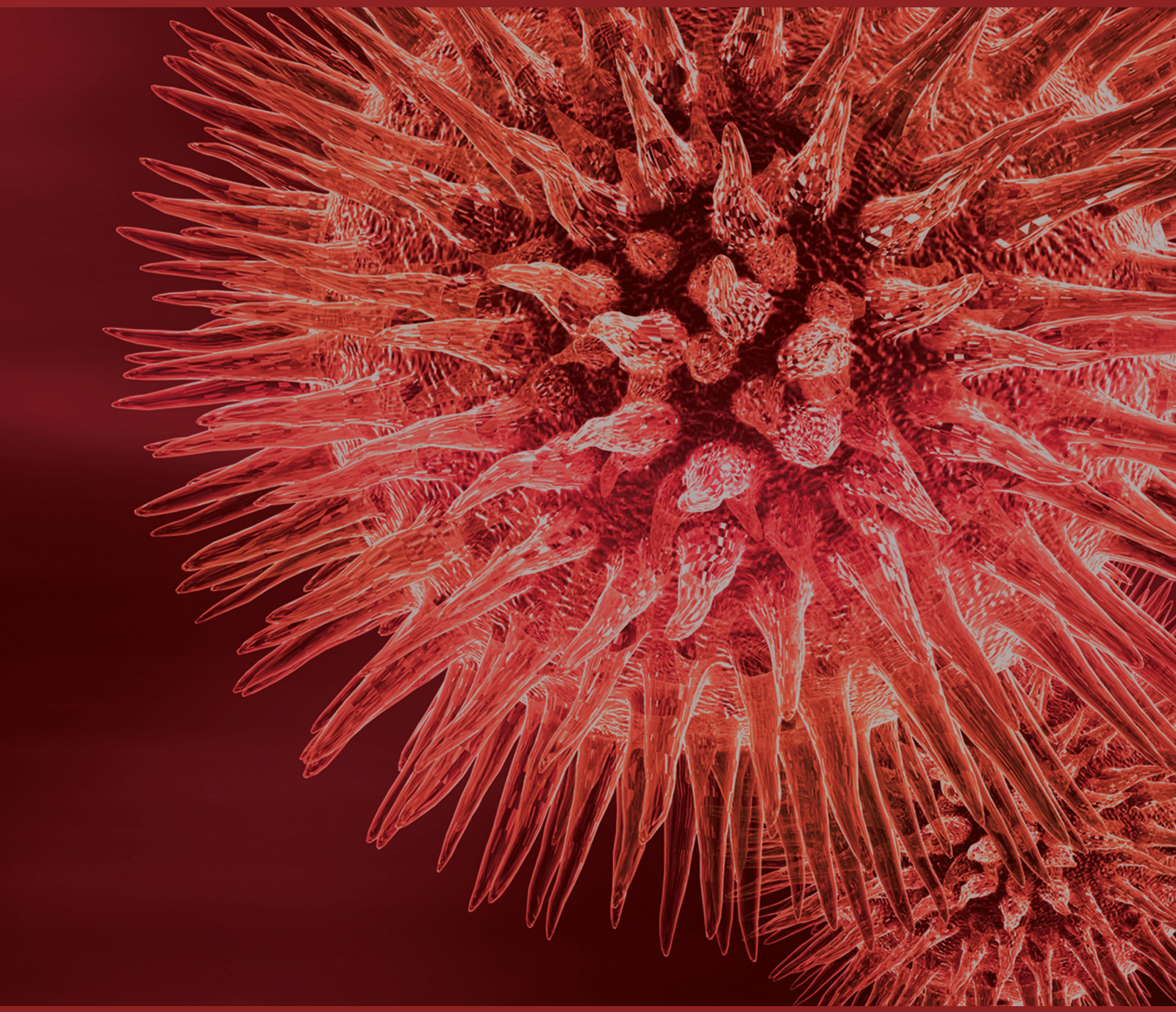
BioMed Research International

Frontiers in Integrative Genomics and Translational Bioinformatics

Guest Editors: Zhongming Zhao, Victor X. Jin, Yufei Huang, Chittibabu Guda, and Jianhua Ruan



Frontiers in Integrative Genomics and Translational Bioinformatics

BioMed Research International

Frontiers in Integrative Genomics and Translational Bioinformatics

Guest Editors: Zhongming Zhao, Victor X. Jin, Yufei Huang, Chittibabu Guda, and Jianhua Ruan

Contents

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 725491, 3 pages
http://dx.doi.org/10.1155/2015/725491

Editorial

Frontiers in Integrative Genomics and Translational Bioinformatics

Zhongming Zhao,[1,2,3] Victor X. Jin,[4] Yufei Huang,[5] Chittibabu Guda,[6] and Jianhua Ruan[7]

[1]*Department of Biomedical Informatics, Vanderbilt University School of Medicine, Nashville, TN 37203, USA*
[2]*Department of Psychiatry, Vanderbilt University School of Medicine, Nashville, TN 37212, USA*
[3]*Department of Cancer Biology, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*
[4]*Department of Molecular Medicine, The University of Texas Health Science Center at San Antonio, San Antonio, TX 78229, USA*
[5]*Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA*
[6]*Department Genetics, Cell Biology and Anatomy, University of Nebraska Medical Center, Omaha, NE 68198, USA*
[7]*Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249, USA*

Correspondence should be addressed to Zhongming Zhao; zhongming.zhao@vanderbilt.edu
and Jianhua Ruan; jianhua.ruan@utsa.edu

Received 21 September 2015; Accepted 21 September 2015

As we have officially entered the big data era, we are embracing numerous opportunities but also meeting strong demand on innovative approaches to managing and analyzing the data in the digital world. The speed of data generation is astonishing—from now until 2020, the volume of the digital data is expected to approximately double every two years, and 90% of the data available to us today were generated in just the last two years. Specifically in biological and biomedical research fields, we have witnessed the rapid advances in biotechnologies, especially the next-generation sequencing and single cell technologies, enabling the investigators to create massive amounts of data for genomics and translational research. While analysis of the data from single domain like mutations or gene expression is often the first choice in a project, integrative genomic approaches have more advantages such as robustness in detecting the biological signals or biomarkers and low false discovery rates due to the evidence from multiple domains. Therefore, we have steadily seen more integrative genomic studies during the past several years. Although these approaches are promising and powerful, there are many challenges that are required to be addressed by bioinformaticians, computational biologists, and other scientists. These challenges include, but are not limited to, data quality and process from different technology platforms, sample size and consistency, data missingness, incomplete and inaccurate knowledgebase (e.g., reference networks and pathways), false discovery, lack of novel algorithms for data integration, computational efficiency, data interpretation, and visualization.

Translational bioinformatics is an emerging field that focuses on applying informatics methodology to the increasing amount of biomedical and genomic data in order to generate knowledge for clinical applications. With the large genomic data linked to phenotype and medical records, we now can not only discover interesting biological features and regulations using genomic approaches, but also translate some of the findings for clinical practice. For example, investigators have been interested in finding actionable mutations that can be used for development of precision medicine strategies from thousands of mutations or even more in an individual genome. In addition to the challenges above, there are other topics that require immediate attention such as ownership and privacy of the findings, data sharing, efficient clinical decision support system, and design and development of specific gene panel for fast patient screening, among others.

Therefore, we launched this special issue to address the demand for integrative genomics and translational bioinformatics. We are interested in both new algorithms/tools and applications. The special issue welcomes the genomics,

bioinformatics, and computational work in broad areas such as various omics technologies, multidimensional data integration, systems biology approaches, precision medicine studies, single cell research, pharmacogenomics, machine learning, high performance computing, and visualization. Special call for papers went through The International Conference on Intelligent Biology and Medicine (ICIBM 2014, held on December 4–6, 2014, http://compgenomics.utsa.edu/icibm2014/) and *BioMed Research International* journal website. After rigorous peer review, articles were selected for this special issue. We briefly describe the research projects presented in these articles as follows.

Three papers present the work to advance the next-generation sequencing technologies. In "A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference," A. Cornish and C. Guda performed a systematic evaluation of variant callers to determine which pipeline has the best performance in variant calling. They compared six different aligners and five different variant callers—a total of 30 combinations—using the data generated by NIST Genome in a Bottle Consortium. For single nucleotide variant call, the authors found that Novoalign combined with GATK UnifiedGenotyper had the highest sensitivity while keeping a low false positive rate. However, calling insertion and deletion (indel) variants still remained a big challenge—none of the tools could achieve an average sensitivity higher than 33% or a positive predictive value (PPV) higher than 53%. In the paper entitled "RNAseq by Total RNA Library Identifies Additional RNAs Compared to Poly(A) RNA Library," Y. Guo et al. evaluated the ability of detecting RNA for two popularly used RNA libraries in RNA sequencing: the poly(A) captured RNA library, which captures RNA based on the presence of poly(A) tails at the $3'$ end, and the total RNA library, which captures total RNA. By using the two breast cancer cell lines, the authors found that the RNA expression values captured by both RNA libraries were highly correlated, but the number of RNA molecules captured by the total RNA library was significantly higher than that by the poly(A) library. The authors also identified several specific RNA sets that could not be captured by the poly(A) library. In the paper entitled "Assessing Computational Steps for CLIP-Seq Data Analysis," Q. Liu et al. presented a systematic evaluation of major computational steps for identifying RNA-binding protein (RBP) using a special technology: CLIP-Seq. CLIP (cross-linking and immunoprecipitation) is designed to study protein-RNA interactions *in vivo*, such as RNA and RBP interactions. The authors evaluated data analysis steps including preprocessing, selection of control samples, peak normalization, and motif discovery. The authors reported three factors (avoiding PCR amplification artifacts, normalizing input RNA or mRNAseq, and defining the background model from control samples) could help reduce the bias due to the RNA abundance and could improve detecting binding sites. The work is helpful for analysis of CLIP-Seq data.

Cancer is a common complex disease and can occur in many tissue types and different parts of the body. Molecular data may be useful to classify cancer sites or subtypes. In the paper "Classification of Cancer Primary Sites Using Machine Learning and Somatic Mutations," Y. Chen et al. attempted to classify cancer primary sites using large-scale somatic mutations observed in cancer genomes and machine learning method. Specifically, they examined the patterns of 1,760,846 somatic mutations identified from 230,255 cancer patients covering 17 tumor sites using support vector machine (SVM). Through a multiclass classification experiment and using gene symbol, somatic mutation, chromosome, and gene functional pathway as predictors, the authors reported the performance of the baseline using only gene features to be 0.57 in accuracy, but it was improved to 0.62 when adding the information of mutation and chromosome. Moreover, F-measure values could reach 0.70 in five primary sites with the large intestine being 0.87. The study suggested that the somatic mutation information is useful for prediction of primary tumor sites. In another machine learning paper, entitled "Construction of Pancreatic Cancer Classifier Based on SVM Optimized by Improved FOA," H. Jiang et al. introduced an improved quantum fruit fly optimal algorithm (FOA) based method. Specifically, the improved FOA was used to optimize the parameters of SVM and a classifier was constructed based on the optimized SVM. The authors applied their method to classify pancreatic cancer and showed improved performance.

Systems pharmacology has emerged as a major computational field, which systems biology approaches have often applied to large-scale complex drug data. This special issue includes two papers in this area. In the paper "How to Choose In Vitro Systems to Predict In Vivo Drug Clearance: A System Pharmacology Perspective," L. Wang et al. evaluated the performance of different recombinant human enzyme expression systems for predicting hepatic clearance in human body. The performance of different *in vitro* systems was compared after *in vitro-in vivo* extrapolation. Among the four systems (*Escherichia coli* system, yeast system, lymphoblastoid system, and baculovirus system) they compared, baculovirus system had the best performance and was suggested to be the most suitable system for the large-scale drug clearance prediction. In the paper entitled "Predicting Drug-Target Interactions via Within-Score and Between-Score," J. Y. Shi et al. presented their computational prediction of drug-target interactions (DTIs). They characterized each drug-target pair (DTP) as a feature vector of within-scores and between-scores so that their approach has consistent form of DTPs, a reduced bias, and sharing the same visualized space between known DTIs and unapproved DTPs. They evaluated the effectiveness of their approach by comparing with other popular methods under cross-validation and predicting potential interactions for DTPs under the validation in existing databases.

In the paper "Coexpression Network Analysis of miRNA-142 Overexpression in Neuronal Cells," I. Thapa et al. applied a correlation network model to find the coexpressed genes and how miRNA-142 overexpression impacts on the network. The authors focused on miRNA-142 because it was found to be upregulated in neurons and its overexpression plays important roles in other genes like *SIRT1* and *MAOA*. They found that several nervous system development related genes such as *TEAD2*, *PLEKHA6*, and *POGLUT1* were affected by miRNA-142 overexpression.

In the paper "OperomeDB: A Database of Condition-Specific Transcription Units in Prokaryotic Genomes," K. Chetal and S. C. Janga present OperomeDB, a database that ensembles all the predicted operons for bacterial genomes using available RNA-sequencing datasets across a wide range of experimental conditions. The database currently contains nine bacterial organisms and 168 transcriptomes from which operons were predicted by the authors. Web interface, visualization, data query, and other functions are provided.

In the paper entitled "Building Integrated Ontological Knowledge Structures with Efficient Approximation Algorithms," Y. Xiang and S. C. Janga tackled a basic problem on integrating a pair of ontology tree structures with a given closeness matrix. After they identified optimal structures for the problem, the authors proposed optimal and efficient approximation algorithms for integrating a pair of ontologies as well as multiple ontologies. Their results using Gene Ontology and National Drug File Reference Terminology suggested that the method should be effective on association studies between biomedical terms.

Acknowledgments

We would like to acknowledge the anonymous reviewers for their critical comments that helped to improve the quality of the papers in this special issue. We would like to also acknowledge the organizers and committee members of The International Conference on Intelligent Biology and Medicine (ICIBM 2014, held on December 4–6, 2014) for their efforts to provide a forum to discuss integrative genomics and computational systems medicine, through which this special issue was made possible. We thank the National Science Foundation (NSF Grant IIS-1451135) and University of Texas Health Science at San Antonio for financial support of ICIBM 2014.

Zhongming Zhao
Victor X. Jin
Yufei Huang
Chittibabu Guda
Jianhua Ruan

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 501528, 14 pages
http://dx.doi.org/10.1155/2015/501528

Research Article

Building Integrated Ontological Knowledge Structures with Efficient Approximation Algorithms

Yang Xiang[1] and Sarath Chandra Janga[2]

[1]*Department of Biomedical Informatics, The Ohio State University, Columbus, OH 43210, USA*
[2]*Department of BioHealth Informatics, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, USA*

Correspondence should be addressed to Sarath Chandra Janga; scjanga@iupui.edu

Received 22 October 2014; Revised 30 December 2014; Accepted 1 January 2015

Academic Editor: Dongchun Liang

The integration of ontologies builds knowledge structures which brings new understanding on existing terminologies and their associations. With the steady increase in the number of ontologies, automatic integration of ontologies is preferable over manual solutions in many applications. However, available works on ontology integration are largely heuristic without guarantees on the quality of the integration results. In this work, we focus on the integration of ontologies with hierarchical structures. We identified optimal structures in this problem and proposed optimal and efficient approximation algorithms for integrating a pair of ontologies. Furthermore, we extend the basic problem to address the integration of a large number of ontologies, and correspondingly we proposed an efficient approximation algorithm for integrating multiple ontologies. The empirical study on both real ontologies and synthetic data demonstrates the effectiveness of our proposed approaches. In addition, the results of integration between gene ontology and National Drug File Reference Terminology suggest that our method provides a novel way to perform association studies between biomedical terms.

1. Introduction

In recent years, ontologies are becoming increasingly important in knowledge engineering. Generally speaking, an ontology is a collection of concepts and their relations. It has wide applications in computer science and life science. For example, in computer science, semantic web uses web ontology language (OWL) to represent knowledge bases [1]. In life sciences, numerous important data structures and tools are built on ontologies. One of the most popular ontologies is gene ontology. Researchers use it frequently to measure the enrichment of gene clusters and to identify potential biomarkers. Two most famous ontology databases in the biomedical field are Unified Medical Language System (UMLS) [2] and NCBO BioPortal (https://bioportal.bioontology.org/). The former has more than 100 ontology datasets and the latter has more than 300 ontology datasets.

Although ontologies can be modeled as a directed graph, many ontologies are in fact hierarchical trees or have hierarchical tree-like structures. In the BioPortal website, users can find basic hierarchical properties of an ontology, such as the maximum depth and the maximum number of children. In the UMLS, the hierarchical structure of an ontology is documented in the "MRHIER.RRF" file with each line being a path from a term to its root. We can build a hierarchical tree from these paths by merging the common nodes starting from the root. Because the hierarchical structures of some ontologies are in fact directed acyclic graphs, the hierarchical tree may contain some duplicated concepts. To simplify our study, we treat them as independent concepts in this work.

An important knowledge discovery task is to identify knowledge associations. In life science, this task includes finding the associations between diseases and genes [3, 4] and between phenotypes and genotypes [5]. With the presence of ontologies, such a task has been extended from identifying the associations between terms to the associations between ontologies as a whole. For the latter, we should not only consider the term associations, but also the term associations in the context of their ontological structures. For example, if the parent and children of term a (a from ontology A) are

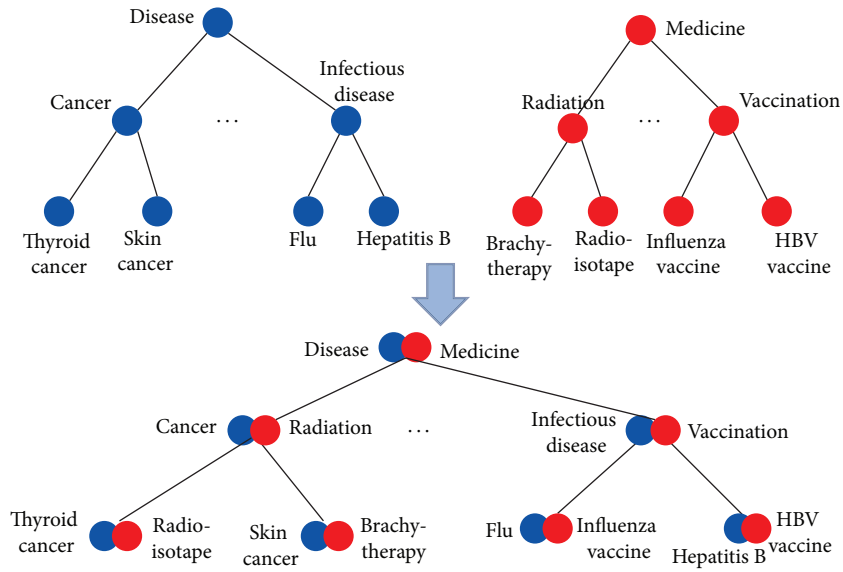


FIGURE 1: A simple example of integrating two hypothetical ontologies.

similar to those of term b (b from ontology B), Then a may be a good choice to be associated with b.

Early studies on ontology integration relied on domain experts to manually set up the integration rules [6]. However, this approach cannot meet the ever increasing volume of ontology datasets. Automatic ontology integration methods have been developed to address this issue. However, as we will see in the discussions of Section 1.3, these methods are often heuristic or have not demonstrated the effectiveness in integrating a large volume of ontology datasets. Thus, our goal is to develop an ontology integration method that is able to deliver optimal or close-to-optimal solutions for integrating a large volume of ontology datasets (particularly from the biomedical domain). As discussed above, we focus on ontologies with hierarchical tree-like structures which are often available in the biomedical domain. In addition, we assume the ontology term closeness measurement is available. This assumption is reasonable because many applications are able to identify ontology term similarities via additional data sources. For example, a closeness matrix between two sets of biomedical terms can be generated by using UMLS knowledge discovery methods such as KDLS [7]. Our problem can be formally described as follows.

1.1. Problem Formulation. The basic ontology integration problem in our work can be formulated as follows. Given ontology tree structures T_A and T_B and a closeness matrix M_{AB}, how can we efficiently generate an integrated ontology tree structure T_{AB} meeting the following two basic criteria?

(1) For any two vertices x and y in T_A (or T_B), the lowest common ancestor $\mathrm{LCA}_{T_A}(x, y)$ (or $\mathrm{LCA}_{T_B}(x, y)$) is contained by $\mathrm{LCA}_{T_{AB}}(x, y)$.

(2) It holds that $\mathrm{argmax}_{T_{AB}} f(T_{AB}) = \sum_{X \in V(T_{AB})} M_{AB}(X)$. Here $M_{AB}(X)$ is the entry value in the closeness matrix for the corresponding two vertices (one from T_A and the other from T_B) contained in the node X. $M_{AB}(X) = 0$ if X, a node of T_{AB}, contains only one vertex from T_A or T_B.

We name $f_{T_{AB}}$ the cohesion function of the integrated ontology T_{AB} and its value is the overall cohesion score of integrating T_A and T_B into T_{AB}. Correspondingly, we define function $g(T_A, T_B) = \max_{T_{AB}}(\sum_{v \in V(T_{AB})} M_{AB}(v))$ as the maximum cohesion function for integrating the ontologies T_A and T_B and its value is the maximum overall cohesion score (or, simply, maximum cohesion score). In a hierarchical ontology, the common part of any two terms can be described by their lowest common ancestor. For example, in Figure 1, flu and hepatitis B are both infectious diseases, and flu and cancer are both diseases. Thus, we use Criterion (1) to ensure that the basic logic of an ontology is preserved after integration.

An example of integrating two hypothetical ontologies that satisfy Criterion (1) is given in Figure 1. To facilitate the understanding of our problem definition, we also provide another example of integrating two ontologies in Figure 2. As we can see in Figure 2, the lowest common ancestor of nodes containing thyroid cancer and infectious disease is the node containing cancer instead of disease. We conclude that the integration is a violation of Criterion (1). In fact, we can easily see that there are multiple pairs of nodes with incorrect lowest common ancestors in Figure 2.

In Section 2.2, we will extend the basic problem definition to handle the integration of multiple (>2) ontologies. The two basic criteria will be extended correspondingly.

1.2. Main Contributions. We made the following major contributions in this work.

(i) We proposed a novel ontology integration problem that optimizes the cohesion function. We identified optimal structures in this problem and developed

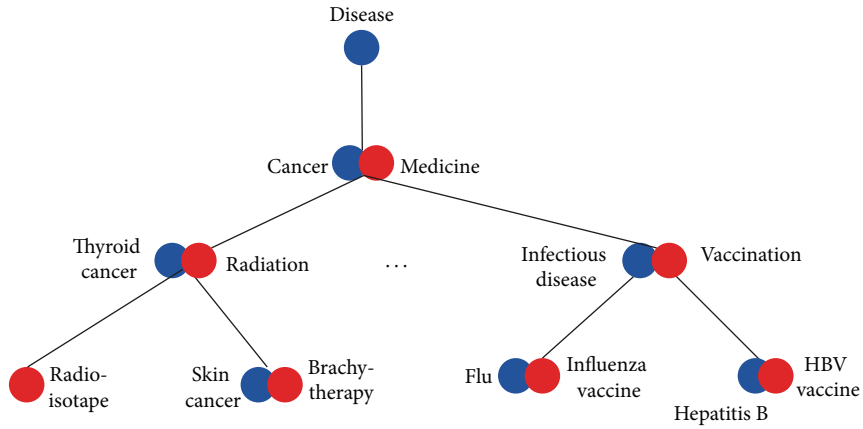


Figure 2: Another example of integrating the two ontologies in Figure 1. This integration violates Criterion (1).

optimal as well as efficient approximation solutions for this problem.

