnegative kernel function, which integrates to unity; that is, $\int_{-\infty}^{\infty} \kappa(x)dx = 1$.

In our study, the prevailing Gaussian kernel function was applied to estimate the PDF of stride interval; that is,

$$\kappa[x - x(l)] = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{-[x - x(l)]^2}{2\sigma^2}\right\}, \tag{8}$$

where σ denotes the spread parameter of the Gaussian function. It is worth noting that the center of the Gaussian function is located at the amplitude of each stride observation $x(l)$, and the spread parameter σ determines the Gaussian kernel window width [20].

The effectiveness of nonparametric PDF estimate by means of the Parzen-window method depends on the optimal choice of the spread parameter [22]. In order to select the best spread parameter, the estimated PDF was compared with the histogram of stride interval with the same resolution; that is, the discrete scale of the stride PDF is equal to the number of histogram bins. In the searching range of $[0.001, 0.1]$, with an increment step of 0.001, the spread parameter of 0.01 that matched the minimization criterion of the mean-squared error between the Parzen-window PDF and the histogram of stride interval was chosen as the optimal value [22]. Then, the ASI value can be calculated as the mean of stride interval based on the estimated Parzen-window PDF [17] as

$$\text{ASI} = \int_{-\infty}^{\infty} x\widehat{p}(x)\, dx. \tag{9}$$

We computed the ASI values for all 50 children participants and also applied the Mann-Whitney U test (implemented with IBM SPSS Statistics, Version 20) to study the statistical differences of ASI among three different age groups (significance level: $p < 0.01$).

2.3. Ensemble Learning Algorithms. With the SampEn and ASI features obtained, we may perform effective gait pattern classifications for further analysis. For two decades, multiple learner systems trained by advanced ensemble learning algorithms have received extensive attentions in the machine learning community [23–26]. Ensemble learning is also referred to as committee machine learning, which follows a so-called "divide-and-conquer" strategy [27]. An ensemble paradigm commonly divides a complex classification or regression problem into a few simple tasks with lower computational expense and then combines a group of trained component learners to provide a comprehensive solution [28]. In the present work, we used the Boosting and Bagging algorithms, two most popular ensemble learning paradigms, to distinguish the gait patterns of the children participants into three age groups.

2.3.1. AdaBoost Algorithm. Boosting algorithms work by sequentially generating a number of weak learners to solve a classification or regression problem together [29]. In a typical boosting procedure, the training data for each weak learner are regenerated in order to correct the mistakes made by the previous learner. The AdaBoost algorithm is a representative boosting method that intends to accomplish the training of weak learners by reweighting or resampling the data samples [30]. Researchers have developed the family of AdaBoost

algorithms with plenty of extension versions, such as AdaBoost.R [30], AdaBoost.M1 [31], and AdaBoost.M2 [32], to solve different types of regression or classification problems. In the present work, we implemented the AdaBoost.M2 ensemble method that involved a total of 50 decision trees as the base learners to implement the gait pattern classifications. The computation process of the AdaBoost.M2 algorithm for the classification of children's gait patterns is summarized as follows.

Computation Process of the AdaBoost.M2 Algorithm

Input:

Gait data set: $\{\mathbf{f}_n, t_n\}_{n=1}^{N}$, $N = 50$ is the number of children, $t_n \in \{1, 2, 3\}$ is the class label;

Weak learner model (decision tree): $h(\mathbf{f}_n)$;

Number of ensemble learning iteration: I.

Initialization:

Initialize the gait data distribution $\ell_1(\mathbf{f}_n) = 1/N$.