(ii) We extended the basic problem to handle the integration of large number of ontologies, and we developed both greedy and fast approximation algorithms for the extended problem.

(iii) We studied the proposed algorithms on both real and synthetic datasets and confirmed their effectiveness in integrating large volume of ontology datasets.

1.3. Related Work. Automatic ontology generation and integration are desirable in many applications and have been studied in the past decade. Although available methods for automatic ontology generation produce ontologies from a given type of data, such as gene networks [8], textual data [9], dictionary [10], and schemata [11, 12], they do not contribute to the integration of different types of ontologies which will bring innovative results on annotation/knowledge reuse and association studies. To address this issue, a number of studies have been focused on ontology integration [13, 14] and its medical domain applications [15]. The ontology integration methods available and used in these works can be generally classified into three categories.

Manually or Semiautomatic Setups. In [6], the authors presented a methodology for ontology integration by custom-tailored integration operations which include algebraic-specified 39 basic operations and 12 nonbasic operations derived from them. The authors identified a set of criteria, such as modularize, specialize, and diversify each hierarchy, for guiding the knowledge integration. In [16], the authors designed a semiautomatic approach for ontology merging. The ontology developers will be assisted by the system and guided to tasks needing their interventions.

Using Machine Learning Methods. Reference [17] describes an ontology mapping system GLUE that uses machine learning techniques for building ontology mappings. Specifically, GLUE uses multiple learning strategies. Each type of information from the ontologies is handled by a different learning strategy. The authors demonstrate that GLUE works effectively on taxonomy ontologies. Similarly, [18] also used multistrategy learning in matching pair identification. However, the ontologies used in the experiments of [18] contain less than 10 nodes. Although [17] studied the integration of larger ontologies in the experiments, those ontologies contain only around 34 to 176 nodes, much smaller than many ontologies used in the biomedical field.

Using Heuristic Approaches. Many automatic ontology integration methods [19, 20] fall into this category. They perform ontology integrations by using heuristic approaches from different perspectives. For example, [20] uses heuristic policies for selecting axioms and candidate merging pairs. From a quite different angle, [19] uses view-based query for guiding the ontology integrations.

These methods have a few major weaknesses including (1) lack of a systematic measurement to quantify the goodness for the ontology integration; (2) being generally heuristic with no theoretical results to show that the proposed integration approach is globally optimal or close to optimal; (3) being not for integrating large volume of ontologies. These weaknesses motivated us to develop efficient and near optimal solutions for integrating large ontology datasets.

2. Methods

2.1. Integrating a Pair of Ontologies. In this section, we focus on the basic problem of integrating two ontologies as formulated in Section 1.1. We will prove optimal structures in the problem and propose an optimal and an efficient approximation solution for this problem. These solutions are also the basis for solving the problem of integrating a large number of ontologies as described in Section 2.2.

2.1.1. Brutal-Force and Heuristic Solutions. Given Criteria (1) and (2), a brutal-force approach will pick up a best solution from all the solutions that start with integration involving at least one of the roots of the two ontology trees and iteratively integrate their descendants. Considering an extreme case where each ontology tree is a path of n vertices, we

```
push (0, 0) into queue q; {Integrating virtual roots of the two ontologies}
while |q| > 0 do
  (a, b) = pop(q);
  for i ∈ children of a on T_A do
    max_score = −1;
    to_merge = NULL;
    to_merge_root = NULL;
    for j ∈ children of b on T_B do
      if j is chosen then
        continue; {The subtree rooted at j can only be chosen once for integration, as illustrated in Figure 3.}
      else
        identify vertex k in the subtree rooted at j such that arg max_k(M_AB(i, k)/β^distance(k,j));
        if M_AB(i, k) > max_score then
          max_score = M_AB(i, k);
          to_merge = k;
          to_merge_root = j;
        end if
      end if
    end for
    if to_merge = NULL then
      break;
    else
      save merge pair (i, to_merge) in T_AB;
      mark to_merge_root as chosen;
      push (i, to_merge) into queue q;
    end if
  end for
end while
return T_AB;
```

Algorithm 1: HeuristicMerge(T_A, T_B, M_{AB}).

conclude that such a brutal-force approach needs to pick up a best solution from an exponential number of solutions. The brutal-force approach is clearly not acceptable for integrating large ontologies and may not even work for ontologies with only a few dozens of vertices.

A heuristic solution can be developed by following an idea similar to the above brutal-force approach. However, instead of trying all possibilities, the heuristic solution will greedily merge vertices following the topological order. When selecting a matching vertex for vertex a from ontology T_A, the heuristic approach will greedily select a vertex b from allowable candidates in T_B and iteratively apply such selections to descendants of a. According to Criterion (1), if a is associated with b, then none of a's descendants are allowed to be associated with vertices other than b's descendants. In addition, if a's child a' is associated with b's child b' or its descendants, then none of a's other children are allowed to be associated with b' or its descendants any more. Given this, a greedy choice may very easily end up in a local optimum by choosing a best matching vertex at one step, while denying integrating opportunities of other vertices that may lead to a better final solution.

It is easy to see that the deeper a vertex being chosen for integration is, the more integration opportunities are lost. To alleviate such a situation, we propose a greedy approach by considering the relative depth (rdepth) of a chosen vertex with regard to an allowable vertex closest to the root. That is, given a vertex a from T_A, a vertex b from T_B is chosen when $M_{AB}(a, b)/\beta^{\text{rdepth}(b)}$ is maximized. When $\beta = 1$, the depth information does not take effective and when $\beta = \infty$, each vertex will only be associated with an allowable vertex closest to the root.

Algorithm 1 describes the pseudocode of the heuristic integration. It starts by integrating virtual roots of the two ontologies. After that, the integration will be carried out iteratively from top to bottom by following Criterion (1) and the heuristic strategy described above. In the empirical study, we will see that the heuristic algorithm works better when the depth information is considered. However, in terms of the overall cohesion score, it is no match for our optimal and approximation solutions as described below.

2.1.2. Optimal and Approximation Solutions. By dividing the integration of two trees into node merging and subtree integrations, we have identified optimal structures in the basic problem, as stated by Lemmas 1 and 2. These optimal structures make it possible for us to develop efficient algorithms (Algorithms 2 and 3) that achieve optimal or approximation solutions. In the following, we first describe the two important lemmas suggesting the optimal structures and their proofs before describing our proposed algorithms.

Lemma 1. *Let r_a be the root of tree T_A and r_b the root of tree T_B. Let T_{A-r_a} and T_{B-r_b} represent two sets of sub trees rooted at r_a of*

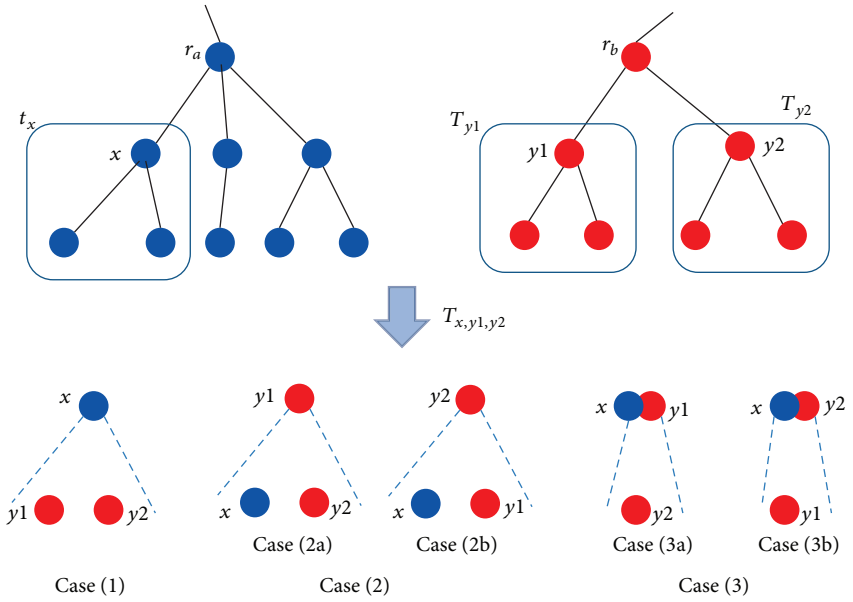


Figure 3: An illustration of three cases in the proof of Lemma 2.

```
Sort vertices in T_A and T_B in the topological order;
for i = |V(T_A)| to 0 do
  for j = |V(T_B)| to 0 do
    score = M_AB(i, j) + MaxMatch(T_{A−i}, T_{B−j});
    score_a = MaxMatch(T_{A−i}, T_j);
    score_b = MaxMatch(T_i, T_{B−j});
    cohesion_matrix(i, j) = max(score, score_a, score_b);
  end for
end for
return cohesion_matrix;
```

Algorithm 2: BuildCohesionMatrix(T_A, T_B).

T_A *and* r_b *of* T_B, *respectively.* T_{A-r_a} *does not include* r_a *and* T_{B-r_b} *does not include* r_b. *One has* $g(T_A, T_B) = \max(M_{AB}(r_a, r_b) + g(T_{A-r_a}, T_{B-r_b}), g(T_A, T_{B-r_b}), g(T_{A-r_a}, T_B))$.

Proof. We can divide the integration of tree T_A with tree T_B into two cases according to the merging of their roots.

(1) The roots of T_A and T_B are merged together.

(2) The roots of T_A and T_B are not merged together.

For case (1), it is clear that the cohesion score is $M_{AB}(r_a, r_b) + g(T_{A-r_a}, T_{B-r_b})$.

For case (2), we conclude that either T_A is integrated with T_{B-r_b} (r_b is out of integration) or T_B is integrated with T_{A-r_a} (r_a is out of integration). Otherwise, we will have a merged tree T_{AB} with two roots r_a and r_b, a contradiction to the fact that T_{AB} is a tree. Therefore, the cohesion score is either $g(T_A, T_{B-r_b})$ or $g(T_{A-r_a}, T_B)$.

Combining cases (1) and (2) and according to Criterion (2), we have

$$g(T_A, T_B) = \max\left(M_{AB}(r_a, r_b) + g\left(T_{A-r_a}, T_{B-r_b}\right), g\left(T_A, T_{B-r_b}\right), g\left(T_{A-r_a}, T_B\right)\right). \quad (1)$$

□

Lemma 2. *Let* T_{A-r_a} *and* T_{B-r_b} *represent two sets of trees obtained by removing the root vertices* r_a *from* T_{A-r_a} *and* r_b *from* T_{B-r_b}. *One has*

$$g\left(T_{A-r_a}, T_{B-r_b}\right) = \max\left(\sum_{(T_x, T_y)\in R}\left(g\left(T_x, T_y\right)\right)\right). \quad (2)$$

Here $T_x \in T_{A-r_a}$, $T_y \in T_{B-r_b}$, *and* R *is a matching of trees in* T_{A-r_a} *with trees in* T_{B-r_b}.

Proof. To prove this lemma, we first prove that for any tree $T_x \in T_{A-r_a}$, it can be integrated with no more than one tree in T_{B-r_b}. We will prove this claim by contradiction. Assume there are two trees $T_{y_1} \in T_{B-r_b}$ and $T_{y_2} \in T_{B-r_b}$ and they integrate with a tree $T_x \in T_{A-r_a}$ into an integrated tree T_{x,y_1,y_2}. There are three cases for the root r of T_{x,y_1,y_2}, as illustrated in Figure 3:

(1) r contains only the root of T_x;

(2) r contains only the root of T_{y_1} or T_{y_2};

(3) r contains the root of T_x and the root of T_{y_1} or T_{y_2}.

For case (1), the lowest common ancestor of the roots of T_{y_1} and T_{y_2} in the integrated tree T_{x,y_1,y_2} will no longer contain their lowest common ancestor in T_B, a contradiction to

```
push (0, 0, NULL) into queue q;
while |q| > 0 do
  (a, b, c) = pop(q);
  if a = −1 then
    make T_B(b) as a subtree of T_AB rooted at b; {T_B(b) is a sub tree of T_B rooted at b}
  else if b = −1 then
    make T_A(a) as a subtree of T_AB rooted at a; {T_A(a) is a sub tree of T_A rooted at a}
  else
    let d = (a, b) and make d a child of c on T_AB;
    score = M_AB(a, b) + MaxMatch(T_A−a, T_B−b);
    score_a = MaxMatch(T_A−a, T_b);
    score_b = MaxMatch(T_a, T_B−b);
    if score >= score_a && score >= score_b then
      push matching results of T_A−a, T_B−b with d into q;
    else if score_a >= score_b then
      push matching results of T_A−a, T_b with d into q;
    else
      push matching results of T_a, T_B−b with d into q;
    end if
  end if
end while
return T_AB;
```

Algorithm 3: BuildNewOnto(cohesion_matrix, T_A, T_B).

Criterion (1). For cases (2) and (3), the root of T_{y_2} (or the root of T_{y_1}) will be the descendant of the root of T_{y_1} (or the root of T_{y_2}) in the integrated tree T_{x,y_1,y_2}, a contradiction to Criterion (1). For integration involving more than two trees from T_{B-r_b}, we can still follow the above procedure to reach contradictions. Thus, the claim is proven.

Without loss of generality, we can see that, for any tree $T_y \in T_{B-r_b}$, it can be integrated with no more than one tree in T_{A-r_a}. Therefore, the integration between T_{A-r_a} and T_{B-r_b} corresponds to a matching in a weighted bipartite graph in which two sets of nodes represent trees from T_{A-r_a} and T_{B-r_b}, respectively, and edges represent corresponding cohesion scores. According to Criterion (2), $g(T_{A-r_a}, T_{B-r_b}) = \max(\sum_{(T_x,T_y)\in R}(g(T_x, T_y)))$ and we conclude that $g(T_{A-r_a}, T_{B-r_b})$ corresponds to the weight of a maximum weighted matching in the above bipartite graph. □

Given Lemma 2, we can see that the following corollary is correct.

Corollary 3. *Define* $MaxMatch(X, Y) = \sum_{(T_x,T_y)\in R} g(T_x, T_y)$, *where R is a maximum weighted matching of trees in forests X and Y given* $g(T_x, T_y)$ *for any tree pair* $(T_x, T_y) \in X \times Y$. *One concludes that, for any two forests* T_A *and* T_B, $g(T_A, T_B) = MaxMatch(T_A, T_B)$.

With Lemma 1 and Corollary 3, we are able to design an efficient dynamic programming algorithm achieving the global optimum for the ontology integration problem. The pseudocode for calculating the maximum cohesion score is described in Algorithm 2, which visits ontology vertices in reverse topological order when filling up the cohesion matrix.

At the end of Algorithm 2, the cohesion matrix is filled up with optimal cohesion scores and the maximum cohesion score is saved at entry (0, 0), as described by Theorem 4.

Theorem 4. *It holds that* $cohesion\_matrix(i, j) = g(T_{A-i}, T_{B-j})$ *and* $cohesion\_matrix(0, 0) = g(T_A, T_B)$.

Proof. We will prove this theorem by mathematical induction.

Let $|V(T_A)| = n$ and $|V(T_B)| = m$; it is easy to see that cohesion_matrix(n, m) is optimal because n and m correspond to leaf nodes in the topological order. Thus T_{A-n} and T_{B-m} are empty sets and cohesion_matrix$(n, m) = M_{AB}(n, m) = g(T_n, T_m)$.

When $i = n$ and $j < m$, according to Lemma 1, to integrate T_n with T_j, either T_n (the leaf vertex) is merged with T_j or T_n is integrated with T_{B-j}. The maximum score of integrating T_n with T_{B-j} is available at the time cohesion_matrix(n, j) is being calculated because of the reverse topological visit. Thus, according to both Lemma 1 and Corollary 3, we conclude that $g(T_n, T_j) = \max(M_{AB}(n, j), \text{MaxMatch}(T_n, T_{B-j})) =$ cohesion_matrix(n, j). Similarly, we can conclude that $g(T_i, T_m) = \max(M_{AB}(i, m), \text{MaxMatch}(T_{A-i}, T_m)) =$ cohesion_matrix(i, m).

When $i < n$ and $j < m$, again, according to Lemma 1 and Corollary 3, we have $g(T_i, T_j) = \max(\text{score}, \text{score}_a, \text{score}_b) =$ cohesion_matrix(i, j). Due to the reserve topological visit, MaxMatch(T_{A-i}, T_{B-j}), MaxMatch(T_{A-i}, T_j), and MaxMatch(T_i, T_{B-j}) are available at the time cohesion_matrix(i, j) is being calculated. □

Given the definition of $g(T_A, T_B)$, Theorem 4 in fact proves that the proposed approach achieves the global optimum. However, the global optimal solution is built upon

the maximum weighted matching (recall Corollary 3). As discussed in Section 2.1.3, the maximum weighted matching is time-consuming and therefore we propose an approximation solution in that section.

Although Algorithm 2 builds the cohesion matrix with optimal cohesion scores, it does not construct the integrated ontological knowledge structure. We may save the ontology integration details along with the cohesion scores. However, that will cost $O(n^3)$ memory space (assuming each ontology has $O(n)$ vertices) and significantly reduce the capacity of the algorithm in handling large ontology integrations. Quite interestingly, we find that it is not necessary to save the integration details in order to construct the integrated ontology. The construction can be done by a process reverse to the construction of cohesion matrix, as described in Algorithm 3.

Algorithm 3 uses the cohesion matrix constructed by Algorithm 2 and builds the integrated ontology tree still by following Lemma 1 and Corollary 3, but in a reverse way of Algorithm 2. The construction is performed in a Breadth-First fashion which uses a queue q to maintain triples. Each triple (a, b, c) is an association of three elements: a is the matched vertex from ontology T_A; b is the matched vertex from ontology T_B; and c is their parent on the merged ontology T_{AB}. By following the basic idea of the proof of Theorem 4 we can show that Algorithm 3 builds an optimal integrated ontology with the cohesion matrix provided from Algorithm 2. We omit the proof for succinctness.

2.1.3. Time Complexity Analysis and an Approximation Solution. Assume an ontology size is $O(n)$. The cohesion matrix has $O(n^2)$ entries to fill up. The computation for each entry is a matching whose time complexity depends on the implementation. The maximum weighted matching takes $O(n^3)$ using the famous Hungarian algorithm [21], and although it achieves optimum, it is too costly for large ontologies. The maximal weighted matching, however, takes $O(n^2 \log n)$ time and, more importantly, results in an overall $O(n^2 \log n)$ time complexity for Algorithm 2. The analysis is given in the following. For each matching, the algorithm will access previously filled entries and each entry will be accessed only once and be involved only once in a sorting of $O(\log n)$ time. This is because each entry corresponds to two vertices whose cohesion score will be accessed when calculating the cohesion score of their parents. Thus, we conclude that the total time complexity of calculating the cohesion matrix using maximal weighted matching is $O(n^2 + n^2 \log n) = O(n^2 \log n)$. Since building the new ontology has the same time complexity as building the cohesion matrix, this is also the total time complexity for integrating two ontologies by maximal weighted matching. The maximal weighted matching also has a guaranteed lower bound on the results. It achieves a (1/2)-approximation solution (i.e., the overall cohesion score will be at least 1/2 of the optimal cohesion score) as pointed out in [22].

Since the maximal weighted matching results in an overall good performance on time complexity and approximation rate, we used the maximal weighted matching in our empirical study for Algorithms 2 and 3. Readers may also choose other matching algorithms (such as the one described in [22]) to achieve slightly better approximation rates. However, the weighted matching is a replaceable module for our algorithms and it is not the focus of this work to build a fast and close-to-optimal weighted matching algorithm.

Compared to the time complexity of integrating ontologies by the dynamic programming approach as described in Algorithms 2 and 3, the heuristic approach described at the beginning of this section also has $O(n^2)$ in the worst case. However, we conjecture that the heuristic approach has a much smaller average time complexity because, in each step, the heuristic approach may exclude a large number of matching opportunities.

2.2. Integrating Multiple Ontologies. In the previous section we proposed methods for integrating two ontologies. In some biomedical applications [23, 24], we are interested in the associations involving more than two objects. Integration of multiple ontologies of these objects will generate an innovative view on these complex relationships. Similar to the basic problem formulation, we can formulate the multiple ontology integration as follows.

Given k ontology trees $T_1, T_2, \ldots, T_k$ and a closeness matrix M_{ij} for any two trees T_i and T_j, how can we efficiently generate an integrated ontology tree $T_{1,2,\ldots,k}$ meeting the following criteria?

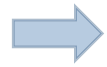
(1) For any two vertices x and y in a tree T_i, their lowest common ancestor $\mathrm{LCA}_{T_i}(x, y)$ is contained by $\mathrm{LCA}_{T_{1,2,\ldots,k}}(x, y)$.

(2) It holds that $\mathrm{argmax}_{T_{1,2,\ldots,k}} f(T_{1,2,\ldots,k}) = \sum_{X \in V(T_{AB})} \sum_{u \in X, v \in X, \sigma(u) < \sigma(v)} M_{\sigma(u),\sigma(v)}(u, v)$. Here $M_{\sigma(u),\sigma(v)}(u, v)$ is the entry value in the closeness matrix for two vertices u and v (one from tree $T_{\sigma(u)}$ and the other from tree $T_{\sigma(v)}$) contained in the node X. For a vertex v from an original ontology, $\sigma(v)$ is defined as its original ontology ID.

Again, we name the function $f_{T_{1,2,\ldots,k}}$ the cohesion of the integrated ontology $T_{1,2,\ldots,k}$. For each node X in the integrated ontology, we define its weight as $weight(X) = \sum_{u \in X, v \in X, \sigma(u) < \sigma(v)} M_{\sigma(u),\sigma(v)}(u, v)$. Correspondingly, we define function $g(T_1, T_2, \ldots, T_k) = \max_{T_{1,2,\ldots,k}} (\sum_{X \in V(T_{1,2,\ldots,k})} weight(X))$ as the maximum cohesion function for integrating the ontologies $T_1, T_2, \ldots, T_k$. As we can see in the above formulation, the overall cohesion score of integration is the summed weight of each node, which is the sum of pairwise closeness scores.