Computation Procedure:

(1) *for* $i = 1, 2, \ldots, I$:
(1) Train a weak learner $h_i(\mathbf{f}_n)$.
(3) Calculate the error of the ith classifier: $e_i = \sum_{n:h_i(\mathbf{f}_n)\neq t_n} \ell_i(\mathbf{f}_n)$.
(4) Set $\alpha_i = (1/2)\ln((1 - e_i)/e_i)$.
(5) Update the distribution:

$$\ell_{i+1}(\mathbf{f}_n) = \frac{\ell_i(\mathbf{f}_n)}{Z_i} \begin{cases} \exp(-\alpha_i), & \text{if } h_i(\mathbf{f}_n) = t_i, \\ \exp(\alpha_i), & \text{if } h_i(\mathbf{f}_n) \neq t_i, \end{cases} \tag{10}$$

where Z_i is a normalization constant that makes $\ell_{i+1}(\mathbf{f}_n)$ be a probability distribution.
(6) *end*

Output:

$$H_{\text{AdaBoost}}(\mathbf{f}_n) = \text{sign}\left(\sum_{i=1}^{I} \alpha_i h_i(\mathbf{f}_n)\right). \tag{11}$$

2.3.2. Bagging Algorithm. Bagging stands for "bootstrap aggregating" [33], which contains the procedures of bootstrap sampling of training data, and aggregation of base learners by voting for classification problem or averaging for regression problem. The Bagging algorithm is able to greatly improve the generalization capability by combining weak learners (e.g., decision trees), rather than stable learners (such as k-nearest neighbor classifiers, radial basis function networks, and support vector machines), which are insensitive to the adjustment of training data with a bootstrap distribution [33].

Given a data set containing N scatter points (gait patterns), the bootstrap sampling approach generates a new training data set of the same size, $\mathbf{f}_n^{\text{bd}}$, for each weak learner by random (the Monte Carlo method) sampling from the original data set $\mathbf{f}_n$ [26]. In the bootstrap sampling process, a data point (or gait pattern) is picked with the uniform probability, $1/N$, irrespective of whether being selected before or not. Such a bootstrap sampling mechanism may result in several data points appearing more than once, whereas some other points are replaced with these repetitions in the new training data set. When predicting a testing gait pattern, the Bagging algorithm aggregates the outputs of the weak learners by voting the class labels and then makes the most voted label as the ensemble decision [34]. Breiman [33] demonstrated that the generalization error of the Bagging ensemble would be greatly reduced in comparison with the prediction error of a single base learner. In the present study, we used the Bagging algorithm that combined 50 weak learners in the form of decision trees (the same number of learners as that of the AdaBoost.M2 algorithm for comparison purpose), to accomplish the children's gait pattern classification tasks. The detailed computation process of the Bagging algorithm is provided as follows.

Computation Process of the Bagging Algorithm

Input:

Gait data set: $\{\mathbf{f}_n, t_n\}_{n=1}^{N}$, $N = 50$ is the number of children, $t_n \in \{1, 2, 3\}$ is the class label;

Weak learner model (decision tree): $h(\mathbf{f}_n)$;

Number of weak learners: I.

Computation Procedure:

(1) *for* $i = 1, 2, \ldots, I$:
(2) Train a weak learner $h_i(\mathbf{f}_n^{\text{bd}})$ with a data set of bootstrap distribution $\mathbf{f}_n^{\text{bd}}$.
(3) Predict the class labels of the input patterns with the trained learners $h_i(\mathbf{f}_n; \mathbf{f}_n^{\text{bd}})$.
(4) *end*

Output:

$$H_{\text{Bagging}}(\mathbf{f}_n) = \arg\max_{t\in\{1,2,3\}} \sum_{i=1}^{I} \left[h_i\left(\mathbf{f}_n; \mathbf{f}_n^{\text{bd}}\right) = t\right]. \tag{12}$$

2.4. Classification Performance Evaluation. With the purpose of categorizing children's gait patterns into multiple classes (three age groups), we considered the one-versus-rest strategy, which makes the classifiers train and test with the patterns of a specified class as positive cases and all other cases as negative ones. Such a classification process was alternately implemented for each class. The classification results of the AdaBoost.M2 and Bagging algorithms were then evaluated with the recall, precision, and accuracy metrics. Let TP_t, FP_t, and N_t denote the number of true positive (correct classification) cases, the number of predicted positive cases, and the total number of cases for a specified class ($t \in \{1, 2, 3\}$),

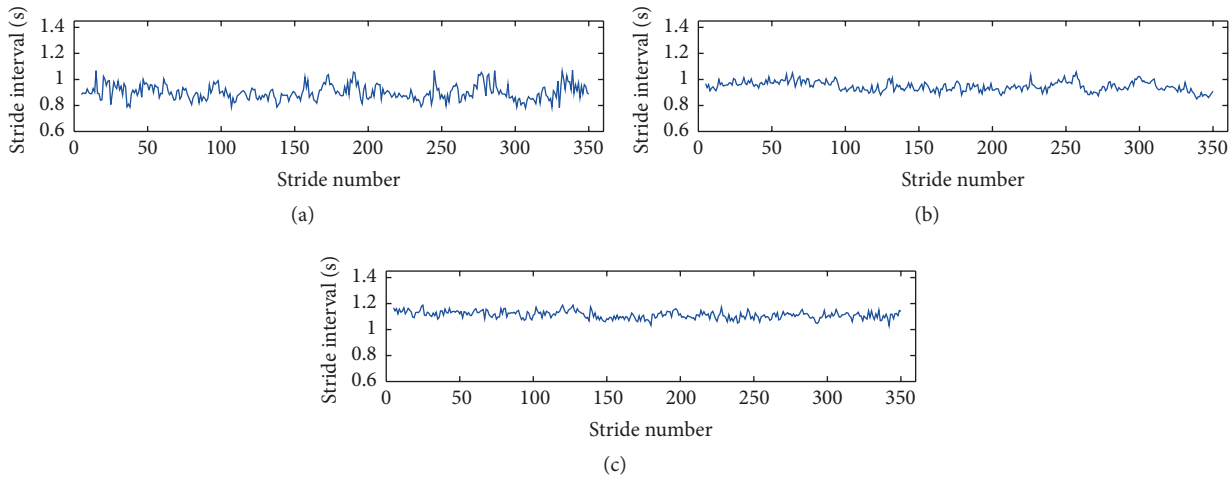


Figure 1: Series of stride interval of the children (a) aged 47 months, (b) aged 88 months, and (c) aged 148 months, respectively. The first strides come after the start-up walking for 60 s, and the strides during the last 5 s walking are excluded in the stride series.

respectively. Recall is defined as the true positive rate or sensitivity; that is,

$$\text{Recall}_t = \frac{\text{TP}_t}{N_t}. \tag{13}$$

Precision represents the positive predictive value, which is expressed as

$$\text{Precision}_t = \frac{\text{TP}_t}{\text{TP}_t + \text{FP}_t}. \tag{14}$$

Accuracy is the percentage ratio of all correct classified cases over the total number of cases:

$$\text{Accuracy} = \frac{\sum_{t=1}^{3} \text{TP}_t}{\sum_{t=1}^{3} N_t} \times 100\%. \tag{15}$$

3. Results and Discussions

Figure 1 plots the series of stride interval of three children in the corresponding age groups, respectively. The beginning strides in the first 60 s and the ending strides in the final 5 s during the gait monitoring period have been excluded in the gait series records. The outliers were also removed in the stride series by the median filter developed by Wu and Krishnan [20].

Figure 2 shows different SampEn and ASI values in bar graphics for the children participants in three age groups. It can be observed that the SampEn values consistently decrease from 0.408 bits (young age group) to 0.194 bits (middle age group), until 0.1 bits (elder age group). However, the ASI values slightly raise from 0.904 s (young children) to 0.961 s (middle children), until 1.059 s (elder children). Reduction of the SampEn results indicates that the irregularity in the series of stride interval has been ameliorated with the body maturation in children. Increase of the ASI values suggests that the children participants are able to coordinate larger strides when they grow up.

In the present study, we also normalized the SampEn and ASI parameters by the leg length and body mass for each participant, respectively. Statistical results of the original and normalized SampEn values, along with the original and normalized ASI values, for the children in the young, middle, and elder age groups are provided in Table 2.