The formulation of multiple ontology integration is similar to the basic version, and it is not difficult to show that optimal structures described in Lemmas 1 and 2 can be extended to a high dimension. However, the extension of algorithms described in Section 2.1.2 for integrating two ontologies is not feasible for solving the multiple ontology integration. This is because if we need to extend Algorithms 2 and 3 to this problem, we need to build a cohesion matrix of k dimensions. It implies that we need at least $O(n^k)$ operations to fill up the score matrix assuming the size of an ontology

	A	B	C	D
A	—	10	6	8
B	10	—	7	5
C	6	7	—	9
D	8	5	9	—



	AB	C	D
AB	—	*	*
C	*	—	9
D	*	9	—

(*) Entries to update

Figure 4: An example of the InterOntology matrix's change at the first iteration in Algorithm 4 for integrating four ontologies.

build the $k \times k$ InterOntology matrix;
for $i = 1$ to $k - 1$ **do**
  identify the active tree pair $\langle T_X, T_Y \rangle$ that corresponds to the highest score in the InterOntology matrix;
  integrate T_X and T_Y into $T_{X,Y}$;
  mark T_X and T_Y as inactive;
  update relationship matrices;
  update InterOntology matrix;
end for
return $T_{1,2,\ldots,n}$;

Algorithm 4: GreedyMultiInt($\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$).

is $O(n)$. This is clearly not acceptable for high dimensional ontology integration.

2.2.1. Greedy Approach. From the above discussion we can see that direct extension of Algorithms 2 and 3 for integrating two ontologies is practically not feasible for integrating a large number of ontologies. However, we can still use these algorithms for integrating multiple ontologies, by iteratively integrating two ontologies and generating a new closeness matrix. Given the ontologies $T_1, T_2, \ldots, T_k$, we can first integrate T_1 and T_2 into $T_{1,2}$ and then build the closeness matrix between $T_{1,2}$ and T_3 using the relationship matrices between T_1 and T_2 and between T_1 and T_3. Specifically, assume X is a node on the integrated ontology $T_{1,2}$, and X contains a vertex a from T_1 and a vertex b from T_2. Then, the entry (X, c) of the closeness matrix between $T_{1,2}$ and T_3 is $M_{T_{1,2},T_3}(X, c) = M_{T_1,T_3}(a, c) + M_{T_2,T_3}(b, c)$. After the new closeness matrix is generated, we can continue integrating $T_{1,2}$ and T_3 into $T_{1,2,3}$ and generating another new closeness matrix. We will eventually get the integrated ontology $T_{1,2,\ldots,k}$ by repeating the above process. To facilitate the following discussions, we name the above approach the *basic multiple integration approach*.

Although the basic multiple integration approach can finish integrating multiple ontologies, it blindly integrates ontologies without using any cohesion information between ontologies that may lead to a better integration result. To improve the basic multiple integration, we propose a greedy approach that uses the cohesion information between ontologies to guide the integration. The basic steps of the greedy approach are outlined in Algorithm 4. To facilitate the understanding of Algorithm 4, we use Figure 4 to illustrate an example of the InterOntology matrix's change at the first iteration of integrating four ontologies A, B, C, and D.

The key idea in Algorithm 4 is to maintain an InterOntology matrix which guides the integration. Initially, this matrix is filled with the overall cohesion score of every pair of ontologies. In each step, this matrix is updated with overall cohesion scores between newly integrated ontology and existing active ontologies. The integration will take place between two active ontologies which have the highest score in the InterOntology matrix.

When we used the overall cohesion score between two ontologies to update the InterOntology matrix, we observed an interesting phenomenon that the integration in most cases is a process continuously expanding an integrated ontology. Consequently, the greedy approach is likely to yield a result similar to the basic approach.

This phenomenon can be explained by the definition of maximum cohesion function, which takes into account all pairwise closeness between merged terms. Thus, the more ontologies contained in an integrated ontology are, the more likely it will have high overall cohesion scores with other ontologies. As a result, it creates unfairness for the integration selection. To fix this issue, we use the adjusted overall cohesion scores in updating the InterOntology matrix as follows.

Given an ontology T_X and an ontology T_Y where X and Y are nonempty sets of ontology IDs, we define the adjusted cohesion score between T_X and T_Y as $AdjCoh(T_X, T_Y) = \sum_{Z \in V(T_{XY})} \sum_{x \in Z, y \in Z, \sigma(x) \in X, \sigma(y) \in Y} M_{\sigma(x),\sigma(y)}(x, y)/|X||Y|$, where T_{XY} is the integrated ontology built by Algorithms 2 and 3. The adjusted cohesion score is in fact the weight increase by integrating T_X and T_Y, divided by the size of X times the size of Y; that is, $AdjCoh(T_X, T_Y) = \sum_{Z \in V(T_{XY})}(weight(Z)) - \sum_{X \in V(T_X)}(weight(X)) - \sum_{Y \in V(T_Y)} weight(Y)/|X||Y|$. For each node merging, closeness scores will be added to the total weight when the merging takes place between vertices from ontology set X and vertices from ontology set Y. Thus, the weight increase by integrating T_X and T_Y is proportional to the number of ontologies in X times the number of ontologies in Y, and consequently we averaged the weight increase by $|X||Y|$.

2.2.2. Fast Approximation Algorithm. Although the basic multiple integration and the greedy multiple integration approaches discussed above are able to integrate multiple ontologies, none of them provide any guarantee on the results

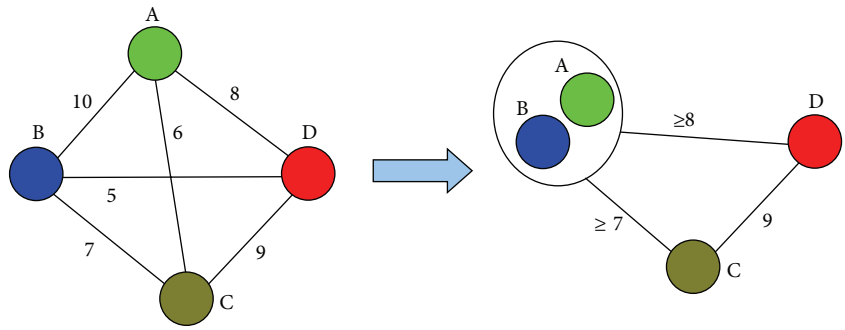


Figure 5: An illustration of vertex/edge contraction and weight updates in an iteration of Algorithm 5. Each vertex represents an ontology.

```
for i = 1 to k − 1 do
  for j = i + 1 to k do
    push ⟨g(T_i, T_j), i, j⟩ into a set S ordered by the first element in descending order;
  end for
end for
while |S| > 0 do
  ⟨z, x, y⟩ = pop(S);
  if adding (x, y) does not form a cycle in 𝒢 then
    adding (x, y) to 𝒢;
    integrate T_C(x) and T_C(y) into T_C(x)∪C(y); {C(v) is a set of vertices including v that form a connected component in 𝒢.}
  end if
end while
return T_1,2,...,n;
```

Algorithm 5: FastMultiInt($\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$).

in comparison with the optimal solutions. By studying the maximum cohesion scores between ontologies under a graph setting, we identified an approximation structure and developed an approximation algorithm for integrating multiple ontologies. We name it fast approximation algorithm because it not only has a lower bound on the results, but also runs faster than the greedy multiple integration algorithm proposed above.

The fast approximation algorithm for integrating multiple ontologies is sketched in Algorithm 5. It only calculates the maximum cohesion score between every pair of ontologies once during the initial stage, and uses this information throughout the integration process even after it becomes stale. More importantly, this approach not only saves the time for recalculating the maximum cohesion scores, but also provides a lower bound guarantee as stated in Theorem 5, whose correctness is built on two important lemmas (Lemmas 6 and 7) which will be described subsequently.

Theorem 5. *The tree weight of the integrated tree $T_{1,2,\ldots,k}$ obtained by FastMultiInt algorithm is at least $1/(k-1)$ the weight of the optimal solution.*

Proof. We will use Lemmas 6 and 7 to prove this theorem. The proofs of Lemmas 6 and 7 are provided after the proof of this theorem. To facilitate the proof of this theorem, we build a fully connected weighted graph $\mathcal{G}$ in which each node corresponds to a tree for integration, and the weight of each edge corresponds to the weight increase (initially, this is the cohesion score) for integrating the corresponding trees. According to Lemma 6, $g(T_1, T_2, \ldots, T_k)$ (i.e., optimal cohesion score) is no more than the summed weight of edges in $\mathcal{G}$ (Claim 1).

Given $\mathcal{G}$, the integration by Algorithm 5 is a process of $k-1$ node contractions. After each contraction, the adjacent edge weights (cohesion scores) will be updated accordingly. According to Lemma 7, the weight of an updated edge will only increase over (or at least remain the same as) the maximum weight of the two contracted edges (Figure 5 provides an illustration of vertex/edge contraction and weight updates.) Thus, the overall cohesion score of the integration by Algorithm 5 is no less than the weight of the maximum spanning tree of $\mathcal{G}$ (Claim 2).

It is easy to see that the weight of a maximum spanning tree is no less than $1/(k-1)$ of the summed edge weight of $\mathcal{G}$, given the simple observation that each edge in $\mathcal{G}$ is either an edge of the maximum spanning tree or adjacent to an edge of the maximum spanning tree with an equal or heavier weight (Claim 3).

Combining Claims 1, 2, and 3, we complete the proof of this theorem. □

Lemma 6. *It holds that $g(T_1, T_2, \ldots, T_k) \leq \sum_{1 \leq i < j \leq k} g(T_i, T_j)$.*

Proof. According to the problem definition,

$$\begin{aligned} & g(T_1, T_2, \ldots, T_k) \\ & = \max\left(\sum_{X \in V(T_{1,2,\ldots,k})} weight(X)\right) \\ & = \max\left(\sum_{X \in V(T_{1,2,\ldots,k})} \sum_{u \in X, v \in X, \sigma(u) < \sigma(v)} M_{\sigma(u),\sigma(v)}(u, v)\right) \\ & = \left(\sum_{1 \le i < j \le k} f_{T_{1,2,\ldots,k}}(T_{i,j})\right) \le \sum_{1 \le i < j \le k} g(T_i, T_j). \end{aligned} \quad (3)$$

$f_{T_{1,2,\ldots,k}}(T_{i,j})$ is the cohesion score of $T_{i,j}$ whose integration is induced from $T_{1,2,\ldots,k}$. □

Lemma 7. *It holds that* $g(T_{P,Q}, T_S) - f(T_{P,Q}) \ge \max(g(T_P, T_S), g(T_Q, T_S))$.

Proof. According to the problem definition, integrating $T_{P,Q}$ with T_S will result in an integrated tree T_U where $U = P \cup Q \cup S$, and $g(T_{P,Q}, T_S) = \max_{T_U}(\sum_{X \in V(T_U)} weight(X)) = \max_{T_U}(f(T_{P,S}) + f(T_{Q,S}) + f(T_{P,Q}))$, where $T_{P,S}$, $T_{Q,S}$, and $T_{P,Q}$ are induced from T_U. Since $T_{P,Q}$ has been determined, we have $g(T_{P,Q}, T_S) = \max_{T_U}(f(T_{P,S}) + f(T_{Q,S})) + f(T_{P,Q})$. Without loss of generally, let us assume $\max(f(T_{P,S})) \ge \max(f(T_{Q,U}))$. Then, by restricting the integration between T_P and T_S in T_U following the integration that leads to $\text{argmax}_{T_{P,S}} f(T_{P,S})$, we will get a cohesion score no less than $g(T_P, T_S)$. Thus, we complete the proof for $g(T_{P,Q}, T_S) - f(T_{P,Q}) \ge \max(g(T_P, T_S), g(T_Q, T_S))$. □

2.2.3. Time Complexity Analysis. For the fast approximation algorithm (Algorithm 5), the time complexity for generating graph $\mathcal{G}$ (calculating the overall cohesion score for every pair of ontologies) is $O(k^2 n^2 \log n)$, assuming we use maximal weighted matching. Each integration will take $O(n^2 \log n)$ with an update of at most k closeness matrices which takes $O(k * n^2)$. There are at most k integrations; therefore the total time complexity is still $O(k^2 n^2 \log n)$.

Following the above analysis, we conclude that the time complexity of the greedy multiple integration algorithm is the same as the fast approximation algorithm. However, it requires updating of InterOntology matrix which takes an excessive $O(k^2 n^2 \log n)$ time. The empirical study shows that the fast approximation algorithm is much faster than the greedy multiple integration algorithm.

Finally, it is easy to see that the basic multiple integration approach takes $O(kn^2 \log n)$ time and is the fastest, but its overall cohesion scores are the worst as we will see in the empirical study.

2.2.4. Limitations. Integrating multiple ontologies may face two potential problems in real applications. First, how can we efficiently generate a closeness matrix for every pair of ontologies to be integrated? Our current method kDLS or onGrid is efficient for generating the closeness matrix for one pair of ontologies in most cases, but not efficient enough for generating closeness matrices for many pairs of ontologies. Second, not every pair of ontologies can be meaningfully integrated. It remains a problem to efficiently identify the feasibility of integrating a pair of ontologies. Therefore, the main purpose of Section 2.2 is to demonstrate that our proposed approach can be extended to integrate multiple ontologies, and we use synesthetic datasets in Section 3.3 to study the performance of algorithms proposed in Section 2.2.

3. Results and Discussion

We would like to study the performances of the proposed ontology integration methods by experiments on both real and synthetic datasets. We implemented five approaches in C++:

(1) Heuristic: heuristic approach for integrating two ontologies as described in Section 2.1.1;

(2) Approximate: approximate approach (Algorithms 2 and 3) with maximal weighted matching for guaranteeing the (1/2)-approximation rate;

(3) Basic: basic multiple integration approach as described in Section 2.2.1;

(4) Greedy: greedy multiple integration approach (Algorithm 4);

(5) FastApproximate: fast approximation multiple integration approach (Algorithm 5).

In the following, we report our study on the performances of (1) and (2) for integrating two ontologies on real datasets and (3), (4), and (5) for integrating multiple ontologies on synthetic datasets. All the experiments are carried out on a Linux cluster with 2.4 GHz AMD Opteron processors.

3.1. Integrating a Pair of Ontologies. The knowledge of drug-gene relationships is desirable in many pharmacology applications [25, 26]. By integrating the gene ontology and the drug ontology, we will be able to obtain rich information on the associations between drugs and genes under the ontology structures. Thus, in this set of experiments, we simulate real world knowledge discovery applications by integrating two real ontologies, gene ontology (GO) and National Drug File Reference Terminology (NDFRT). Both were obtained from the Unified Medical Language System (version: 2012AA). The closeness matrices between GO terms and drug terms were generated using onGrid [27] with a 4-neighborhood broadcast range (i.e., $k = 4$ with regard to [7]). onGrid follows the kDLS approach [7] and measures the closeness between two concepts based on the discovered paths (with length greater than one) between them. However, unlike kDLS, onGrid takes into consideration of concept semantic types in the closeness measurement. In the study performed in [27], the advantages of onGrid over kDLS are well illustrated.

The overall cohesion scores of Heuristic and Approximate on integrating GO and NDFRT are listed

Table 1: Cohesion scores of integrating real datasets.

Depth	GO term number	NDFRT term number	Cohesion scores			
			Heuristic $\beta = 1$	Heuristic $\beta = 6$	Heuristic $\beta = 100$	Approximate
3	66	6004	0.0505331	0.229958	0.21999	1.24696
4	710	6972	0.290392	0.284363	1.46585	9.3835
5	5355	14582	0.285923	1.37714	0.528289	33.9056
6	16231	32841	0.307941	0.341588	0.673406	74.0293

Table 2: Top 5 matched terms by the Approximate algorithm (depth = 6).

Rank	GO terms	NDFRT terms	Closeness score
1	C1135918 smooth muscle contractile fiber	C0282606 muscle neoplasms	1.63205
2	C0010813 cytokinesis	C0086376 GTP-binding proteins	1.15967
3	C0027747 axon terminus	C0030584 parovarian cyst	1.13352
4	C1155065 T cell activation	C0007082 carcinoembryonic antigen	1.00879
5	C0007595 cell growth	C0294028 BRCA2 protein	0.945284

Table 3: Top 5 matched terms by the Heuristic algorithm (depth = 6, $\beta = 100$).

Rank	GO terms	NDFRT terms	Closeness score
1	C0031845 biological_process	C0042890 VITAMINS	0.244862
2	C1166607 cellular_component	C1657248 apoptosome	0.142857
3	C0027540 tissue death	C0065932 MENADIOL	0.0434219
4	C0025519 metabolic process	C0042849 VITAMIN B	0.0370248
5	C0030012 oxidation-reduction process	C0027996 NICOTINIC ACID	0.0327109

in Table 1. To observe the integration over the ontology size change, in each experiment we use the ontology tree structure from the root to the specified depth (first column in Table 1) for integration. The sizes of the ontology terms involved in the integration are listed in the second and third columns of Table 1.

Recall in Section 2.1.1; $\beta^{\text{rdepth}(b)}$ is used to regulate the selection of vertices from high depths. Thus, we tested the Heuristic under $\beta = 1$ (depth information is nullified) and $\beta = 100$ (the vertex depth plays a critical role in the selection). Since nonleaf vertices in these datasets have around 6 children on average, we heuristically add a set of experiments by setting $\beta = 6$ so that $\beta^{\text{rdepth}(b)}$ will be close to the number of vertices excluded from the future integration.

From Table 1, we can see that Heuristic performs better when using the depth information to regulate the selection of vertices. However, Approximate is much better than Heuristic at all settings. Compared to the best cohesion scores of Heuristic in each row of Table 1, Approximate constructs an integrated ontology with the overall cohesion score ranging from 5.4 times to 109.9 times that of Heuristic. This clearly demonstrates the effectiveness of the proposed Approximate approach. Nevertheless, the heuristic approach has a much faster average running time as a result of excluding a large number of matching opportunities in each step.

Although the running time of Approximate is longer than Heuristic, it takes less than two hours to finish integrating two ontologies with about $16k$ and $33k$ vertices. Most of the biomedical ontologies are smaller than or similar to these sizes and Approximate approach will benefit the association study of these ontologies. For extremely large ontology pairs in which Approximate is unable to finish the integration within a reasonable time, Heuristic may provide a quick view on their integration.

3.2. Understanding the Merged Ontology Terms. To understand what terms are merged in integrating real ontologies, we use the integration of GO and NDFRT at depth 6 as an example. Tables 2 and 3 list the top 5 pairs of merged terms (sorted by their closeness scores) by Heuristic and Approximate, respectively. As mentioned above, these scores are from the closeness matrix generated by onGrid based on the discovered paths between them. For example, "C1155065:T-Cell Activation – – is_physiologic_effect_of_chemical_or_drug – – > C0393002: Carcinoembryonic Antigen Peptide 1 – – has_target – – > C0007082:Carcinoembryonic Antigen" is such a path.

From Tables 2 and 3 we can observe that the Approximate algorithm merges terms with much higher similar scores than the Heuristic algorithm. Quite interestingly, we observed that the top ranked merging in Table 3 is between "biological_process" and "VITAMINS." The "biological_process" is an abstract term which is very close to the root of the GO ontology. Such a fact suggests that the top level terms will likely preempt the merging choices over their descendants. As a result of this greedy approach,

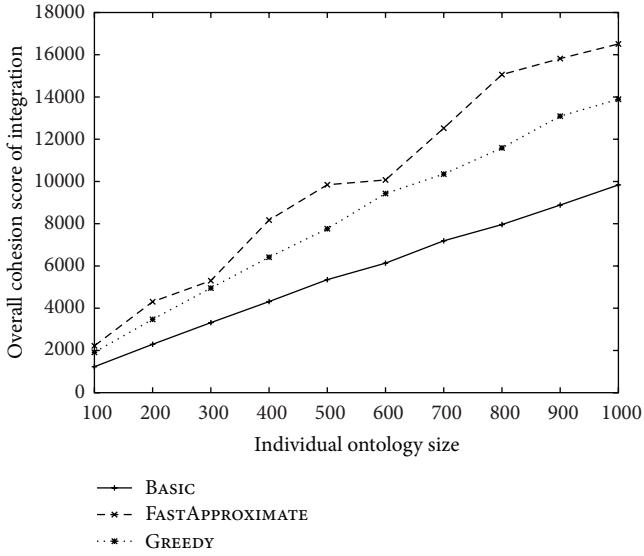


Figure 6: The change of overall cohesion score over the increase of the size of each ontology. The number of ontologies is fixed at 10.

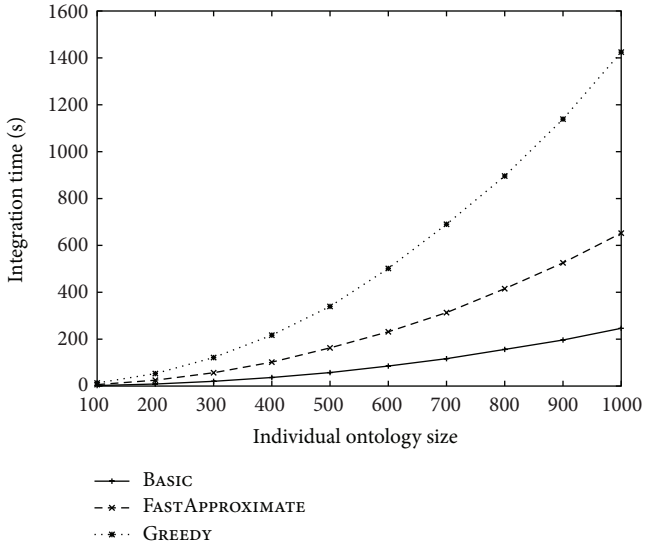


Figure 8: The change of integration time over the increase of the size of each ontology. The number of ontologies is fixed at 10.

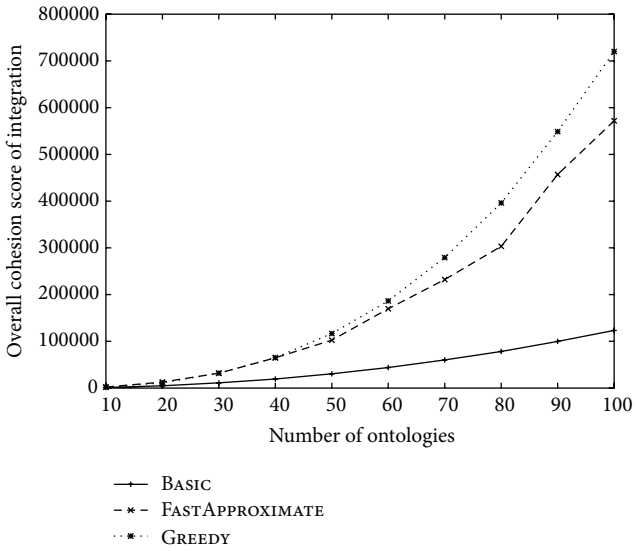


Figure 7: The change of overall cohesion score over the increase of the number of ontologies. The size of each ontology is fixed at 100.