It is clear that the changes of the original SampEn and ASI parameters between any two age groups are over the statistical significance level of the Mann-Whitney U test ($p < 0.01$). The SampEn normalized by leg length significantly reduces more than a half, from 0.755 bits/m to 0.304 bits/m, when the children grow up until 8 years old. For the children aged 10–14 years, the SampEn value normalized by leg length becomes 0.129 bits/m on average, with a decrement of 0.626 bits/m versus that of the young children aged 3–5 years. The SampEn normalized by body mass also decreases from 0.023 to 0.003 bits/kg for the children aged 10–14 years. Such results indicate that the gait irregularity, parameterized with the normalized SampEn by leg length and body mass, has been greatly improved in a close relationship with the maturation of motor control and musculoskeletal development in adolescence. The gait irregularity is reduced rapidly when the children become 8 years old, and the stride variability continues to decrease in children until the age of 14 years. The ASI values normalized by leg length and body mass are consistently becoming smaller in children with aging over the significance level ($p < 0.01$). However, the original ASI value is with an increasing trend, which is different from the normalized values. Such results indicate that the musculoskeletal development and gain in weight are more remarkable than the increase of stride interval in children. Both of the SampEn and ASI results suggest that the growth of musculoskeletal and neurological systems enable the children to better modulate the gait cadence rhythm, which confirms the observations in previous related studies of Hausdorff et al. [9] and Xiang et al. [17].

Table 2: Statistics of the original and normalized SampEn(2, 0.05, 350) and the original and normalized ASI values for the children in the young, middle, and elder age groups. Statistical differences between pairs of age groups are evaluated by the Mann-Whitney U hypothesis test (significance level $p < 0.01$). SampEn: sample entropy. ASI: average stride interval. $*$: U test between the young and middle age groups; $**$: U test between the middle and elder age groups; $***$: U test between the young and elder age groups.

Entropy parameters	Statistics (mean ± standard deviation)			p value
	Young group (aged 3–5 years)	Middle group (aged 6–8 years old)	Elder group (aged 10–14 years)	
SampEn (bit)	0.408 ± 0.109	0.194 ± 0.088	0.1 ± 0.058	0.001^{*} 0.001^{**} 0.001^{***}
Normalized SampEn by leg length (bit/m)	0.755 ± 0.229	0.304 ± 0.139	0.129 ± 0.084	0.001^{*} 0.001^{**} 0.001^{***}
Normalized SampEn by body mass (bit/kg)	0.023 ± 0.008	0.008 ± 0.003	0.003 ± 0.002	0.001^{*} 0.001^{**} 0.001^{***}
ASI (s)	0.904 ± 0.041	0.961 ± 0.041	1.059 ± 0.063	0.004^{*} 0.001^{**} 0.001^{***}
Normalized ASI by leg length (s/m)	1.661 ± 0.132	1.495 ± 0.122	1.35 ± 0.106	0.001^{*} 0.002^{**} 0.001^{***}
Normalized ASI by body mass (s/kg)	0.051 ± 0.008	0.039 ± 0.005	0.026 ± 0.004	0.001^{*} 0.001^{**} 0.001^{***}

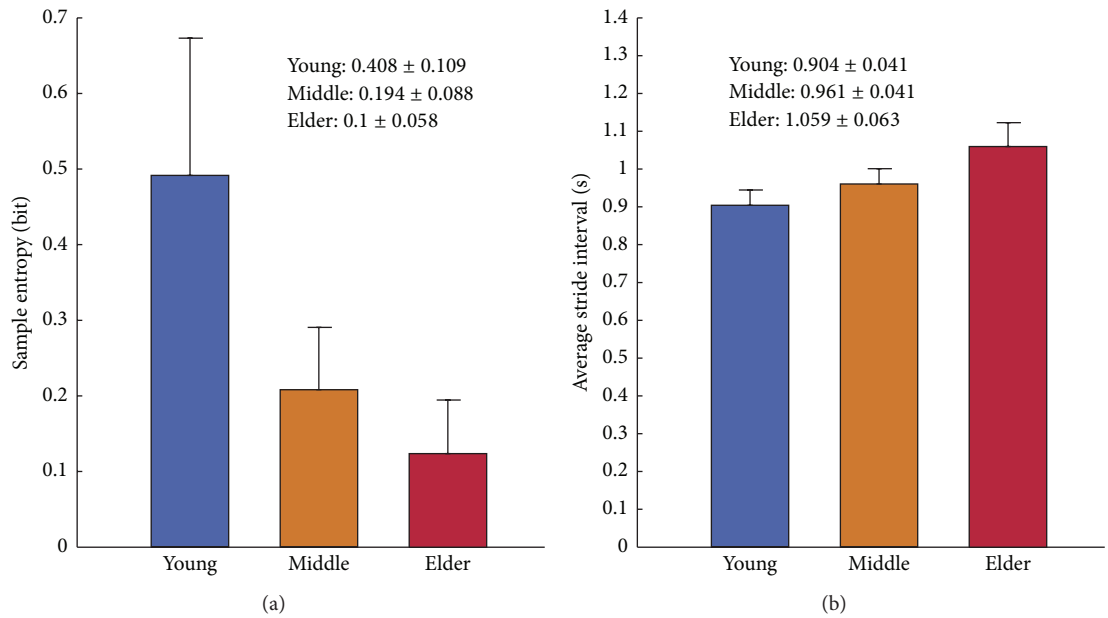


Figure 2: Statistics of (a) sample entropy (SampEn) and (b) average stride interval (ASI) of the children in the young (3–5 years old), middle (6–8 years old), and elder (10–14 years old) age groups, respectively.

The gait pattern classification results are tabulated in Table 3. Both of the AdaBoost.M2 and Bagging algorithms provided excellent overall accurate rates (AdaBoost.M2: 90%, Bagging: 92%). The Bagging algorithm correctly categorized all 14 gait patterns in the young children group, whereas the AdaBoost.M2 algorithm misclassified a child of 45 months after birth into the middle age group. Thus, the Bagging algorithm outperformed the AdaBoost.M2 algorithm with better results in terms of recall (Bagging: 0.9286 versus AdaBoost.M2: 0.8571) and precision (Bagging: 0.84 versus

Table 3: Gait pattern classification results obtained by the AdaBoost.M2 and Bagging ensemble methods.

Classification evaluation metrics	Ensemble methods	
	AdaBoost.M2	Bagging
Accuracy (%)	90%	92%
Recall		
Young (3–5 years old)	0.8571	0.9286
Middle (6–8 years old)	1	1
Elder (10–14 years old)	0.8	0.8
Precision		
Young (3–5 years old)	1	1
Middle (6–8 years old)	0.8077	0.84
Elder (10–14 years old)	1	1

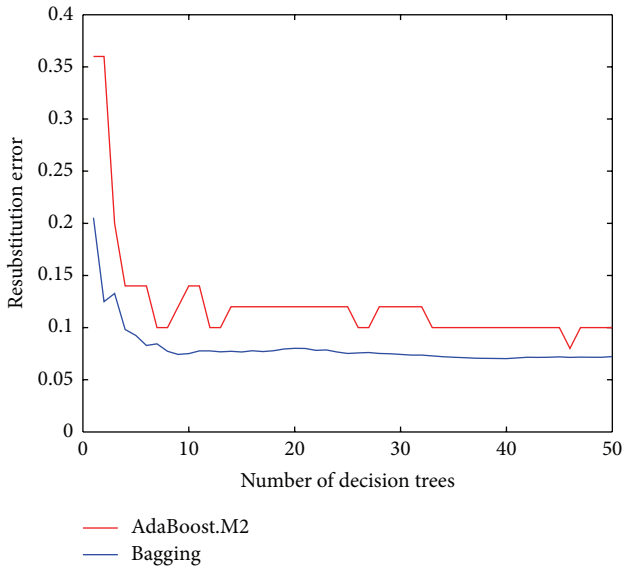


Figure 3: Resubstitution errors of the ensembles in relation to the increasing number of decision trees that are involved in the AdaBoost.M2 and Bagging algorithms, respectively.