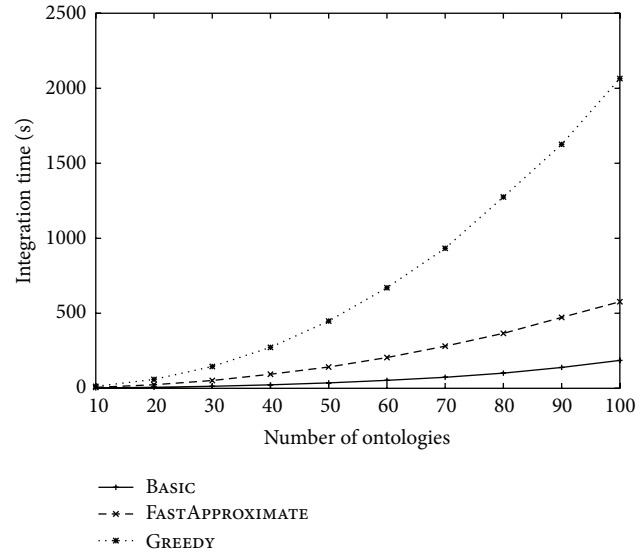


Figure 9: The change of integration time over the increase of the number of ontologies. The size of each ontology is fixed at 100.

the Heuristic algorithm will end at a local optimum which is far from being optimal.

A snapshot of ontology integration by Approximate as shown in Figure 10 provides a good insight on the algorithm work. In each bracket of two merged terms, the left part is the closeness score and the right part is the cohesion score. We can observe that most closeness scores are zero or close to zero, while the corresponding cohesion scores are much higher. This is understandable because the snapshot is primarily on the top level terms of both ontologies. For these terms, they have a large number of subclass (descendant) terms, and optimizing the integration of their subclass terms far outweighs integrating of themselves. The result of such integration provides novel knowledge of association between ontology terms. That is, even if two terms are not that close according to some closeness measurement, they can be structurally associated under their ontology context. For example, the GO term "biological_process" is merged with the NDFRT term "chemical ingredients"; even their closeness score is zero from the onGrid output. However, such integration is interesting because it shows that the merging is trying to link the chemical compounds with the biological/cellular processes so that corresponding associations between the cellular processes and chemical structures can be established. This demonstrates the purpose of integrating two ontologies, that is, identifying associations with respect to both term similarities and structural contexts.

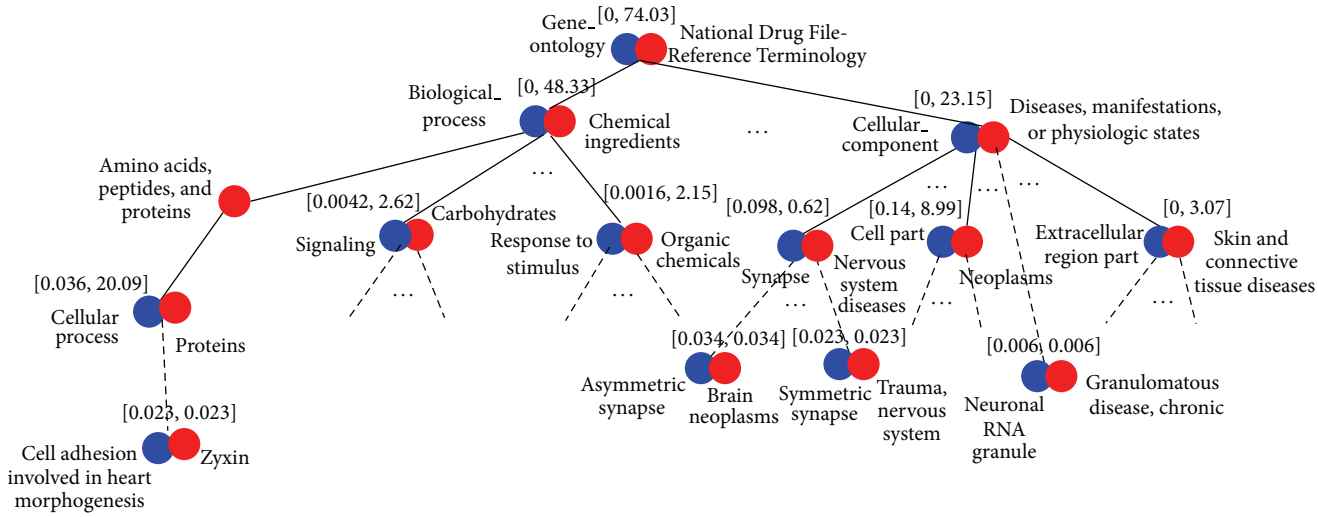


Figure 10: A snapshot of ontology integration between GO and NDFRT by Approximate (depth = 6). Blue nodes are terms from GO and red nodes are terms from NDFRT. Solid lines are edges and dashed lines are paths consisting of 1 or more edges. In each bracket, the left part is the closeness score and the right part is the cohesion score.

In fact, there are multiple studies to justify the structural associations seen in Figure 10, such as the association between "signaling" and "carbohydrates" [28] and the association between "extracellular region part" and "skin and connective tissue diseases" [29].

In addition, we have noticed a number of meaningful integrations between GO terms and neurological terms in the NDFRT. For example, synapse is a brain related structure and the term "symmetric synapse" is associated with "trauma," and the term "asymmetric synapse" is associated with "brain neoplasms." Similarly, it is reasonable to see that "neuronal RNA granule" is integrated with "granulomatous disease," a granule associated disease. As another example, it is very interesting to notice that "zyxin" is associated with "cell adhesion involved in heart morphogenesis" and that provides a link with the formation of heart.

The above observations suggest a novel way of using our ontology integration method to perform association studies between biomedical concepts.

3.3. Integrating Multiple Ontologies. In the following experiments we will study the performances of Basic, Greedy, and FastApproximate in integrating multiple ontologies. All the three approaches are built upon Approximate, which performs very well in the previous study for integrating two real ontology datasets.

We use two sets of synthetic datasets in this study. In the first set of datasets, we fix the number of ontologies to be 10 and vary the size of each ontology from 100 to 1000. In the second set of datasets, we fix the size of each ontology to be 100 and vary the number of ontologies from 10 to 100. All the ontologies are randomly generated by constructing a minimal spanning tree from a random matrix. The relationship matrix between every pair of ontologies is also randomly generated with entry values ranging from 0 to 1. For each experiment, we generate 10 random datasets and the results reported in the following are the average results over the 10 random datasets.

The overall cohesion scores of the three approaches over different ontology sizes and over different numbers of ontologies were reported in Figures 6 and 7, respectively. FastApproximate outperforms all the other approaches in Figure 6, which is consistent with the analysis of its approximation rate. However, Greedy slightly outperforms FastApproximate in Figure 7 especially when the ontology number is large. This is understandable because when the number of ontology (k) increases, the approximation rate (as stated in Theorem 5) decreases and becomes less significant. This result also justifies the choice of adjusted cohesion score for Greedy as described at the end of Section 2.2.1.

The integration time of the three approaches over different ontology sizes and over different numbers of ontologies was reported in Figures 8 and 9, respectively. These figures are consistent with the time complexity analysis given in Section 2.2.3. In particular, we noticed that the integration time of Greedy deteriorates sharply over the ontology number increase. In contrast, FastApproximate is much more scalable and has a time curve similar to Basic.

These results suggest that FastApproximate has the best overall performance in integrating multiple ontologies.

4. Conclusions

In this work, we started with a basic problem on integrating a pair of ontology tree structures with a given closeness matrix, and later we advanced the basic problem to the problem of integrating large number of ontologies. We proved optimal structures in the basic problem and developed both optimal and efficient approximation solutions. Although the multiple ontology integration problem has similar optimal structures, it is not feasible to extend the optimal and efficient approximation solutions for the basic problem to efficiently handle

multiple ontology integration. To tackle the challenge of integrating a large number of ontologies, we developed both an effective greedy approach and a fast approximation approach. The empirical study not only confirms our analysis on the efficiency of the proposed method, but also demonstrates that our method can be used effectively for biomedical association studies.

Conflict of Interests

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

References

[1] D. L. McGuinness, F. van Harmelen, and M. K. Smith, Owl web ontology language overview. W3C recommendation, 10(2004-03):10, 2004.

[2] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, supplement 1, pp. D267–D270, 2004.

[3] X. Wu, R. Jiang, M. Q. Zhang, and S. Li, "Network-based global inference of human disease genes," *Molecular Systems Biology*, vol. 4, no. 1, 2008.

[4] B. Linghu, E. S. Snitkin, Z. Hu, Y. Xia, and C. DeLisi, "Genome-wide prioritization of disease genes and identification of disease-disease associations from an integrated human functional linkage network," *Genome Biology*, vol. 10, no. 9, article R91, 2009.

[5] D. Botstein and N. Risch, "Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease," *Nature Genetics*, vol. 33, pp. 228–237, 2003.

[6] H. S. Pinto and J. P. Martins, "A methodology for ontology integration," in *Proceedings of the 1st International Conference on Knowledge Capture (K-CAP '01)*, pp. 131–138, ACM, 2001.

[7] Y. Xiang, K. Lu, S. L. James, T. B. Borlawsky, K. Huang, and P. R. O. Payne, "K-Neighborhood decentralization: a comprehensive solution to index the UMLS for large scale knowledge discovery," *Journal of Biomedical Informatics*, vol. 45, no. 2, pp. 323–336, 2012.

[8] J. Dutkowski, M. Kramer, M. A. Surma et al., "A gene ontology inferred from molecular networks," *Nature Biotechnology*, vol. 31, no. 1, pp. 38–45, 2013.

[9] R. Navigli, P. Velardi, and A. Gangemi, "Ontology learning and its application to automated terminology translation," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 22–31, 2003.

[10] J. Jannink and G. Wiederhold, "Thesaurus entry extraction from an on-line dictionary," in *Proceedings of the Fusion Conference*, Sunnyvale, Calif, USA, 1999.

[11] C. Papatheodorou, A. Vassiliou, and B. Simon, "Discovery of ontologies for learning resources using word-based clustering," in *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 1523–1528, 2002.

[12] D. L. Rubin, M. Hewett, D. E. Oliver, T. E. Klein, and R. B. Altman, "Automating data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and xml," *Pacific Symposium on Biocomputing*, pp. 88–99, 2001.

[13] N. F. Noy, "Semantic integration: a survey of ontology-based approaches," *ACM SIGMOD Record*, vol. 33, no. 4, pp. 65–70, 2004.

[14] S. Abels, L. Haak, and A. Hahn, "Identification of common methods used for ontology integration tasks," in *Proceedings of the 1st International Workshop on Interoperability of Heterogeneous Information Systems (IHIS '05)*, pp. 75–78, ACM, November 2005.

[15] A. Gangemi, D. Pisanelli, and G. Steve, "Ontology integration: experiences with medical terminologies," in *Formal Ontology in Information Systems*, vol. 46, pp. 98–94, IOS Press, Amsterdam, The Netherlands, 1998.

[16] N. F. Noy and M. A. Musen, "SMART: automated support for ontology merging and alignment," in *Proceedings of the 12th Workshop on Knowledge Acquisition, Modelling, and Management (KAW '99)*, Banff, Canada, 1999.

[17] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to map between ontologies on the semantic web," in *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pp. 662–673, ACM, May 2002.

[18] J. Xie, F. Liu, and S.-U. Guan, "Tree-structure based ontology integration," *Journal of Information Science*, vol. 37, no. 6, pp. 594–613, 2011.

[19] D. Calvanese, G. De Giacomo, and M. Lenzerini, "A framework for ontology integration," in *The Emerging Semantic WebSelected Papers from the First Semantic Web Working Symposium*, pp. 201–214, 2002.

[20] O. Udrea, L. Getoor, and R. J. Miller, "Leveraging data and structure in ontology integration," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '07)*, pp. 449–460, ACM, June 2007.

[21] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 1-2, pp. 83–97, 1955.

[22] D. E. Drake Vinkemeier and S. Hougardy, "A linear-time approximation algorithm for weighted matchings in graphs," *ACM Transactions on Algorithms*, vol. 1, no. 1, pp. 107–122, 2005.

[23] A. J. J. Wood, W. E. Evans, and H. L. McLeod, "Pharmacogenomics—drug disposition, drug targets, and side effects," *The New England Journal of Medicine*, vol. 348, no. 6, pp. 538–549, 2003.

[24] R. B. Altman, "Pharmgkb: a logical home for knowledge relating genotype to drug response phenotype," *Nature Genetics*, vol. 39, no. 4, article 426, 2007.

[25] R. B. Kim, D. O'Shea, and G. R. Wilkinson, "Interindividual variability of chlorzoxazone 6-hydroxylation in men and women and its relationship to CYP2E1 genetic polymorphisms," *Clinical Pharmacology & Therapeutics*, vol. 57, no. 6, pp. 645–655, 1995.

[26] T. N. Ferraro and R. J. Buono, "The relationship between the pharmacology of antiepileptic drugs and human gene variation: an overview," *Epilepsy and Behavior*, vol. 7, no. 1, pp. 18–36, 2005.

[27] A. Albin, X. Ji, T. B. Borlawsky et al., "Enabling online studies of conceptual relationships between medical terms: developing an efficient web platform," *JMIR Medical Informatics*, vol. 2, no. 2, article e23, 2014.

[28] S. Chandrasekaran, J. W. Dean III, M. S. Giniger, and M. L. Tanzer, "Laminin carbohydrates are implicated in cell signaling," *Journal of Cellular Biochemistry*, vol. 46, no. 2, pp. 115–124, 1991.

[29] J. Uitto and D. Kouba, "Cytokine modulation of extracellular matrix gene expression: relevance to fibrotic skin diseases," *Journal of Dermatological Science*, vol. 24, pp. S60–S69, 2000.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 350983, 9 pages
http://dx.doi.org/10.1155/2015/350983

Research Article

Predicting Drug-Target Interactions via Within-Score and Between-Score

Jian-Yu Shi,[1] Zun Liu,[2] Hui Yu,[2] and Yong-Jun Li[2]

[1]*School of Life Sciences, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*
[2]*School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

Correspondence should be addressed to Jian-Yu Shi; jianyushi@nwpu.edu.cn

Received 19 November 2014; Accepted 6 January 2015

Academic Editor: Liam McGuffin

Network inference and local classification models have been shown to be useful in predicting newly potential drug-target interactions (DTIs) for assisting in drug discovery or drug repositioning. The idea is to represent drugs, targets, and their interactions as a bipartite network or an adjacent matrix. However, existing methods have not yet addressed appropriately several issues, such as the powerless inference in the case of isolated subnetworks, the biased classifiers derived from insufficient positive samples, the need of training a number of local classifiers, and the unavailable relationship between known DTIs and unapproved drug-target pairs (DTPs). Designing more effective approaches to address those issues is always desirable. In this paper, after presenting better drug similarities and target similarities, we characterize each DTP as a feature vector of within-scores and between-scores so as to hold the following superiorities: (1) a uniform vector of all types of DTPs, (2) only one global classifier with less bias benefiting from adequate positive samples, and (3) more importantly, the visualized relationship between known DTIs and unapproved DTPs. The effectiveness of our approach is finally demonstrated via comparing with other popular methods under cross validation and predicting potential interactions for DTPs under the validation in existing databases.

1. Introduction

Since experimental determination of compound-protein interactions or potential drug-target interactions remains very challenging (e.g., requiring a huge amount of money and taking a very long period) [1], there is a need to develop computational methods to assist those experiments. Nowadays, the number of available drug-target interactions (DTIs) in public database, including KEGG [2], PubChem [3], DrugBank [4], and ChEMBL [5], is increasing which brings out two observations. The first one is that one drug can interact with one or more proteins. Another is symmetrically the fact that one protein can be targeted by one or more drugs. These two observations led to the formation of DTI network [6] and made it possible to utilize DTIs (approved drug-target pairs) to predict potential interactions among unapproved drug-target pairs (DTPs). The task to validate those predicted potential interactions is called drug repositioning or drug repurposing [7].

In terms of DTI network, predicting newly potential DTI is equivalent to predicting new edges in the network. Researchers developed network-based inference model (NBI) to deduce the potential interactions among unapproved DTPs in given DTI networks and further confirmed them from *in vitro* assays [7]. However, NBI cannot run the prediction for any DTP between which no reachable path (a set of consecutively connected edges) in network is available. In fact, a DTI network usually contains several isolated subnetworks. A difficult case for NBI is, for example, to predict the interaction between the drug in one subnetwork and the target in another. Besides, predicting interactions for a drug node d, the resulting targets usually bias to the target nodes of more degrees or the target nodes near to drug d.

With a different idea of regarding similarity matrices of drugs and targets as kernel matrices, kernel-based techniques of classification, such as bipartite local model (BLM) [8–10], are also popularly applied to DTI prediction. As a local classification model, for each target, BLM assigns known

DTIs and unapproved DTPs between drugs and the concerned target as positive and negative samples, respectively. Then a kernel-based classifier is built on drug similarity matrices and applied to assign confidence scores to unlabeled samples (concerned unapproved DTPs). Similarly, for each drug, another kernel-based classifier can be also built. For each drug-target pair, we need to build two classifiers of which the output scores further are aggregated as the final score [8, 9]. BLM, however, generates the biased prediction in the case of few positive samples (known DTIs). Also it cannot predict the interaction between a new drug (without linking to any known target) and a new target (without linking to any known drug) because no positive samples are available to train its classifier model. BLM-NII, an extension of BLM, recently developed a weighted strategy and integrated it into BLM to tackle the case of no positive sample available [10]. However, the biased prediction still remains when few positive samples are available. More importantly, since drug-target pairs are separately put into different classifier spaces, neither BLM nor BLM-NII is able to investigate the relationship between them. Such relationship is helpful for further predicting the potential interactions in both drug discovery and drug repositioning.

To summarize, three issues in existing predictive models are not yet solved. (1) Predicting interactions between drugs and targets occurring in isolated subnetworks of DTI network is difficult. (2) Inadequate positive samples usually cause biased local classifiers and local classification approach requires a number of classifiers. (3) The global relationship between approved DTIs and unapproved DTPs cannot be investigated in a consistent space.

Except for the predictive model, similarity measuring is another crucial factor in DTI prediction because similar drugs tend to interact with similar targets [11]. To capture pairwise similarities between drugs or targets in a better way, a topological similarity based on DTI network was proposed, such as Gaussian interaction profile (GIP) [9] and was linearly integrated into chemical structure-based similarity between drugs or protein sequence-based similarity between targets under the framework of BLM. Nevertheless, simple linear combination may not work optimally because the topological similarity is always related to the drug/target node degrees, which follow the power-law distribution [12]. In addition, for any two drugs/targets, GIP only considers the targets/drugs not interacting with them but has no consideration of the targets/drugs shared by them. So GIP may lose some information derived from those common targets/drugs between drugs/targets. Besides, since all possible values of the topological similarity proposed in GIP falls into $(0, 1]$, GIP is an incomplete similarity metric which may not adequately characterize the dissimilarity between those very different drugs/targets.

In this paper, we believe that the difference between the similarities of drugs/targets sharing targets/drugs and the similarities of drugs/target sharing no target/drug in DTI network should be statistically significant. To address above-mentioned issues, we first characterized each drug-target pair from the views of both drugs and targets, respectively. Under the publicly acceptable assumption that similar drugs tend to target similar protein receptors [11], two within-scores were presented to capture the similarities between drugs/targets sharing common targets/drugs. Based on our observation that similar drugs, in part, do not tend to target dissimilar proteins, two between-scores were also presented to capture the similarities between drugs/targets share no targets/drugs.

Subsequently, we represented each drug-target pair as a feature vector which uniformly consists of four scores, regardless of the available path between drugs and targets. Each drug-target pair was labeled as positive or negative sample, depending on whether it is an approved DTI or an unapproved DTP. The use of all DTIs can guarantee that enough positive samples can be used to train the only one global classifier. After performing principal component analysis on feature vectors, we generated a drug-target pair space which provides a visualized way to investigate the relationship between known DTIs and unapproved DTPs.

In addition, to obtain a better combination between topological similarity and chemical/sequence similarity, we proposed an adaptive combination rule instead of the former linear combination and introduced a complete metric of topological similarity of drugs/targets by considering both the targets/drugs shared by two drugs/targets and the targets/drugs interacting with none of them.

Finally, based on four benchmark datasets, we demonstrated the effectiveness of our approach, by comparing with NBI, BLM, and BLM's extensions in cross validation and predicting potential interactions in unapproved DTPs under checking in existing databases.

2. Materials and Method

2.1. Datasets. In this paper, the adopted datasets, involving targets of ENZYME, ION CHANNEL, GPCR, and NUCLEAR RECEPTOR, were originally from [13] and further used in subsequent works [8–10]. All of drug-target interactions in the original datasets were collected from KEGG database. In short, we denote the four DTI datasets as EN, IC, GPCR, and NR, respectively. The brief information of four datasets is listed in Table 1. Notably, NR (the sparest DTI network in the given datasets) contains the most proportions of isolated subnetworks and is the most difficult case to predict the potential DTI [10] because it has most the proportion of unreachable paths between drugs and between targets. More details can be found in the original work [13].

2.2. Drug Similarity and Target Similarity. The metrics of drug similarity and target similarity popularly adopted in former methods are chemical structure-based similarity and protein sequence-based similarity, respectively [8–10]. By representing a chemical structure as a graph, the chemical structure similarity between two drugs is defined as $S_d^{\text{chem}}(d_u, d_v) = |d_u \cap d_v|/|d_u \cup d_v|$, where $|\cdot|$ denotes the number of nodes in graph, $d_u \cap d_v$ is the maximal common subgraph between d_u and d_v, and $d_u \cup d_v$ is their union [14]. The protein sequence similarity between two targets is calculated by sequence alignment and is defined as $S_t^{\text{seq}}(t_u, t_v) = \text{align}(t_u, t_v)/\sqrt{\text{align}(t_u, t_u)\text{align}(t_v, t_v)}$, where

Table 1: Four datasets used in this work.

Dataset name	#Drugs	#Targets	#Interactions	Proportion of unreachable paths between drugs	Proportion of unreachable paths between targets
EN	445	664	2926	0.479	0.479
IC	210	204	1476	0.019	0.029
GPCR	223	95	635	0.345	0.593
NR	54	26	90	0.615	0.778

# denotes the number of drugs, targets, or drug-target interactions in dataset.

align(t_u, t_v) is the Smith-Waterman alignment score [15] between t_u and t_v.

In order to capture the real similarity between drugs/targets sharing common targets/drugs in a better way, former methods tried to propose new similarities and integrate them into abovementioned similarities. Under the framework of BLM, Gaussian interaction profile (GIP) was introduced to measure topological similarity between drugs/targets by considering DTI matrix as the adjacent matrix of DTI network [9]. However, for any two drugs/targets, GIP only considers the targets/drugs not interacting with them so that it may lose some information derived from their common targets/drugs. In addition, GIP is not a mathematically complete similarity since its similarity values fall into $(0, 1]$. So it may not be enough to characterize the dissimilarity between very different drugs/targets. Therefore, we applied a complete metric to measure the similarities between nodes of both drugs and targets, respectively, according to the DTI network. The topological similarity, named matching index (MI) [16], between drugs and the topological similarity between targets are defined as follows:

$$S_d^{\text{topo}}\left(d_i, d_j\right) = \frac{T - \left|d_i\right| - \left|d_j\right| + 2\left|d_i \cap d_j\right|}{T},$$
$$S_t^{\text{topo}}\left(t_p, t_q\right) = \frac{D - \left|t_p\right| - \left|t_q\right| + 2\left|t_p \cap t_q\right|}{D}, \tag{1}$$

where $|\cdot|$ denotes the degree of nodes and $|x \cap y|$ is the number of sharing neighbors of two nodes. For drugs, $S_d^{\text{topo}}(d_i, d_j)$ considers the proportion of their shared target nodes as well as target nodes not interacting with them. For targets, $S_t^{\text{topo}}(t_p, t_q)$ holds the similar consideration. Moreover, all possible values of MI fall into $[0, 1]$.