AdaBoost.M2: 0.8077). Figure 3 displays the resubstitution errors produced by the AdaBoost.M2 and Bagging algorithms when generating new decision tree learners. It is worth noting that both ensemble methods can greatly reduce the output errors. The error curve of the Bagging algorithm is consistently below that of the AdaBoost.M2 algorithm, which confirms the effectiveness and superiority of the Bagging algorithm for solving the children's gait pattern classification problem.

4. Conclusion

Computer-aided quantification of stride dynamics and analysis of gait patterns may provide useful information on the neuromotor development in adolescence. In the present work, the SampEn and ASI parameters were computed to investigate the degree of gait regularity and the average gait cadence duration in children. The SampEn parameter can adapt to a small length of gait signal, such that it is not necessary to require the children participants to walk for a long-term gait monitoring. It is therefore very suited for the gait maturation assessment in adolescents, especially for young children who may have muscular fatigue in long-distance walking. Our results show that the SampEn and ASI values are significantly changing in adolescents aged from 3 to 14 years. The classification results demonstrated the effectiveness of the AdaBoost.M2 and Bagging ensemble algorithms in the identification of gait patterns for the children in different age groups. In the future study, we plan to recruit more gender-matched children participants in the three age groups for more accurate and unbiased statistical analysis of gait patterns during short-term and long-term walking monitoring. More temporal and computational tools [28] would be considered to analyze other stride phases, such as stance interval, swing interval, and double support time.

Conflict of Interests

There is no conflict of interests.

Acknowledgments

This work was partially funded by the National Natural Science Foundation of China under Grant nos. 31200769 and 81101115. Yunfeng Wu and Meihong Wu were supported by the Program for New Century Excellent Talents in Fujian Province University.

References

[1] D. Sutherland, "The development of mature gait," *Gait & Posture*, vol. 6, no. 2, pp. 163–170, 1997.

[2] D. Sutherland, R. A. Olshen, E. N. Biden, and M. P. Wyatt, *The Development of Mature Walking*, MacKeith, Oxford, UK, 1988.

[3] K. G. Holt, E. Saltzman, C.-L. Ho, M. Kubo, and B. D. Ulrich, "Discovery of the pendulum and spring dynamics in the early stages of walking," *Journal of Motor Behavior*, vol. 38, no. 3, pp. 206–218, 2006.

[4] S. J. Hillman, B. W. Stansfield, A. M. Richardson, and J. E. Robb, "Development of temporal and distance parameters of gait in normal children," *Gait and Posture*, vol. 29, no. 1, pp. 81–85, 2009.

[5] V. L. Chester, M. Tingley, and E. N. Biden, "A comparison of kinetic gait parameters for 3–13 year olds," *Clinical Biomechanics*, vol. 21, no. 7, pp. 726–732, 2006.

[6] S. R. Menkveld, E. A. Knipstein, and J. R. Quinn, "Analysis of gait patterns in normal school-aged children," *Journal of Pediatric Orthopaedics*, vol. 8, no. 3, pp. 263–267, 1988.

[7] D. C. Johnson, D. L. Damiano, and M. F. Abel, "The evolution of gait in childhood and adolescent cerebral palsy," *Journal of Pediatric Orthopaedics*, vol. 17, no. 3, pp. 392–396, 1997.

[8] V. Agostini, A. Nascimbeni, A. Gaffuri, P. Imazio, M. G. Benedetti, and M. Knaflitz, "Normative EMG activation patterns of school-age children during gait," *Gait & Posture*, vol. 32, no. 3, pp. 285–289, 2010.

[9] J. M. Hausdorff, L. Zemany, C.-K. Peng, and A. L. Goldberger, "Maturation of gait dynamics: stride-to-stride variability and its temporal organization in children," *Journal of Applied Physiology*, vol. 86, no. 3, pp. 1040–1047, 1999.

[10] K. G. Holt, E. Saltzman, C.-L. Ho, and B. D. Ulrich, "Scaling of dynamics in the earliest stages of walking," *Physical Therapy*, vol. 87, no. 11, pp. 1458–1467, 2007.

[11] K. Keller, A. M. Unakafov, and V. A. Unakafova, "Ordinal patterns, entropy, and EEG," *Entropy*, vol. 16, no. 12, pp. 6212–6239, 2014.

[12] R. M. Rangayyan, F. Oloumi, Y. Wu, and S. Cai, "Fractal analysis of knee-joint vibroarthrographic signals via power spectral analysis," *Biomedical Signal Processing and Control*, vol. 8, no. 1, pp. 23–29, 2013.

[13] J.-R. Huang, S.-Z. Fan, M. F. Abbod, K.-K. Jen, J.-F. Wu, and J.-S. Shieh, "Application of multivariate empirical mode decomposition and sample entropy in EEG signals via artificial neural networks for interpreting depth of anesthesia," *Entropy*, vol. 15, no. 9, pp. 3325–3339, 2013.

[14] Q. Wei, Q. Liu, S.-Z. Fan et al., "Analysis of EEG via multivariate empirical mode decomposition for depth of anesthesia based on sample entropy," *Entropy*, vol. 15, no. 9, pp. 3458–3470, 2013.

[15] R. Sharma, R. B. Pachori, and U. R. Acharya, "Application of entropy measures on intrinsic mode functions for the automated identification of focal electroencephalogram signals," *Entropy*, vol. 17, no. 2, pp. 669–691, 2015.

[16] Y. Wu, Z. Zhong, M. Lu, and J. He, "Statistical analysis of gait maturation in children based on probability density functions," in *Proceedings of the 33rd Annual International Conference of IEEE Engineering in Medicine and Biology Society (EMBC '11)*, pp. 1652–1655, Boston, Mass, USA, September 2011.

[17] N. Xiang, S. Cai, S. Yang et al., "Statistical analysis of gait maturation in children using nonparametric probability density function modeling," *Entropy*, vol. 15, no. 3, pp. 753–766, 2013.

[18] A. Shumway-Cook and M. H. Woollacott, *Motor Control: Theory and Practical Applications*, Lippincott Williams & Wilkins, Philadelphia, Pa, USA, 2nd edition, 2000.

[19] J. M. Hausdorff, Z. Ladin, and J. Y. Wei, "Footswitch system for measurement of the temporal parameters of gait," *Journal of Biomechanics*, vol. 28, no. 3, pp. 347–351, 1995.

[20] Y. Wu and S. Krishnan, "Statistical analysis of gait rhythm in patients with Parkinson's disease," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 150–158, 2010.

[21] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, John Wiley & Sons, Hoboken, NJ, USA, 1999.

[22] R. M. Rangayyan and Y. Wu, "Screening of knee-joint vibroarthrographic signals using probability density functions estimated with Parzen windows," *Biomedical Signal Processing and Control*, vol. 5, no. 1, pp. 53–58, 2010.

[23] L. I. Kuncheva, *Combining Pattern Classiers: Methods and Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[24] Y. Wu and S. Krishnan, "Combining least-squares support vector machines for classification of biomedical signals: A case study with knee-joint vibroarthrographic signals," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 23, no. 1, pp. 63–77, 2011.

[25] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, Boca Raton, Fla, USA, 2012.

[26] S. Cai, S. Yang, F. Zheng, M. Lu, Y. Wu, and S. Krishnan, "Knee joint vibration signal analysis with matching pursuit decomposition and dynamic weighted classifier fusion," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 904267, 11 pages, 2013.

[27] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Englewood Cliffs, NJ, USA, 2nd edition, 1998.

[28] Y. Wu, X. Luo, F. Zheng, S. Yang, S. Cai, and S. C. Ng, "Adaptive linear and normalized combination of radial basis function networks for function approximation and regression," *Mathematical Problems in Engineering*, vol. 2014, Article ID 913897, 14 pages, 2014.

[29] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[30] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, part 2, pp. 119–139, 1997.

[31] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[32] G. Eibl and K.-P. Pfeiffer, "Multiclass boosting for weak classifiers," *Journal of Machine Learning Research*, vol. 6, pp. 189–210, 2005.

[33] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[34] G. Fumera, F. Roli, and A. Serrau, "A theoretical analysis of bagging as a linear combination of classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1293–1299, 2008.