In former work [9], the final similarities of drug and target are usually generated by linearly combining S_d^{topo} and S_t^{topo} with S_d^{chem} and S_d^{topo}, respectively. Nevertheless, such linear combination may not work optimally because the topological similarity is always related to the node degrees which follows the power-law distribution [12].

We observed that the topological similarity always works better when those drugs link to a target node of small degree; in contrast, chemical similarity always works better when those drugs link to a target node of large degree, respectively. Consequently, we designed an adaptive combination rule to expectedly achieve better prediction for MI. For target t_p linking to g_{t_p} drugs, the similarity between d_i and d_j among g_{t_p} drugs is defined as follows:

$$S_d\left(d_i, d_j\right) = \begin{cases} S_d^{\text{chem}}\left(d_i, d_j\right) & g_{t_p} \geq u_t \\ \max\left(S_d^{\text{chem}}\left(d_i, d_j\right), S_d^{\text{topo}}\left(d_i, d_j\right)\right) & l_t < g_{t_p} < u_t \\ S_d^{\text{topo}}\left(d_i, d_j\right) & g_{t_p} \leq l_t \end{cases}$$
$$u_t = 0.5 * \max\left(\left\{g_{t_p}\right\}\right), \quad l_t = \frac{\sum_{p=1}^{T} g_{t_p}}{T}, \quad p = 1, \ldots, T. \tag{2}$$

The similarity between targets t_p and t_q can be defined in the similar way.

2.3. Within-Score and Between-Score of a Drug-Target Pair. A publicly acceptable assumption is that similar drugs tend to target similar protein receptors [11]. Based on this assumption, by considering the similarities between drugs/targets sharing common targets/drugs, we shall present two within-scores to capture them. Based on our additional observation that similar drugs, in part, do not tend to target dissimilar proteins, we shall also propose two between-scores to capture the similarities between drugs sharing no target and the similarities between targets sharing no drug respectively. The calculation of within-scores and between-scores is depicted in the following paragraphs.

Given D drugs and T targets, and their known interactions, our task is to predict potential but unapproved interactions between drugs and targets. All drug-target pairs are usually organized as an interaction matrix $A_{D \times T}$, in which $a_{ij} = 1$ when there is a known interaction between drug d_i and target t_j, and $a_{ij} = 0$ otherwise.

For drug d_i interacting with T_i targets, t_p^i and $t_q^{\tilde{i}}$ denote the target interacting and not interacting with d_i, respectively. In order to characterize the potential interaction $P(t_x, d_i)$ between drug d_i and a queried target t_x, we define within-score $C_t^w(t_x, d_i)$ and between-score $C_t^b(t_x, d_i)$ from drug view as follows:

$$C_t^w\left(t_x, d_i\right) = \max\left(\left\{S_t\left(t_x, t_p^i\right)\right\}\right), \quad p = 1, 2, \ldots, T_i,$$
$$C_t^b\left(t_x, d_i\right) = \max\left(\left\{S_t\left(t_x, t_q^{\tilde{i}}\right)\right\}\right), \quad q = 1, 2, \ldots, T - T_i, \tag{3}$$

where $S_t(t_x, t_p^i)$ is the similarity between t_x and t_p^i and $S_t(t_x, t_q^{\tilde{i}})$ is the similarity between t_x and $t_q^{\tilde{i}}$. Then, the drug-view feature of $P(t_x, d_i)$ is defined as $f(t_x, d_i) = [C_t^w(t_x, d_i), C_t^b(t_x, d_i)]$.

For target t_j interacting with D_j drugs, d_u^j is the drug interacting with it and $d_v^{\tilde{j}}$ is the drug not interacting with it. Symmetrically, from target view, we define within-score $C_d^w(d_y, t_j)$ and between-score $C_d^b(d_y, t_j)$ as follows:

$$C_d^w\left(d_y, t_j\right) = \max\left(\left\{S_d\left(d_y, d_u^j\right)\right\}\right), \quad u = 1, 2, \ldots, D_j,$$
$$C_d^b\left(d_y, t_j\right) = \max\left(\left\{S_d\left(d_y, d_v^{\tilde{j}}\right)\right\}\right), \quad v = 1, 2, \ldots, D - D_j, \tag{4}$$

where $S_d(d_y, d_u^j)$ is the similarity between d_y and d_u^j and $S_d(d_y, d_v^{\tilde{j}})$ the similarity between d_y and $d_v^{\tilde{j}}$. Again, the target-view feature of the potential interaction $P(d_y, t_j)$ is defined as $g(d_y, t_j) = [C_d^w(d_y, t_j), C_d^b(d_y, t_j)]$. Consequently, for the pair (d_y, t_x), we can obtain a combined feature vector:

$$\begin{aligned} F\left(d_y, t_x\right) &= \left[f\left(t_x, d_y\right), g\left(d_y, t_x\right)\right] \\ &= \left[C_t^w\left(t_x, d_y\right), C_t^b\left(t_x, d_y\right), C_d^w\left(d_y, t_x\right), C_d^b\left(d_y, t_x\right)\right]. \end{aligned} \tag{5}$$

2.4. Types of Interactions. Totally, we group all interactions into four types according to DTI network (Figure 1): multiple, drug-centered, target-centered, and single interacting motifs. The summary of their counts in four adopted datasets can be found in Table S1 in Supplementary Material available online at http://dx.doi.org/10.1155/2015/350983.

Either the target or the drug of a multiple interaction has >1 links to drugs or targets, respectively. The target of a drug-centered interaction has only one link to the drug interacting with >1 targets. The drug of a target-centered interaction has only one link to the target interacting with >1 drugs. Both the target and the drug of a single interaction only link to each other. A single interaction is usually newly approved [6]. The drug-target pairs in multiple motif are just shown in formula (5) in previous section. The drug-target pairs involving in drug-centered, target-centered, and single motifs are the special cases of multiple motif and are shown as follows:

$$\begin{aligned} F_d\left(d_y, t_x\right) &= \left[C_t^w\left(t_x, d_y\right), C_t^b\left(t_x, d_y\right), \text{null}, C_d^b\left(d_y, t_x\right)\right], \\ F_t\left(d_y, t_x\right) &= \left[\text{null}, C_t^b\left(t_x, d_y\right), C_d^w\left(d_y, t_x\right), C_d^b\left(d_y, t_x\right)\right], \\ F_s\left(d_y, t_x\right) &= \left[\text{null}, C_t^b\left(t_x, d_y\right), \text{null}, C_d^b\left(d_y, t_x\right)\right], \end{aligned} \tag{6}$$

where null means that the score cannot be calculated directly. We adopted a bottom-line strategy to cope with the null cases by assigning ones to null entries.

With the representation of feature vector, we can map all drug-target pairs, including the pairs between new drugs and new targets, into the same space regardless of whether the drug and the target are in the same subnetwork or not.

2.5. Drug-Target Pair Space. To check whether or not known interactions and unapproved pairs can be classified well in certain dimensions, we made the distributions of C_t^w, C_d^w, C_t^b, and C_d^b scores in feature vectors by histograms for four types of DTIs. As an illustration, the score distributions of four motifs of GPCR dataset [13] are shown in Figure 2. The distributions of all datasets can be found in Figures S1, S2, S3, and S4.

Known DTIs and unapproved DTPs show separations in terms of distributions of four scores. That is to say, they can be classified in certain dimensions (scores). In detail, (1) for multiple motifs (Figure 2(a)), known interactions (purple) and unapproved DTPs (cyan) can be separated significantly by C_t^w, moderately separated by either C_t^b or C_d^w, and almost mixed together in terms of C_d^b. (2) For drug-centered motifs whose C_d^w is unavailable (Figure 2(b)), C_t^w, C_t^b, and C_d^b show the best, the moderate, and the worst separations, respectively. (3) Likewise, for target-centered motifs whose C_t^w is unavailable (Figure 2(c)), C_d^w shows the best separation while neither C_d^b nor C_t^w provides an acceptable separation. (4) Single motifs only show C_t^b and C_d^b which both provide moderate separations (Figure 2(d)).

In terms of C_t^w and C_d^w, the separability of distributions between known interactions and unapproved DTPs denotes how their distribution meets the popular assumption that similar targets/drugs tend to interact with similar drugs/targets. Our results show that both C_t^w and C_d^w can follow the assumption well and the former is better than the latter.

On the other hand, both C_t^b and C_d^b cannot provide a good separability between known interactions and unapproved pairs. However, they follow our observation that similar drugs, in part, do not tend to target dissimilar proteins. More importantly, in the case of meeting our observation, C_t^b and C_d^b may help in prediction when they are combined with C_t^w and C_d^w together.

Therefore, integrating all four scores together by combination, such as principal component analysis (PCA), can hopefully generate a better separation because known DTIs and unapproved DTPs can be classified in individual dimensions. After performing PCA on these four scores, we showed a space of drug-target pairs on the first three principal components (in Figure 3). In the space, the greatly significant separation between known interactions and unapproved drug-target pairs is observed.

3. Result and Discussion

In this section, we shall first demonstrate the effectiveness of our topological similarity metric and our adaptive combination of similarities, compare our approach with other popular methods, including NBI [7] and BLM [8] and its

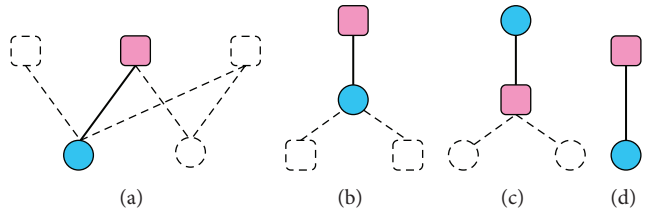


Figure 1: Topological motifs in drug-target network. (a) Multiple, (b) drug-centered, (c) target-centered, and (d) single pairs. Drugs and targets are denoted by circle nodes and rounded squares nodes, respectively. The pairs between concerned drugs (blue) and concerned targets (pink) are denoted by thick lines. The interactions between concerned nodes (filled by colors) and other nodes (hollow) are represented by dotted lines.

Table 2: Comparison between topological similarities.

	GIP (AUC/AUPR)	MI (AUC/AUPR)
BLM	**0.662/0.321**	**0.762/0.434**
Ours	**0.918/0.757**	**0.949/0.786**

extensions BLM-GIP [9] and BLM-NII [10], build a drug-target interaction space by PCA to elucidate the relationship between known DTIs and unapproved DTPs afterwards, and finally utilize the space to predict the potential interactions for DTPs.

By applying PCA on feature vectors of all drug-target pairs, we used the distances of both known interactions and unapproved pairs to the origin as the confidence scores for both validating the performance of our approach and predicting potential drug-target interactions (more details in Section 3.3). Besides, the popular measurements including area under the curve (AUC) and area under the precision-recall curve (AUPR) [17] were used to assess the computational effectiveness of approaches.

3.1. The Effectiveness of New Similarity and New Combination. To illustrate why our approach achieved better results, we first compared GIP similarity and our MI similarity under BLM framework and our approach, respectively. Using the topological similarities only, we selected the sparsest DTI network (NR dataset) from the work [8] to perform the comparison (Table 2). The results demonstrate that our new topological similarity is better than GIP similarity.

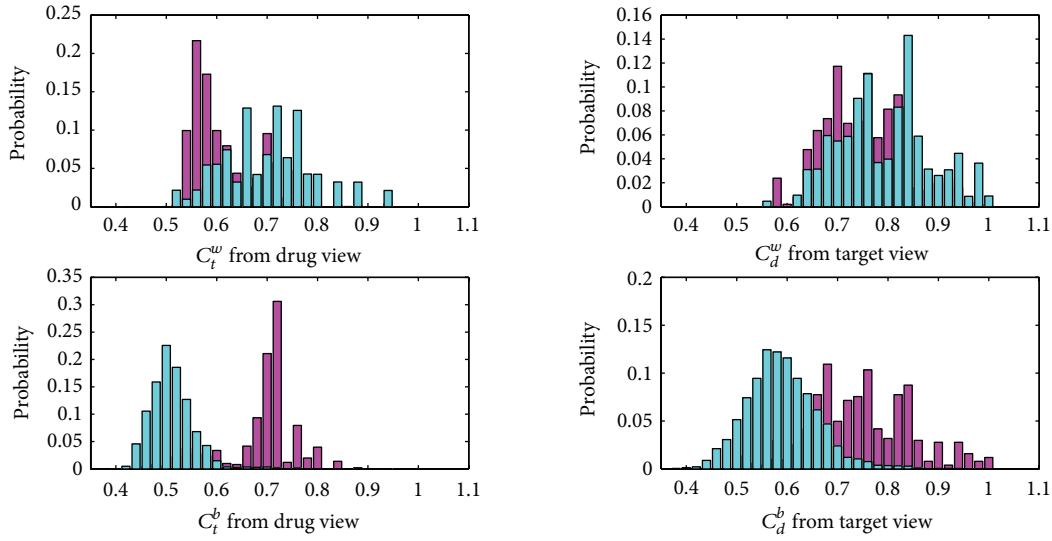
Then, we also applied linearly weighted combination to integrate MI with chemical structure similarity/sequence similarity in our approach, respectively. In terms of the values of AUC and AUPR, the linear combination achieved 0.977 and 0.826 while the adaptive combination achieved 0.982 and 0.949. Again, our adaptive combination is better than the linear combination.

3.2. Comparison with Other Methods. To validate the effectiveness of our approach, we made a comparison with other approaches [7–10] which adopted the same datasets [13] (see also Section 2.1), the same testing strategy (leave-one-out cross validation, LOOCV), and the same assessment (AUC and AUPR) [17]. First, we run predictions with only chemical similarities of drugs and sequence similarities of targets and compared the results with those of BLM and BLM-GIP. Then after integrating topological similarities, our approach compared with NBI [7], BLM-GIP, and BLM-NII. All results on four datasets are listed in Table 3. In terms of AUC, our approach outperforms on all datasets. In terms of AUPR, our approach has about 7%~10% increase on EN, GPCR, and NR, though it shows ~5% decrease on IC when compared with BLM-NII. Totally, the proposed approach has better predicting performance.

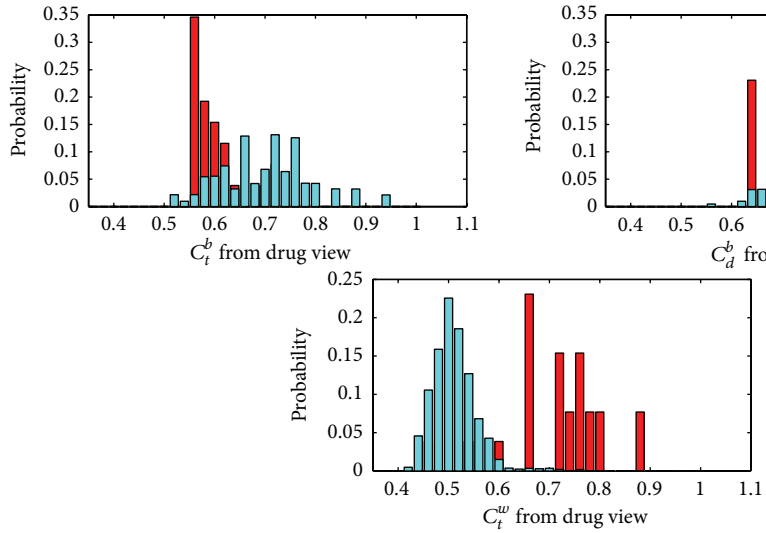
Moreover, our approach has other advantages. First, our approach holds a sufficient number of positive samples (all known DTIs) even if the number of negative samples is large, while BLM may suffer from biased classifier models since each of its local models is trained by few positive samples (even 0 or 1 sample sometimes). Then, our approach only needs to train only one classifier whereas BLM and its extensions need to build many classifiers accounting for all targets and all drugs. Last but most importantly, with the representation of feature vector, we are able to put all drug-target pairs, including the pairs between new drugs and new targets, into the same space regardless of whether the drug and the target in the concerned pair are in the same subnetwork or not. Consequently, our approach is generally superior to other former approaches.

3.3. Drug-Target Pair Space and Its Application to Find Potential Interactions. After performing PCA on feature vectors, we represented all DTPs as points shown by their first three principle components (denoted by X, Y, and Z in Figure 3, resp.). Approved DTPs (DTIs) and unapproved DTPs show two separating groups. The unapproved DTPs (cyan crosses) gather around the origin in a sphere-like shape while the known DTIs were apart from them. Particularly in Figure 3(a), three clusters of interaction motifs are found. The cluster in left contains drug-centered motif (red circles) and multiple motifs (purple squares), and the lower cluster in right comprises target-centered motifs and multiple motifs and the upper cluster in right is composed of all four types of motifs.

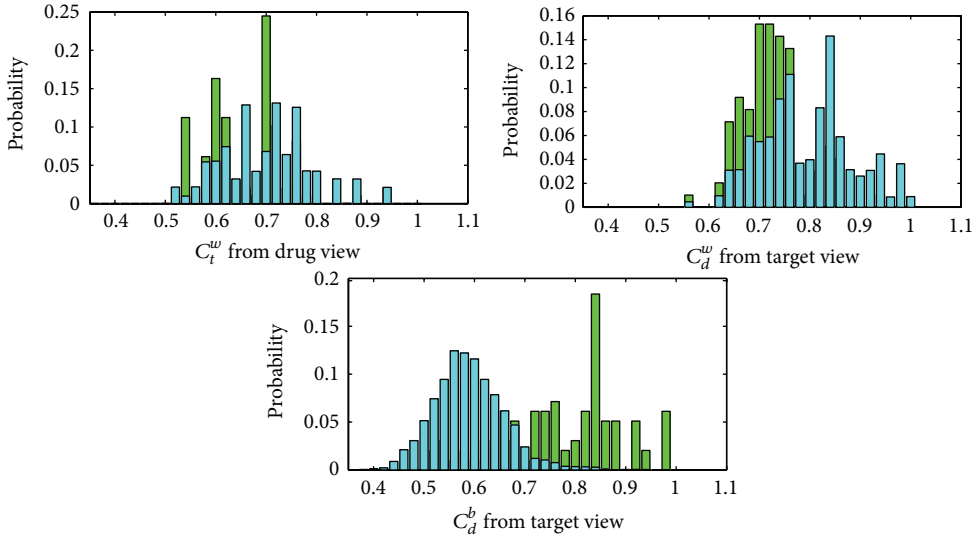
The significant distribution of DTPs in the space allows us to visually investigate the relationship between known DTIs and unapproved DTPs. Therefore, after calculating the distances of all pairs to the origin, we are not only able to build classifiers by training a specific threshold of the distances when testing the performance of our proposed method



(a) Multiple motifs

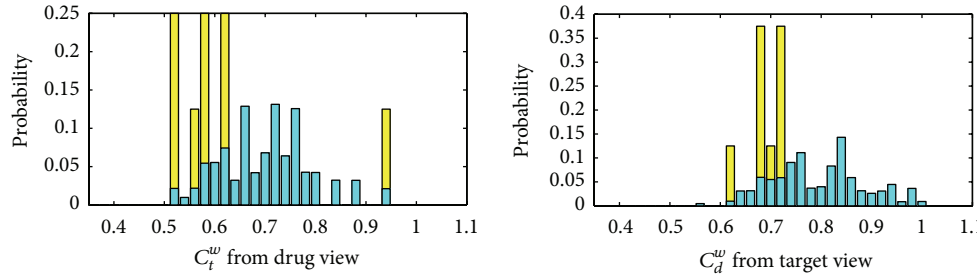


(b) Drug-centered motifs



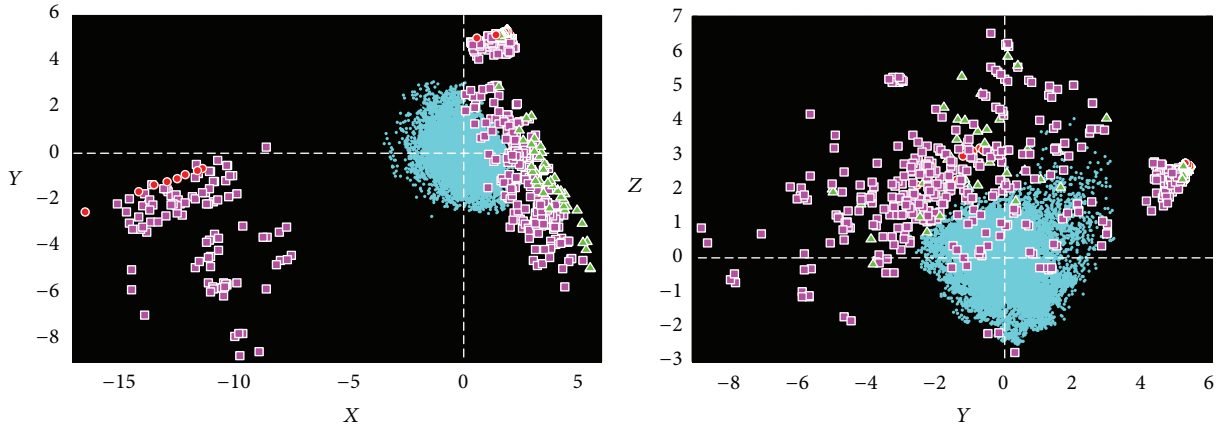
(c) Target-centered motifs

FIGURE 2: Continued.



(d) Single motifs

FIGURE 2: The distributions between known interactions (four types of motifs) and unapproved drug-target pairs. All histograms were generated by sorting scores into specific bins from 0.35 to 1.1. The x-axis in each histogram represents the bins with the intervals of 0.02. The y-axis denotes their heights which are the normalized counts (probabilities) of scores in corresponding bins. The histograms of multiple, drug-centered, target-centered, and single motifs are shown with purple, red, green, and yellow in (a), (b), (c), and (d), respectively. All histograms of unapproved drug-target pairs are rendered with cyan in all subfigures. The color of overlapping parts of two histograms in each subfigure is just the sum of their individual colors.



(a) (b)

FIGURE 3: Drug-target pair space. Unapproved DTPs are marked by cyan crosses. Approved DTPs of drug-centered, target-centered, single, and multiple motifs are marked by red circles, green triangles, yellow diamonds, and purple squares, respectively. X, Y, and Z denote the first three principal components, respectively.

(refer to Sections 3.1 and 3.2) but are also able to adopt them as the confidence scores of being potential interactions when predicting potential interactions for unapproved DTPs.

According to the distribution in DTP space, the farther the pair is from the origin, the more possible it is to be a potential interaction. Thus, we only focused on the unapproved drug-target pairs remarkably far away from the origin. In order to validate them, we selected the top five out of them as the interaction candidates in terms of their distance to the origin for each dataset and checked them in popular drug/compound databases, ChEMBL (C), DrugBank (D), and KEGG (K). Since ChEMBL provides the predicted interactions (not approved yet), we only selected the most confident interactions with the score of 1 under the cut-off of 1 μM [5]. Comparing with ChEMBL, DrugBank, and KEGG, we showed our consistent predictions of the potential interactions of unapproved drug-target pairs for the adopted datasets in Table 4.

4. Conclusions

In this paper, we have addressed crucial issues in predicting drug-target interactions, which have not yet been solved well by former methods. These issues include the powerless inference in the case of isolated subnetworks, the biased classifiers derived from few positive samples, the need of training a number of classifiers, and the unavailable relationship between known DTIs and unapproved DTPs.

By characterizing each drug-target pair as a feature vector of within-scores and between-scores, our approach has the following advantages: (1) all types of drug-target pairs are treated in a same form, regardless of the available path between drugs and targets; (2) enough positive samples are able to reduce the bias of training model and only one classifier needs to be trained; (3) more importantly, the relationship between known DTIs and unapproved DTPs can be investigated in the same visualized space.

Table 3: Comparison with other three methods under LOOCV.

	BLM*	BLM-GIP*	Our*	NBI#	BLM-GIP#	BLM-NII#	Our#
EN	0.976/0.833	0.966/0.845	**0.985/0.849**	0.975	0.978/0.915	0.988/0.929	**0.999/0.998**
IC	0.973/0.781	0.971/0.807	**0.977/0.820**	0.976	0.984/0.943	0.990/**0.950**	**0.997**/0.897
GPCR	0.955/0.667	0.947/0.660	**0.975/0.772**	0.946	0.954/0.790	0.984/0.865	**0.998/0.971**
NR	0.881/0.612	0.864/0.547	**0.946/0.774**	0.838	0.922/0.684	0.981/0.866	**0.982/0.949**

*Using chemical similarity for drugs and sequence similarity for targets only.
#Combining topological similarities (MI) with chemical similarity and sequence similarity, respectively. NBI only provides AUC values and run tests under 5-fold cross validation (5CV) which is statistically same as LOOCV when the number of samples is enough.

Table 4: The top five predicted interactions of nuclear receptor.

Rank	En		IC		GPCR		NR	
	Validation	Pair	Validation	Pair	Validation	Pair	Validation	Pair
1	D	D05458 hsa:4128	D, K	D00438 hsa779	C	D03966 hsa2914	C, K	D00348 hsa5915
2	D	D00947 hsa:4129	—	D00619 hsa3749	—	D03966 hsa2917	C, K	D00348 hsa5916
3	—	D00039 hsa:587	—	D00816 hsa3781	—	D01346 hsa2916	—	D01132 hsa6097
4	—	D00437 hsa:1585	D	D00619 hsa776	K	D00442 hsa6755	C	D00348 hsa6256
5	—	D03365 hsa:1548	—	D00619 hsa3736	—	D00049 hsa8843	C	D00348 hsa6257

C, D, and K label the validated interactions in ChEMBL, DrugBank, and KEGG, respectively.

In addition, to capture similarity better, we have introduced a complete metric of topological similarity of drugs/targets by considering both the targets/drugs shared by two drugs/targets and the targets/drugs interacting with none of them. We also have proposed an adaptive combination rule, instead of the former linear combination between topological similarity and chemical/sequence similarity, by considering that the drug/target nodes' degrees follow the power-law distribution.

Finally, the effectiveness of our approach is demonstrated by comparing with existing popular methods under the cross validation and predicting potential interactions for DTPs under the validation in existing databases.

Conflict of Interests

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

Acknowledgments

This work was supported by Hong Kong Scholars Program (no. XJ2011028) and China Postdoctoral Science Foundation (no. 2012M521803) and was partially supported by NWPU Foundation for Fundamental Research (no. JCY20130137).

References

[1] M. R. Hurle, L. Yang, Q. Xie, D. K. Rajpal, P. Sanseau, and P. Agarwal, "Computational drug repositioning: from data to therapeutics," *Clinical Pharmacology & Therapeutics*, vol. 93, no. 4, pp. 335–341, 2013.

[2] M. Kanehisa, M. Araki, S. Goto et al., "KEGG for linking genomes to life and the environment," *Nucleic Acids Research*, vol. 36, supplement 1, pp. D480–D484, 2008.

[3] E. E. Bolton, Y. Wang, P. A. Thiessen, and S. H. Bryant, "PubChem: integrated platform of small molecules and biological activities," *Annual Reports in Computational Chemistry*, vol. 4, pp. 217–241, 2008.

[4] V. Law, C. Knox, Y. Djoumbou et al., "DrugBank 4.0: shedding new light on drug metabolism," *Nucleic Acids Research*, vol. 42, no. 1, pp. D1091–D1097, 2014.

[5] A. Gaulton, L. J. Bellis, A. P. Bento et al., "ChEMBL: a large-scale bioactivity database for drug discovery," *Nucleic Acids Research*, vol. 40, no. D1, pp. D1100–D1107, 2012.

[6] M. A. Yildirim, K.-I. Goh, M. E. Cusick, A.-L. Barabási, and M. Vidal, "Drug—target network," *Nature Biotechnology*, vol. 25, no. 10, pp. 1119–1126, 2007.

[7] F. Cheng, C. Liu, J. Jiang et al., "Prediction of drug-target interactions and drug repositioning via network-based inference," *PLoS Computational Biology*, vol. 8, no. 5, Article ID e1002503, 2012.

[8] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug-target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[9] T. van Laarhoven, S. B. Nabuurs, and E. Marchiori, "Gaussian interaction profile kernels for predicting drug-target interaction," *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.

[10] J.-P. Mei, C.-K. Kwoh, P. Yang, X.-L. Li, and J. Zheng, "Drug-target interaction prediction by learning from local information and neighbors," *Bioinformatics*, vol. 29, no. 2, pp. 238–245, 2013.

[11] T. Klabunde, "Chemogenomic approaches to drug discovery: similar receptors bind similar ligands," *British Journal of Pharmacology*, vol. 152, no. 1, pp. 5–7, 2007.

[12] F. J. Azuaje, L. Zhang, Y. Devaux, and D. R. Wagner, "Drug-target network in myocardial infarction reveals multiple side effects of unrelated drugs," *Scientific Reports*, vol. 1, article 52, 2011.

[13] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, "Prediction of drug-target interaction networks from the integration of chemical and genomic spaces," *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, 2008.

[14] M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa, "Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways," *Journal of the American Chemical Society*, vol. 125, no. 39, pp. 11853–11865, 2003.

[15] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[16] J. I. F. Bass, A. Diallo, J. Nelson, J. M. Soto, C. L. Myers, and A. J. M. Walhout, "Using networks to measure similarity between genes: association index selection," *Nature Methods*, vol. 10, no. 12, pp. 1169–1176, 2013.

[17] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, ACM, 2006.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 862130, 9 pages
http://dx.doi.org/10.1155/2015/862130

Research Article

RNAseq by Total RNA Library Identifies Additional RNAs Compared to Poly(A) RNA Library

Yan Guo,[1] Shilin Zhao,[1] Quanhu Sheng,[1] Mingsheng Guo,[1] Brian Lehmann,[2] Jennifer Pietenpol,[2] David C. Samuels,[3] and Yu Shyr[1]

[1]*Center for Quantitative Sciences, Vanderbilt University, Nashville, TN 37232, USA*
[2]*Department of Biochemistry, Vanderbilt University, Nashville, TN 37232, USA*
[3]*Center for Human Genetics Research, Vanderbilt University, Nashville, TN 37232, USA*

Correspondence should be addressed to Yan Guo; yan.guo@vanderbilt.edu and Yu Shyr; yu.shyr@vanderbilt.edu

Received 8 December 2014; Revised 27 January 2015; Accepted 15 February 2015

Academic Editor: Xia Li

The most popular RNA library used for RNA sequencing is the poly(A) captured RNA library. This library captures RNA based on the presence of poly(A) tails at the 3′ end. Another type of RNA library for RNA sequencing is the total RNA library which differs from the poly(A) library by capture method and price. The total RNA library costs more and its capture of RNA is not dependent on the presence of poly(A) tails. In practice, only ribosomal RNAs and small RNAs are washed out in the total RNA library preparation. To evaluate the ability of detecting RNA for both RNA libraries we designed a study using RNA sequencing data of the same two breast cancer cell lines from both RNA libraries. We found that the RNA expression values captured by both RNA libraries were highly correlated. However, the number of RNAs captured was significantly higher for the total RNA library. Furthermore, we identify several subsets of protein coding RNAs that were not captured efficiently by the poly(A) library. One of the most noticeable is the histone-encode genes, which lack the poly(A) tail.

1. Introduction

With the advancement of high throughput sequencing technology, advanced data mining techniques have been developed for high throughput DNA sequencing data [1, 2]. Similar data mining techniques can be applied to RNAseq data. RNAseq technology can be categorized into three subclasses by the types of RNA sequenced: messenger RNA (mRNA or protein coding RNA), micro RNA (miRNA), and total RNA. The sequencing method is the same but each differs in the RNA species present for cDNA synthesis and subsequent library construction. The cDNA library for mRNAseq is made only from the poly(A) mRNA. Small RNAs are not captured during oligo-dT based mRNA enrichment. To date, the most popular application of RNAseq technology is mRNA sequencing because most researchers use RNAseq as a replacement for microarray to perform high throughput gene expression profiling [3–6] and coding regions remain the focus of human disease research.

Long noncoding RNA (lncRNA), on the other hand, was traditionally believed to be nonfunctional. However, many recent studies have shown evidence for the functionality of lncRNA [7, 8], such as roles in high-order chromosomal dynamics [9], embryonic stem cell differentiation [10], telomere biology [11], subcellular structural organization [12], and breast cancer [13, 14]. The interest in lncRNA grew considerably as the evidence of lncRNA's role in various biological contexts accumulated in the recent years. LncRNAs are usually defined as noncoding RNA with length more than 200 base pairs [7, 15]. Structurally, lncRNAs and mRNAs are very similar, as both can exhibit polyadenylation (poly(A)). The number of definable lncRNAs varies by study. An early study in 2007 estimated that there are 4 times more lncRNAs than protein coding RNA [16]. Another study claims to have identified 35,000 lncRNAs [17], and many of them have characteristics similar to mRNA such as 5′ capping, splicing, and polyadenylation, with the exception of open reading frames [17]. In the latest effort to quantify human

lncRNA, the Encyclopedia of DNA Elements (ENCODE) [18] project identified 13,333 lncRNAs and further categorized them into four subclasses: (1) antisense, (2) large intergenic noncoding RNAs (lincRNA), (3) sense intronic, and (4) processed transcripts.

While it is possible to study lncRNAs using traditional microarrays, RNAseq has been proven to be the superior technology for this purpose due to its greater sensitivity and the ability to detect novel lncRNAs [19, 20]. The rise in the popularity and affordability of RNAseq technology is primarily responsible for the growing interest in and understanding of lncRNAs as researchers explore the presence of these stowaways in their mRNA data sets. In mRNA sequencing, mRNAs are captured based on the presence of a poly(A) tail. LncRNAs can also be captured provided they have a poly(A) tail. According to a study in 2005, it is estimated that 40% of lncRNA transcripts are nonpolyadenylated [21]. An alternative library preparation method for studying lncRNA is the total RNA library. Only ribosomal RNA is removed leaving small RNAs, mRNAs, and all forms of lncRNAs. This library preparation method is the most inclusive of RNA species but requires more sequencing reads due to the multiple RNA species present in the library, and ribosomal RNA reduction does not completely remove ribosomal RNA from the library due to their high abundance.

Total RNA sequencing theoretically should detect more lncRNAs due to its RNA selection independent of the poly(A) tail. However, total RNAseq costs more than mRNA sequencing (mRNA $500 versus total RNA $650) and the question of how many more lncRNAs does total RNA sequencing capture compared to mRNA sequencing has not been answered. Moreover, whether the mRNAs captured in total RNA sequencing are comparable to mRNA sequencing also remains unknown. To answer these questions, we designed the following study. We hypothesized that total RNA sequencing generates more relevant data than mRNA sequencing for the purpose of lncRNA research. Total RNA and mRNA libraries of two breast cancer cell line samples were built and sequenced. We analyzed the sequencing data and compared their usability for lncRNA and mRNA research.

2. Methods

Total RNAseq on two breast cancer cell lines HS578T and BT549 was performed by the Vanderbilt Technologies for Advanced Genomics (VANTAGE) core. Total RNA was isolated with the Aurum Total RNA Mini Kit. All samples were quantified on the QuBit RNA assay. RNA quality was checked using Agilent Bioanalyzer. RNA integrity number (RIN) for both samples was 10. RNAseq data was obtained by first using the Ribo-Zero Magnetic Gold Kit (human/mouse/rat) (Epicentre) to perform ribosomal reduction on 1 μg total RNA following the manufacturer's protocol. After ribosomal RNA (rRNA) depletion, samples were then purified using the Agencourt RNAClean XP Kit (Beckman Coulter) according to the Epicentre protocol specifications. After purification, samples were eluted in 11 μL RNase-free water. Next, 1 μL ribosomal depleted samples were run on the Agilent RNA 6000 Pico Chip to confirm rRNA removal. After confirmation of rRNA removal, 8.5 μL rRNA-depleted sample was input into the Illumina TruSeq Stranded RNA Sample Preparation kit (Illumina) for library preparation. The libraries were sequenced on Illumina High HiSeq 2500 with paired-end 100 base pair long reads. Raw RNAseq sequencing data generated from the poly(A) library of the same two cell lines were downloaded from the Gene Expression Omnibus (GEO) (GSM1172877: 19.8 million reads and GSM1172855: 15.3 million reads) for comparative purpose. The poly(A) libraries were prepared using Illumina TruSeq RNA Sample Preparation kit. Poly(A) RNA was purified with oligo dT magnetic beads, and the poly(A) RNA was fragmented with divalent cations followed by reverse transcription into cDNA and ligation of Illumina paired-end oligo adapters to the cDNA fragments. More detail of poly(A) library construction can be found at GEO website.

The raw data quality was examined using QC3 [22]. Alignment against human genome reference HG19 was performed using TopHat2 [23]. Novel gene quantification was performed using Cufflinks [24]. Additional quality control was carried out at alignment level based on the alignment quality control concept described in [25]. ENSEMBL gene transfer format (GTF) version GRCh37.35 was used to annotate the gene expression. We categorized the RNA into three subclasses: protein coding RNA, lncRNA, and other RNAs. This GTF contains 20327 protein coding RNAs, 13346 lncRNAs, and 24100 other RNAs (such as pseudogene and antisense). Read count per RNA was computed using HTSeq [26]. To avoid variation caused by total reads sequenced, raw read counts were normalized to the total read count by sample. Log2 transformations were performed on normalized read counts. To avoid log of zeroes, all read counts were increased by 1 before taking the log transformation. Differential expression analyses and additional quality control were conducted between poly(A) capture method and total RNA method using MultiRankSeq [27] which embeds three different RNAseq differential expression analysis methods: DESeq2 [28], edgeR [29], and baySeq [30]. DEseq2's results were selected for further analysis due to its ability to take paired samples into consideration. Cluster analysis was performed using Heatmap3 [31]. Functional analysis was carried out using gene set enrichment analysis (GSEA) [32], and gene ontology (GO) analysis was conducted using WebGestalt [33].

3. Results

Even though the RNAseq data were generated from the same cell lines, there could be potential heterogeneity and batch effect because the cell lines were cultured at two different labs and sequenced at two different facilities. To test if there is potential heterogeneity and batch effect, we conducted a cluster analysis using Heatmap3 [31]. Unsupervised cluster results showed cluster of cell line type rather than sequencing batch (Figure 1) which suggested that the RNAseq data of

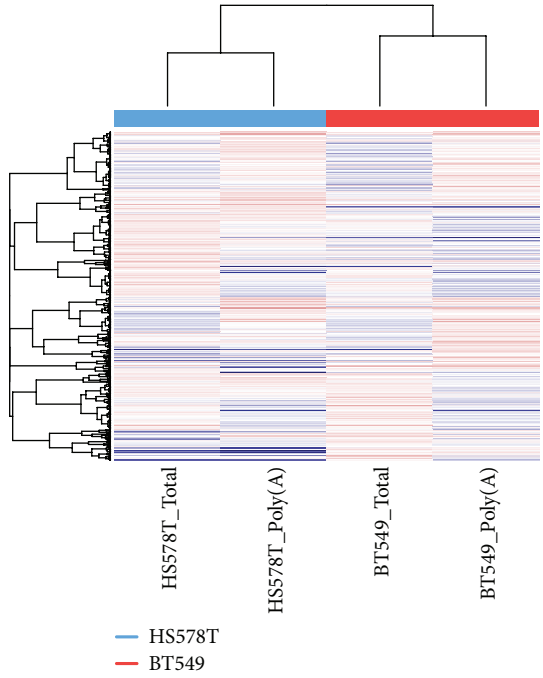


Figure 1: Cluster results of the two breast cancer cell lines. The poly(A) and total RNA libraries were constructed and sequenced by separated facilities. The samples clustered together by cell line type rather than library type or sequencing facility, which suggests that there is no severe heterogeneity of cell line and batch effect between sequencing.

these two cell lines were similar; no severe heterogeneity and batch effect were observed.

The sequencing data went through rigorous quality control. To account for variation in number of reads sequenced within the 4 samples, read counts were adjusted by normalizing the total read count of each sample. In terms of proportion of reads mapped to lncRNA, total RNA library samples (3.62% and 3.23%) had a higher proportion than poly(A) library samples (0.85% and 1.02%). For protein coding RNA, poly(A) library samples (96.34% and 95.38%) mapped a higher proportion of reads than total RNA samples (92.47% and 93.45%). For other species of RNAs, poly(A) library samples (2.81% and 3.59%) and total RNA library samples (3.91% and 3.32%) had similar proportion of reads aligned.

The distributions of read normalized read counts for protein coding RNA, lncRNA, and other RNAs can be seen in Figure S1 (in Supplementary Material available online at http://dx.doi.org/10.1155/2015/862130). All three types of RNAs were detected by both poly(A) and total RNA library building methods. To compare whether RNA expressions are comparable between the two RNA library building methods, we drew a scatter plot and computed their Pearson's correlation coefficients (Figure 2). All three types of RNA expression are highly agreeable between the two methods (protein coding RNA $r = 0.92$, lncRNA Pearson $r = 0.79$, and other RNAs $r = 0.69$). These results are consistent with previous findings [34] which suggest that RNA expression is consistently measured for poly(A) and total RNA sequencing library.

Next, we examine the number of RNAs detectable by each library construction method. To determine whether RNA is detected, a cutoff value of the normalized read count was applied. Because this cutoff is arbitrary, we choose several different thresholds for sensitivity analysis. An RNA is considered detected if its normalized read count is above the detection threshold. We used the following thresholds: >0.1, >0.5, >1, >1.5, and >2. Regardless of which threshold we applied, samples from the total RNA method consistently showed higher numbers of RNAs detected for all three types of RNAs (Figure 3). This suggests that without the restriction of poly(A) selection, the total RNA library is capable of identifying more expressed RNAs (lncRNA t-test $P < 0.0001$, protein coding RNA t-test $P < 0.0001$, and other RNAs t-test $P < 0.0001$). Furthermore, we compared the number of genes that are differentially expressed between the two libraries' construction methods and found there were much higher expressed RNAs (log2 fold change > 2) for total RNA library samples than poly(A) library samples (Figure 4). We also counted the potential novel transcripts identified from Cufflinks. The two poly(A) library samples detected 4122 and 6169 potential new transcripts, and the two total RNA samples detected 53282 and 58111 potential new transcripts, roughly a 10-fold increase.

It has been shown that not all mRNAs necessarily contain a poly(A) tail at their $3'$ ends [35]. For example, the mRNA that encodes histone proteins is nonpolyadenylated [36]. Another study has shown that a significant portion of the mRNA transcript has no poly(A) tail [37]. This can

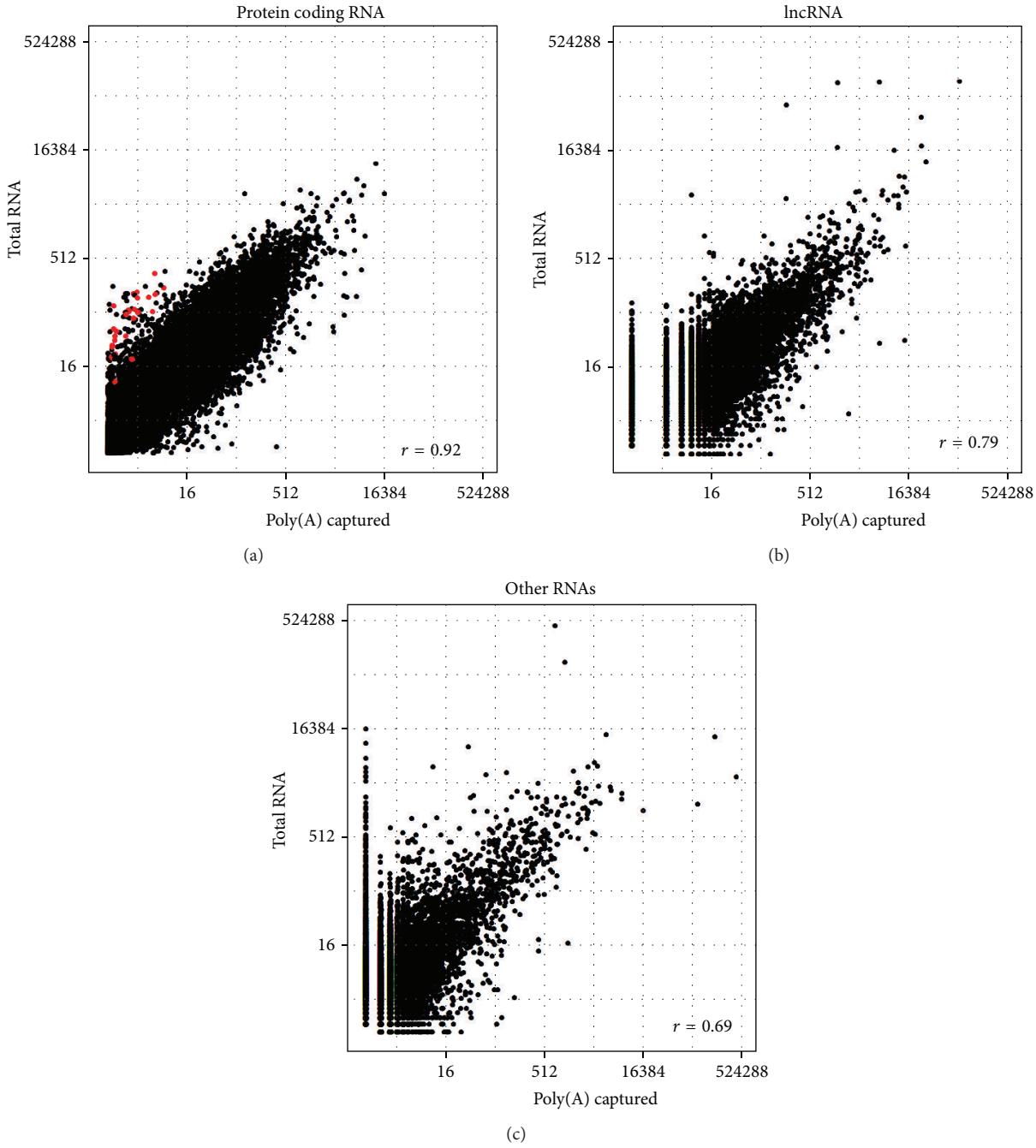


Figure 2: RNA expression level consistency between poly(A) and total RNA library samples. Read counts were normalized by total read count per sample and log2 transformed. (a) Consistency of expression of protein coding RNAs. The red color indicates histone-encoding genes. (b) Consistency of expression of lncRNAs. (c) Consistency of expression of other RNAs.

potentially explain why we observe more protein coding RNA detected by total RNA than the poly(A) method. To test this hypothesis, we searched through the ENSEMBL database and found 38 histone-encoding genes. We conducted enrichment analysis in GSEA using results from DESeq2 against the histone-encoding genes and found that our dataset was highly enriched (FDR < 0.0001) (Figure 5(a)). The expression value of the histone-encoding genes was clearly higher for total RNA library samples (Figure 5(b)). The GSEA showed that total RNA library samples captured histone-encoding genes at a much higher efficiency than the poly(A) library samples. Based on fold change results from DESeq2, there were 737 protein coding RNAs that have a log2 fold change greater than 2 (overexpressed in total RNA samples), which suggests that additional subsets of protein coding RNAs may be better captured using total RNA methods. To better categorize these potential subcategories of protein coding RNAs, we conducted GO analysis using WebGestalt (Figure S2) (Table 1).

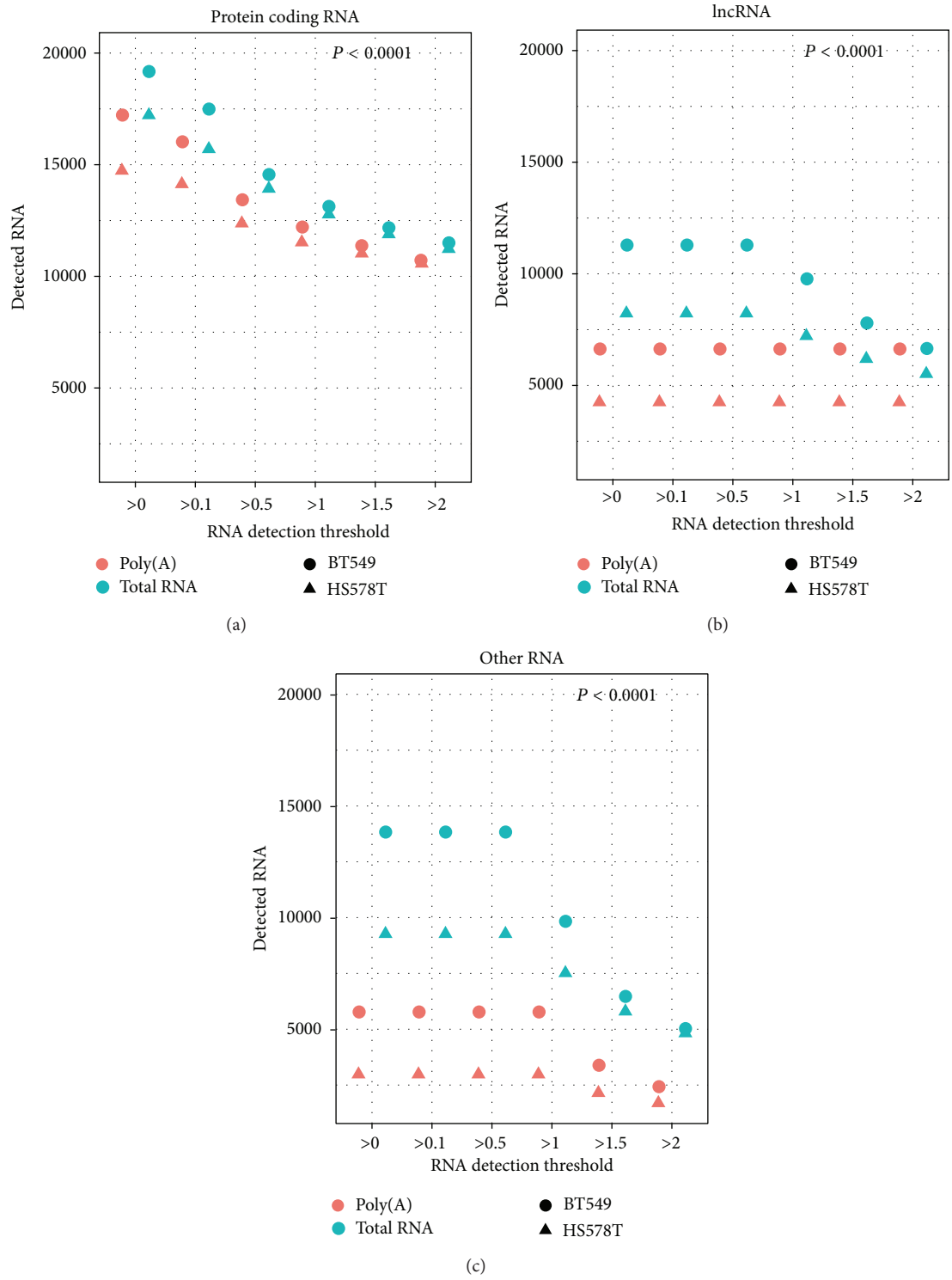


Figure 3: Number of RNAs detected at different detection thresholds for all three types of RNA. Total RNA library samples detected significantly more RNAs than poly(A) RNA library samples at all RNA detection thresholds. (a) Protein coding RNA. (b) lncRNA. (c) Other RNAs.

The top 10 subcategories of genes were found within all three big GO categories: biological process, molecular function, and cellular component. Eleven out of the 30 subcategories primarily consisted of histone-encoding genes. The other 19 subcategories were protein-DNA complex, chromatin, and so forth. No obvious pattern was recognizable. There were also 592 protein coding genes that were captured better by the poly(A) library samples (log2 fold change < -2). We also

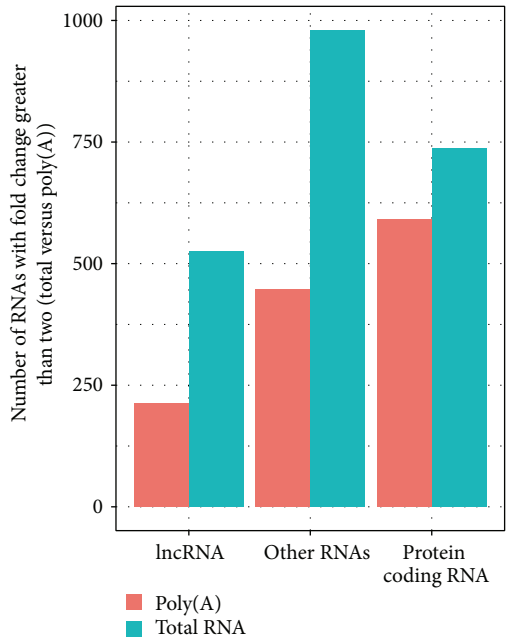


Figure 4: Using log2 fold change >2 as cutoffs, total RNA library samples had more RNAs with higher expression levels than poly(A) samples for all three types of RNAs.

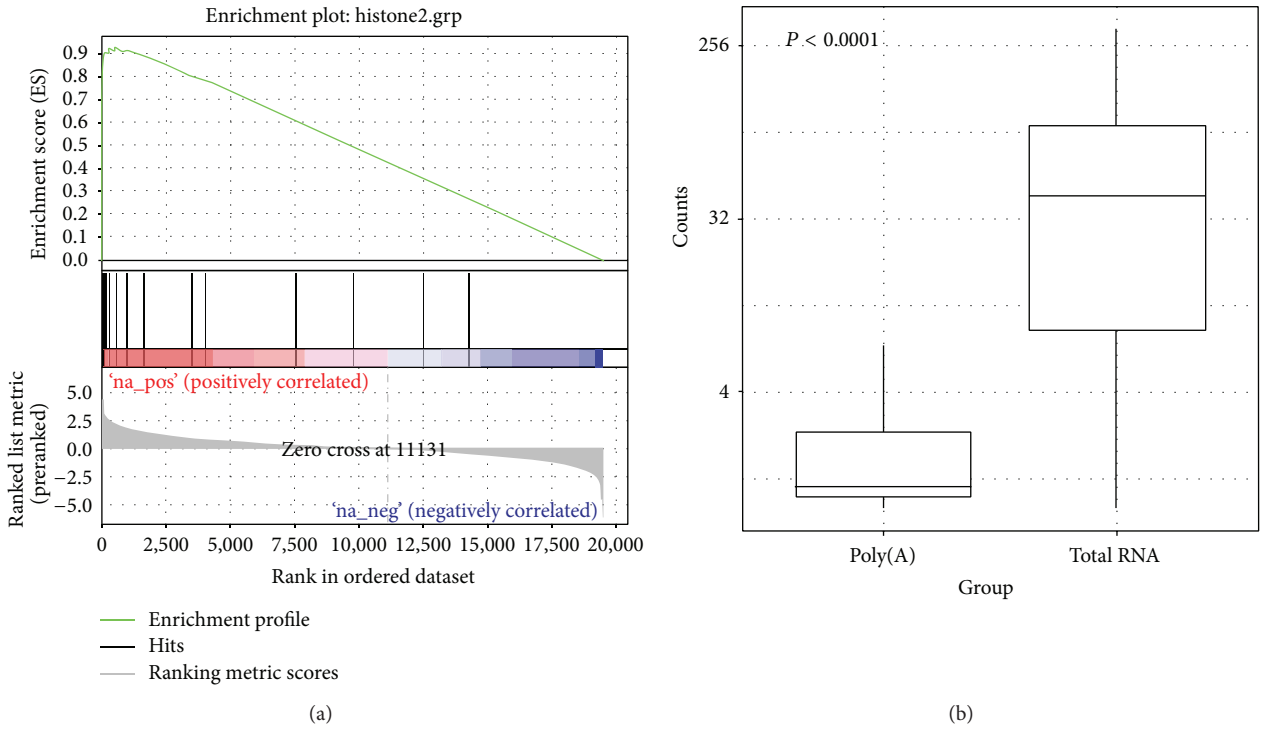


Figure 5: (a) Enrichment plot of histone-encoding genes from GSEA. Based on fold change ranked (total RNA versus poly(A)) gene list, histone-encoding genes were highly enriched (adjust $P < 0.0001$). (b) Normalized read count distribution of the 38 histone-encoding genes between poly(A) and total RNA libraries.

Table 1: Gene ontology results of genes that are captured more by total RNA library.

Major category	Subcategory	Number of genes	Adjusted P
Biological process	Nucleosome assembly (histone)	30	$4.81E-16$
	Protein-DNA complex assembly (histone)	33	$6.27E-16$
	Chromatin assembly (histone)	30	$2.58E-15$
	Protein-DNA complex subunit organization (histone)	33	$9.36E-15$
	Nucleosome organization (histone)	30	$2.36E-14$
	Chromatin assembly or disassembly (histone)	30	$1.81E-13$
	DNA packaging (histone)	30	$2.51E-12$
	DNA conformation change (histone)	30	$7.46E-10$
	Cellular macromolecular complex assembly (histone)	38	$6.50E-03$
	Detection of virus	3	$7.30E-03$
Molecular function	Protein heterodimerization activity (histone)	32	$5.00E-04$
	Ketosteroid monooxygenase activity	3	$4.00E-03$
	Phenanthrene 9,10-monooxygenase activity	3	$4.00E-03$
	cGMP binding	5	$4.00E-03$
	Oxidoreductase activity	7	$6.30E-03$
	Androsterone dehydrogenase activity	3	$6.30E-03$
	Dehydrogenase activity	3	$6.30E-03$
	Cyclic nucleotide binding	6	$1.48E-02$
	N,N-Dimethylaniline monooxygenase activity	3	$1.48E-02$
	Metal ion transmembrane transporter activity	25	$2.83E-02$
Cellular component	Nucleosome (histone)	29	$8.22E-23$
	Protein-DNA complex	30	$4.93E-17$
	Chromatin	35	$1.22E-08$
	Chromosomal part	40	$1.00E-04$
	Chromosome	41	$2.60E-03$
	Extracellular region part	59	$1.33E-02$
	Axoneme	9	$2.28E-02$
	Platelet dense tubular network membrane	3	$6.32E-02$
	Platelet dense tubular network	3	$1.15E-01$
	Desmosome	4	$1.57E-01$

performed GO analysis on these genes (Figure S3) (Table 2). No clear gene pattern was detected.

4. Discussion

In this study, we examined the difference between the RNAs captured through poly(A) and total RNA libraries. Our study was also designed with several limitations. First, we were only able to collect two samples with sequencing data from both RNA libraries. The small sample size might limit our ability to identify true signals. Also, the sample type is limited to breast cancer cell lines. Other tissue types might behave differently.

Using sequencing data from two breast cancer cell lines captured using both libraries, we found that, in terms of expression level, both libraries were highly correlated and the correlation was the highest for protein coding RNAs. This suggests that both methods of RNA library construction are capable of generating consistent data for studying protein coding RNAs. For the three types of RNA we defined: protein coding RNA, lncRNA, and other RNAs; at all gene detection thresholds, total RNA library samples consistently identified more RNAs than poly(A) library samples which suggests that the total RNA library is capable of detecting additional RNA not detected by the poly(A) library. Through gene set enrichment analysis we were able to identify that histone-encoding genes were not captured efficiently by the poly(A) RNA library due to their lack of poly(A) tails. This finding is consistent with previous reports [36, 37]. Through gene ontology analysis we identified several additional subgroups of RNA which were better captured by the total RNA library. This could be explained in several ways. First, the results could be due to random variation, thus not holding any biological significance. Second, the poly(A) tails might have degraded prior to the construction of the poly(A) RNA library. Third, some unknown mechanisms may prevent proper capture of such RNAs through poly(A) identification.

Total RNA library construction costs around $150 more than a poly(A) library, but it allows the detection of additional RNAs. Whether the extra cost is justifiable should be decided during the experimental design stage of RNAseq study. If the goal is to study lncRNA, then it is better to use total RNA library; if the goal is to study protein coding RNAs, then total

Table 2: Gene ontology results of genes that are captured more by poly(A) RNA library.

Major category	Subcategory	Number of genes	Adjusted P
Biological process	RNA metabolic process	174	$1.30E-03$
	Nucleic acid metabolic process	188	$4.50E-03$
	Cellular macromolecule metabolic process	260	$5.30E-03$
	Positive regulation of cell development	17	$5.50E-03$
	Transcription from RNA polymerase II promoter	75	$5.50E-03$
	Cellular component organization	174	$5.80E-03$
	Cellular component organization or biogenesis	177	$6.90E-03$
	Positive regulation of cell morphogenesis	7	$6.90E-03$
	Negative regulation of viral entry into host cell	3	$7.00E-03$
	Regulation of transcription, DNA-dependent	126	$7.00E-03$
Molecular function	Chromatin binding	27	$1.00E-03$
	Protein binding	276	$1.60E-03$
	Binding	400	$3.90E-02$
	D-Erythro-sphingosine kinase activity	2	$5.57E-02$
	Transcription cofactor activity	28	$5.57E-02$
	Lipid kinase activity	3	$5.57E-02$
	Sphinganine kinase activity	2	$5.57E-02$
	Transcription factor binding transcription factor activity	28	$6.34E-02$
	Nucleic acid binding	127	$9.22E-02$
	Protein binding transcription factor activity	28	$9.22E-02$
Cellular component	Nucleus	251	$4.99E-09$
	Membrane-bounded organelle	349	$1.44E-06$
	Intracellular membrane-bounded organelle	349	$1.44E-06$
	Intracellular organelle	375	$5.83E-06$
	Organelle	375	$5.83E-06$
	Nuclear lumen	128	$6.62E-06$
	Nuclear part	139	$8.80E-06$
	Intracellular organelle lumen	144	$3.83E-05$
	Organelle lumen	145	$4.64E-05$
	Nucleoplasm	77	$5.05E-05$

RNA library might not be necessary unless histone-encoding genes are of interest.

Conflict of Interests

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

Acknowledgments

Yan Guo and Yu Shyr were supported by P30 CA68485. Jennifer Pietenpol was supported by National Institute of Health, Grants CA95131 and RC2CA148375. Brian Lehmann was supported by Komen for the Cure Foundation, Grant KG262005. The sequencing of the two total RNA library samples was supported by Vanderbilt Institute for Clinical and Translational Research, Grant VR8688. The authors would also like to thank Margot Bjoring for editorial support.

References

[1] D. C. Samuels, L. Han, J. Li et al., "Finding the lost treasures in exome sequencing data," *Trends in Genetics*, vol. 29, no. 10, pp. 593–599, 2013.

[2] F. Ye, D. C. Samuels, T. Clark, and Y. Guo, "High-throughput sequencing in mitochondrial DNA research," *Mitochondrion*, vol. 17, pp. 157–163, 2014.

[3] Y. Guo, C.-I. Li, F. Ye, and Y. Shyr, "Evaluation of read count based RNAseq analysis methods," *BMC Genomics*, vol. 14, supplement 8, article S2, 2013.

[4] Y. Guo, Q. Sheng, J. Li, F. Ye, D. C. Samuels, and Y. Shyr, "Large scale comparison of gene expression levels by microarrays and RNAseq using TCGA data," *PLoS ONE*, vol. 8, no. 8, Article ID e71462, 2013.

[5] J. Shendure, "The beginning of the end for microarrays?" *Nature Methods*, vol. 5, no. 7, pp. 585–587, 2008.

[6] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[7] T. R. Mercer, M. E. Dinger, and J. S. Mattick, "Long non-coding RNAs: insights into functions," *Nature Reviews Genetics*, vol. 10, no. 3, pp. 155–159, 2009.

[8] M. E. Dinger, P. P. Amaral, T. R. Mercer, and J. S. Mattick, "Pervasive transcription of the eukaryotic genome: functional indices and conceptual implications," *Briefings in Functional Genomics and Proteomics*, vol. 8, no. 6, pp. 407–423, 2009.

[9] P. P. Amaral and J. S. Mattick, "Noncoding RNA in development," *Mammalian Genome*, vol. 19, no. 7-8, pp. 454–492, 2008.

[10] M. E. Dinger, P. P. Amara, T. R. Mercer et al., "Long noncoding RNAs in mouse embryonic stem cell pluripotency and differentiation," *Genome Research*, vol. 18, no. 9, pp. 1433–1445, 2008.

[11] S. Schoeftner and M. A. Blasco, "Developmentally regulated transcription of mammalian telomeres by DNA-dependent RNA polymerase II," *Nature Cell Biology*, vol. 10, no. 2, pp. 228–236, 2008.

[12] T. R. Mercer, M. E. Dinger, S. M. Sunkin, M. F. Mehler, and J. S. Mattick, "Specific expression of long noncoding RNAs in the mouse brain," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 2, pp. 716–721, 2008.

[13] A. Bhan, I. Hussain, K. I. Ansari, S. A. M. Bobzean, L. I. Perrotti, and S. S. Mandal, "Bisphenol-A and diethylstilbestrol exposure induces the expression of breast cancer associated long noncoding RNA HOTAIR *in vitro* and *in vivo*," *The Journal of Steroid Biochemistry and Molecular Biology*, vol. 141, pp. 160–170, 2014.

[14] A. Bhan, I. Hussain, K. I. Ansari, S. Kasiri, A. Bashyal, and S. S. Mandal, "Antisense transcript long noncoding RNA (lncRNA) HOTAIR is transcriptionally induced by estradiol," *Journal of Molecular Biology*, vol. 425, no. 19, pp. 3707–3722, 2013.

[15] J. M. Perkel, "Visiting 'noncodarnia'," *BioTechniques*, vol. 54, no. 6, pp. 301–304, 2013.

[16] P. Kapranov, J. Cheng, S. Dike et al., "RNA maps reveal new RNA classes and a possible function for pervasive transcription," *Science*, vol. 316, no. 5830, pp. 1484–1488, 2007.

[17] P. Carninci, T. Kasukawa, S. Katayama et al., "The transcriptional landscape of the mammalian genome," *Science*, vol. 309, no. 5740, pp. 1559–1563, 2005.

[18] S. Djebali, C. A. Davis, A. Merkel et al., "Landscape of transcription in human cells," *Nature*, vol. 489, no. 7414, pp. 101–108, 2012.

[19] L. Han, K. C. Vickers, D. C. Samuels, and Y. Guo, "Alternative applications for distinct RNA sequencing strategies," *Briefings in Bioinformatics*, 2014.

[20] K. C. Vickers, L. A. Roteta, H. Hucheson-Dilks, L. Han, and Y. Guo, "Mining diverse small RNA species in the deep transcriptome," *Trends in Biochemical Sciences*, vol. 40, no. 1, pp. 4–7, 2015.

[21] J. Cheng, P. Kapranov, J. Drenkow et al., "Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution," *Science*, vol. 308, no. 5725, pp. 1149–1154, 2005.

[22] Y. Guo, S. Zhao, Q. Sheng et al., "Multi-perspective quality control of Illumina exome sequencing data using QC3," *Genomics*, vol. 103, no. 5-6, pp. 323–328, 2014.

[23] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg, "TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions," *Genome Biology*, vol. 14, no. 4, article R36, 2013.

[24] C. Trapnell, B. A. Williams, G. Pertea et al., "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, vol. 28, no. 5, pp. 511–515, 2010.

[25] Y. Guo, F. Ye, Q. Sheng, T. Clark, and D. C. Samuels, "Three-stage quality control strategies for DNA re-sequencing data," *Briefings in Bioinformatics*, vol. 15, no. 6, pp. 879–889, 2014.

[26] S. Anders, P. T. Pyl, and W. Huber, "HTSeq—a Python framework to work with high-throughput sequencing data," *Bioinformatics*, vol. 31, no. 2, pp. 166–169, 2015.

[27] Y. Guo, S. Zhao, F. Ye, Q. Sheng, and Y. Shyr, "MultiRankSeq: multiperspective approach for RNAseq differential expression analysis and quality control," *BioMed Research International*, vol. 2014, Article ID 248090, 8 pages, 2014.

[28] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, vol. 15, no. 12, article 550, 2014.

[29] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2010.

[30] T. J. Hardcastle and K. A. Kelly, "baySeq: empirical Bayesian methods for identifying differential expression in sequence count data," *BMC Bioinformatics*, vol. 11, article 422, 2010.

[31] S. Zhao, Y. Guo, Q. Sheng, and Y. Shyr, "Advanced heat map and clustering analysis using heatmap3," *BioMed Research International*, vol. 2014, Article ID 986048, 6 pages, 2014.

[32] 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.

[33] J. Wang, D. Duncan, Z. Shi, and B. Zhang, "WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013," *Nucleic Acids Research*, vol. 41, pp. W77–W83, 2013.

[34] W. Zhao, X. He, K. A. Hoadley, J. S. Parker, D. N. Hayes, and C. M. Perou, "Comparison of RNA-Seq by poly (A) capture, ribosomal RNA depletion, and DNA microarray for expression profiling," *Acta Veterinaria Scandinavica*, vol. 15, article 419, 2014.

[35] W. F. Marzluff, E. J. Wagner, and R. J. Duronio, "Metabolism and regulation of canonical histone mRNAs: life without a poly(A) tail," *Nature Reviews Genetics*, vol. 9, no. 11, pp. 843–854, 2008.

[36] W. F. Marzluff, "Metazoan replication-dependent histone mRNAs: a distinct set of RNA polymerase II transcripts," *Current Opinion in Cell Biology*, vol. 17, no. 3, pp. 274–280, 2005.

[37] L. Yang, M. O. Duff, B. R. Graveley, G. G. Carmichael, and L.-L. Chen, "Genomewide characterization of non-polyadenylated RNAs," *Genome Biology*, vol. 12, no. 2, article R16, 2011.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 781023, 12 pages
http://dx.doi.org/10.1155/2015/781023

Research Article

Construction of Pancreatic Cancer Classifier Based on SVM Optimized by Improved FOA

Huiyan Jiang,[1] Di Zhao,[1] Ruiping Zheng,[1] and Xiaoqi Ma[2]

[1]*Software College, Northeastern University, Shenyang 110819, China*
[2]*School of Science and Technology, Nottingham Trent University, Nottingham NG17 8NU, UK*

Correspondence should be addressed to Huiyan Jiang; hyjiang@mail.neu.edu.cn

Received 17 April 2015; Revised 1 August 2015; Accepted 13 August 2015

Academic Editor: Jianhua Ruan

A novel method is proposed to establish the pancreatic cancer classifier. Firstly, the concept of quantum and fruit fly optimal algorithm (FOA) are introduced, respectively. Then FOA is improved by quantum coding and quantum operation, and a new smell concentration determination function is defined. Finally, the improved FOA is used to optimize the parameters of support vector machine (SVM) and the classifier is established by optimized SVM. In order to verify the effectiveness of the proposed method, SVM and other classification methods have been chosen as the comparing methods. The experimental results show that the proposed method can improve the classifier performance and cost less time.

1. Introduction

Pancreatic cancer is one of the world's top 10 malignant tumors [1]. Its early and accurate diagnosis is difficult. Once the diagnosis is confirmed, the tumor has reached an advanced stage. It is of great significance to improve prognosis for early detection, early diagnosis, and early treatment [2]. With the development of computer science and computer image-processing technology, computer aided detection (CAD) technology is established. CAD systems are increasingly used as an aid by radiologists for detection and interpretation of diseases [3], reducing the burden of doctors and improving the diagnosis accuracy.

Image recognition is one of the most important parts of CAD technology. The recognition process is mainly divided into two phases, namely, feature extraction and selection and classifier construction. In [4], we argue that tensors can describe space information among image features and need less space than vectors. Multilinear principal component analysis (MPCA) method [5] can be used to select the core tensors. In this paper, we also use tensors to represent CT images and MPCA to select core tensors to reduce the tensor dimension.

There are many methods to establish the medical image classifier. Kovalerchuk et al. [6] and Pendharkar et al. [7] used machine learning and data mining technology in breast cancer detection. In recent years, many researchers have made thorough research on medical image classification. Antonie et al. [8] combined association rule and neural network to mine the texture feature in different regions of breast images and realized the automatic diagnosis of breast cancer. Zhang et al. [9] classified cervix uterus lymphonodus by support vector machine (SVM) and size and shape features. Ramírez et al. [10] proposed to use neural network method in classification of brain images of Alzheimer's disease.

However, the research of pancreatic cancer classification is in a fledging period. Tsai and Kojma [11] proposed the pancreatic tiny anomaly detection method for CT images and introduced the square of logarithm operation in grayscale to enhance the margin of low grayscale. Takada et al. [12] proposed a new pancreatic classification system to distinguish the four parts of pancreas based on the anatomy of pancreas and their own experience. He et al. [13] proposed a novel group search optimizer- (GSO-) based biomarker discovery method for pancreatic cancer diagnosis using mass spectrometry data, compared with a genetic algorithm, evolution strategies, evolutionary programming, and a particle swarm optimizer and achieved better classification performance than other algorithms.

Theoretically, imaging examination of any body tissues and organs can use CAD technology to improve diagnostic accuracy. However, since the position of pancreas is covert and has complex relationship with other organs, the pancreatic cancer image classification is difficult.

In this paper, we employ SVM [14], which are suitable for solving small-sample learning and nonlinear and high dimension problems, to establish the pancreatic cancer classification, and improve fruit fly optimization algorithm (FOA) [15] to optimize parameters of SVM. We provide a new fitness function which is more in line with the actual clinical needs in K-fold cross-validation to assess the classifier performance. Using the above strategies, the classification performance can be improved. Experimental results on pancreatic regions of abdominal CT images demonstrate the feasibility and efficiency of the proposed method.

This paper is organized as follows. Section 2 introduces the background of this researching, including support vector machine, fruit fly optimization algorithm, and the concept of quantum. Section 3 illustrates the method of construction of SVM classifier based on improved FOA. Section 4 presents the experimental data and the evaluation criterions, showing the results of the pancreatic cancer classification based on the improved FOA and other comparative methods. It also discusses the experiment results. Section 5 concludes the work in this paper.

2. Background

We introduce SVM and FOA in this part; the concept of quantum is shown in [4].

2.1. Support Vector Machine. Support vector machine (SVM) [14] is built on statistical learning theory. It is suitable for small-sample learning and nonlinear and high dimension problem. SVM is based on the principle of structural risk minimization and has strong generalization ability. It studies optimal separating hyperplane in the high dimension feature space for sample classification.

SVM mainly aims at binary classification. For linear separable problem, we consider samples as (x_i, y_i). $x_i \in R^n$ is the feature set of medical images, $y_i \in \{+1, -1\}$ is the label of samples, and $i = 1, \ldots, l$, l is the number of samples. The optimal separating hyperplane is $f(x) = \omega^T x + b = 0$. The functional margin which is the distance from a sample point to separating hyperplane is $\widehat{\gamma}_i = y_i(\omega^T x_i + b) = y_i f(x_i)$. The geometrical margin $\widetilde{\gamma} = y\widehat{\gamma} = \widehat{\gamma}/\|\omega\|$ is obtained by normalizing ω, and it is simplified as $\widetilde{\gamma} = 1/\|\omega\|$. The objective is to obtain the maximum value of $\widetilde{\gamma}$. It is equivalent to obtain the minimum value of $\|\omega\|$. Finally, the problem translates into the quadratic programming problem as in (1), where C is penalty coefficient, and ξ_i is slack variable.

The Lagrange duality translation is conducted for (1). And (1) translates into dual problem as (2):

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i\left(w^T x + b\right) \geq 1 - \xi_i, \quad i = 1, \ldots, l \end{aligned} \tag{1}$$

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \left\langle x_i, x_j \right\rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l, \ \sum_{i=1}^{l}\alpha_i y_i = 0. \end{aligned} \tag{2}$$

The optimal separating function is shown as

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i \left\langle x_i, x \right\rangle + b\right). \tag{3}$$

For nonlinear problem, the kernel function is used to translate nonlinear problem in low dimensional space into linear problem in high dimensional space. The optimal separating function is shown as

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i K\left\langle x_i, x \right\rangle + b\right). \tag{4}$$

The staple kernel functions are shown as (5)~(8). In this paper, the radial basis function (RBF) as (7) is used.

Linear kernel is

$$f(x) = x'x. \tag{5}$$

Polynomial kernel is

$$f(x) = \left(\gamma \cdot x' \cdot x + b\right)^d. \tag{6}$$

RBF kernel is

$$f(x) = \exp\left(-\gamma \cdot \left\|x - x'\right\|^2\right). \tag{7}$$

Sigmoid kernel is

$$f(x) = \tanh\left(\gamma \cdot x' \cdot x + b\right). \tag{8}$$

The main influencing factor of recognition performance is the parameters used in SVM. Presently the staple methods to select optimal parameters include grid search [16], genetic algorithm (GA) [17], and particle swarm optimization (PSO) [18] algorithm. In [19], Dorigo et al. proposed ant colony optimization (ACO) algorithm to select optimal parameters value, achieving better classification performance while taking more time. In [20, 21], Xu et al. and Tiwari and Vidyarthi proposed quantum genetic algorithm (QGA) to optimize SVM parameters and verified that quantum operation can increase the scope of the search space and has good searching ability. In [4], Jiang et al. used quantum simulated annealing (QSA) algorithm combined QGA and simulated annealing (SA) algorithm [22] to optimize SVM parameters, tested the classification model based upon pancreatic images, and achieved better and stable accuracy.

2.2. Fruit Fly Optimization Algorithm. Fruit fly optimization algorithm (FOA) [15] is a new method based on fruit fly foraging behavior for global optimization. The flowchart of FOA is shown in Figure 1.

The key steps of FOA are shown as follows.

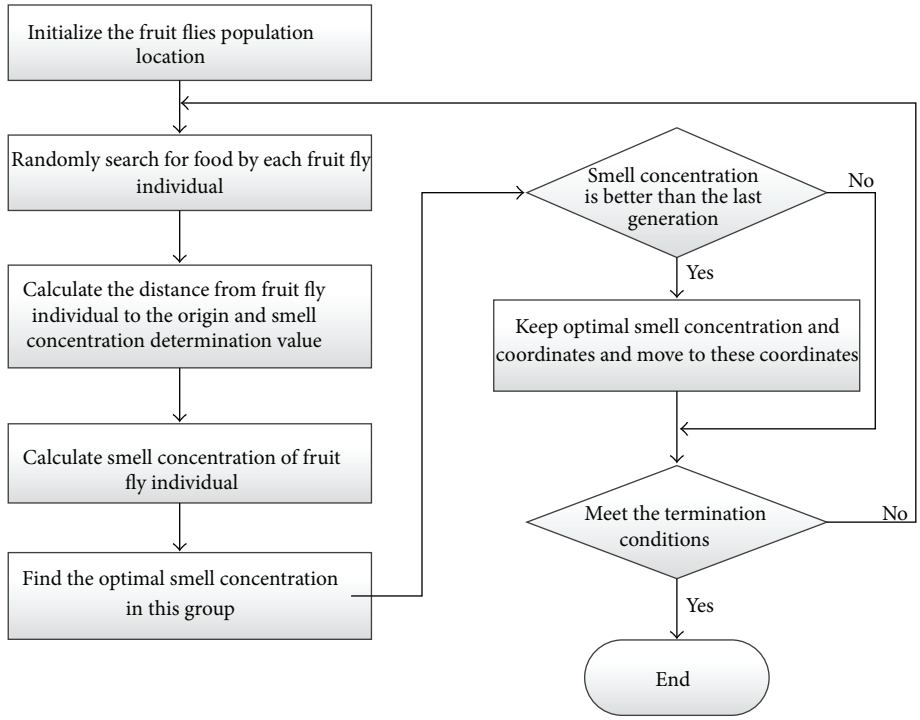


Figure 1: The flowchart of FOA.

Step 1. The position of population, (X_axis, Y_axis), is randomly initialized. X_axis and Y_axis are abscissa value and ordinate value of population's position, respectively.

Step 2. For each fruit fly, the direction and position of flying are randomly evaluated. It is represented as (9). (X_i, Y_i) is the new position of each fruit fly, where $i \in [1, M]$. M is the number of fruit flies in population:

$$\begin{aligned} X_i &= X\text{\_axis} + \text{Random Value} \\ Y_i &= Y\text{\_axis} + \text{Random Value}. \end{aligned} \tag{9}$$

Step 3. The distance (Dist_i) from each fruit fly to the origin and the smell concentration determination value (S_i) of each fruit fly are calculated as

$$\begin{aligned} \text{Dist}_i &= \sqrt{X_i^2 + Y_i^2} \\ S_i &= \frac{1}{\text{Dist}_i}. \end{aligned} \tag{10}$$

Step 4. The smell concentration determination value is used in smell concentration determination function (fitness function) to calculate the smell concentration value as

$$\text{Smell}_i = \text{Function}(S_i). \tag{11}$$

Step 5. The fruit fly which has the best smell concentration is found in population:

$$[\text{bestSmell bestIndex}] = \max(\text{Smell}). \tag{12}$$

Step 6. The best smell concentration and its position (x, y) are saved. The fruit fly population moves to this position by vision.

Step 7. Step 2 to Step 5 are iterated. If the smell concentration is better than previous one, Step 6 is executed.

FOA is one of the intelligent optimization algorithms. It is easy to set up, easy to implement, and fast to optimize. But it also has some problems. In the phase of parameter initialization, FOA uses randomized strategy to determine initial point position. In the phase of fruit fly individual position update, blind search strategy is used. It is slow to converge and easy to fall into extreme values. At present, there are a number of evaluation criteria for classifier performance. In classifier optimization algorithms, classification accuracy and error rate are always used as the fitness function. But those criteria cannot reflect clinical prior knowledge. It is simply to evaluate an operating point and not strong enough when the distribution of class is changed.

3. Methodology

The whole procedure of the proposed method is shown in Figure 2. The detailed process of the proposed method is as follows.

(1) Feature Extraction. We extract gray and fractal dimension features from the segmented pancreatic images, and then we normalize those features.

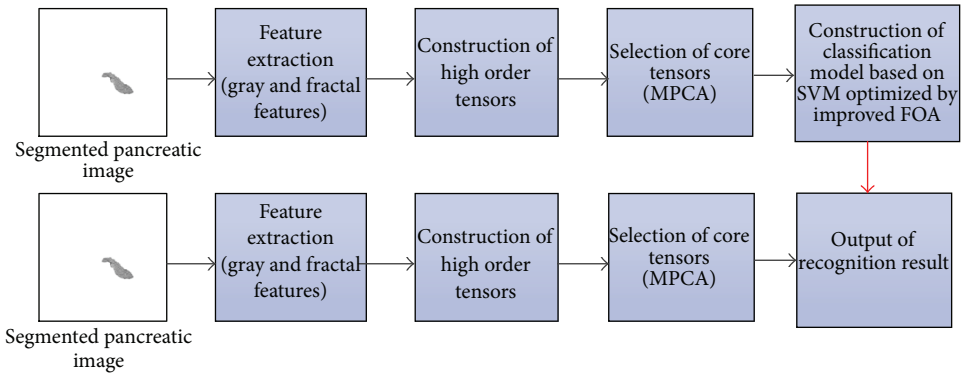


Figure 2: The flowchart of the proposed method.

(2) High Order Tensors Construction. High order tensors are constructed based on the extracted features to represent pancreatic images.

(3) Feature Selection. In this paper we use the MPCA method to extract the eigen tensors for classification.

(4) Pancreatic Cancer Classification. After we obtain the eigen tensors by MPCA, we can treat the eigen tensors as input samples, and then we use the approach of SVM optimized by improved FOA to train classification model of pancreatic cancer image.

In the process, high order tensors construction and feature selection are carried out in accordance with [4]. So in this paper, we will no longer discuss them.

3.1. Improved FOA. Aiming at the existing problem of FOA, we introduce quantum to FOA and redefine a new fitness function as the smell concentration determination function.

3.1.1. Quantum Fruit Fly Coding. In improved FOA (IFOA), quantum phase is used to code fruit flies' position. Compared with FOA which has the same number of fruit flies, the solution search space of quantum fruit flies is the double of the original fruit flies. The quantum fruit flies population position is shown as (13). When initializing, the quantum bit phase angle is $\theta = \pi \cdot (2 \cdot \text{rand} - 1)$, where $\text{rand} \in [0, 1]$, $j \in [1, n]$, and n is the dimension of optimization problem. In this paper, $n = 2$:

$$\begin{aligned} X\_\text{axis} &= \left[\theta_{x1}, \theta_{x2}, \ldots, \theta_{xj}\right] \\ Y\_\text{axis} &= \left[\theta_{y1}, \theta_{y2}, \ldots, \theta_{yj}\right]. \end{aligned} \tag{13}$$

3.1.2. Quantum Fruit Fly Smell Concentration Determination Value. As quantum phase is used to code fruit flies' position, each fruit fly has two solutions, namely, the cosine solution and the sine solution. The distance ($\text{Dist}_j^{(i)}$) from the ith fruit fly to the origin and the smell concentration determination value ($S_j^{(i)}$) of the ith fruit fly can be calculated as

$$\text{Dist}_j^{(i)} = \begin{bmatrix} \sqrt{\cos^2\theta_{xj}^{(i)} + \cos^2\theta_{yj}^{(i)}} \\ \sqrt{\sin^2\theta_{xj}^{(i)} + \sin^2\theta_{yj}^{(i)}} \end{bmatrix} \tag{14}$$

$$S_j^{(i)} = \frac{\text{Dist}_j^{(i)}}{\sqrt{2}}, \tag{15}$$

where $i \in [1, M]$, and M is the number of the fruit flies' populations. In (15), Dist is normalized to $[0, 1]$ and then assigned to S. The reason is to facilitate parameters zooming for optimizing SVM.

3.1.3. Quantum Fruit Fly Smell Concentration Determination Function. False negative rate (FNR) is known as the rate of missed diagnosis. It is the percentage of actual sickness while identified as disease-free. FNR is complementary with the actual diagnostic sensitivity. False positive rate (FPR) is known as the misdiagnosis rate. It is the percentage of the actual disease-free while identified as sickness. FPR is complementary with the actual diagnostic specificity. In the process of actual disease diagnosis, if diagnosis with high sensitivity is used, the higher is the sensitivity, the less is the rate of missed diagnosis. That is to say, FNR is low. When diagnosis with high specificity is used, the misdiagnosis rate is low. That is to say, FPR is low. Therefore, in improved FOA, the mean of weighted sum of FNR and FPR in k-fold cross-validation is used as the smell concentration determination function. It is shown as

$$\text{Fitness} = \frac{1}{K}\sum_{k=1}^{K}\left[w \cdot \text{FNR} + (1 - w) \cdot \text{FPR}\right]. \tag{16}$$

In (16), K is the parameter of k-fold cross-validation, and w is the weight of FNR. If a fruit fly has small smell concentration value, it is good.

3.1.4. Quantum Fruit Fly Mutation Operation. Quantum *not* gate is used to randomly change quantum fruit flies' positions.

Figure 3: The framework of classifier construction.

It not only increases the diversity of the population, but also avoids precocity. The quantum *not* gate based on phase coding is shown as

$$\theta_j^{(i)} = \frac{\pi}{2} - \theta_j^{(i)}. \tag{17}$$

The mutation probability of an individual fruit fly is P_m. If P_m is greater than a random number within $(0, 1)$, the two probability amplitudes of X-coordinate or Y-coordinate of the individual fruit fly randomly selected will be exchanged by quantum *not* gate.

The acceptance probability of mutated new fruit fly position obeys the Boltzmann probability distribution. It is shown as

$$P\left(\theta^{(i)*} \Longrightarrow \theta^{(i)}\right) = \begin{cases} 1, & F\left(\theta^{(i)*}\right) < F\left(\theta^{(i)}\right) \\ P_i, & F\left(\theta^{(i)*}\right) \geq F\left(\theta^{(i)}\right) \end{cases}$$
$$P_i = \left(1 + \exp\left[\frac{F\left(\theta^{(i)}\right) - F\left(\theta^{(i)*}\right)}{l}\right]\right)^{-1}. \tag{18}$$

In (18), $\theta^{(i)*}$ is the mutated new fruit fly position, $\theta^{(i)}$ is the original fruit fly position, $F(\cdot)$ is the smell concentration determination function, and l is iterations. If $F(\theta^{(i)*}) < F(\theta^{(i)})$, the new position will be accepted by probability 1. Otherwise, the new position will be accepted by probability P_i.

3.2. Construction of SVM Classifier Based on Improved FOA. The framework of classifier construction and the flowchart of SVM parameter optimization based on improved FOA are shown in Figures 3 and 4, respectively. The process of classifier construction consists 3 steps, namely, obtaining classifier parameters, training classifier, and testing classifier.

The parameters of SVM, penalty factor C, and RBF kernel function parameter γ have great influence on the performance of classifier. C determines the promotion ability of SVM. The small value of C represents the penalty of empirical error being small, which can lead to "underfitting study." The large value of C represents the penalty of empirical error being large, which can lead to "overfitting study." The optimal value of C is different according to different data subspace, and selecting the optimal value of C can make the promotion ability better. SVM can map the input data of low dimensional space into high dimensional space by the kernel function. Vapnik [14] has found that the parameters of kernel function and penalty factor C have great influence on the performance of SVM. So the selection of parameters of penalty factor C and RBF kernel function parameter γ is important.

The detailed process for optimizing SVM parameters, penalty factor C, and RBF kernel function parameter γ is as follows.

Step 1. The population position (X_axis, Y_axis) is initialized by (13).

Step 2. For each fruit fly, the position and the direction of flying are randomly evaluated. It is shown as

$$X_i = X\text{\_axis} + V_x^{(i)}$$
$$Y_i = Y\text{\_axis} + V_y^{(i)}. \tag{19}$$

In (19), $V \in [-1, 1]$, $i \in [1, M]$, and M is the number of individuals in population.

Step 3. The distance from each fruit fly to the origin and the smell concentration determination value is calculated as (14) and (15).

Step 4. The smell concentration determination value is zoomed to get C and γ. It is shown as (20). Cm and gm are zoom multiples of C and γ, which can be obtained by prior knowledge:

$$C = Cm \cdot S_1^{(i)}$$
$$\gamma = gm \cdot S_2^{(i)}. \tag{20}$$

Step 5. The smell concentration is calculated by (16). We set $K = 5$ and will discuss the value of w in the next section.

Step 6. If the individual fruit fly meets the mutation condition, the mutation operation will be done as (17) and (18).

Step 7. The fruit fly which has the best smell concentration is found as (21). bestSmell is the best smell concentration, bestIndex is the individual fruit fly which has the best smell concentration, and bestPos is the position of best smell concentration of the individual fruit fly:

$$\begin{bmatrix} \text{bestSmell} \\ \text{bestIndex} \\ \text{bestPos} \end{bmatrix} = \min(\text{Smell}). \tag{21}$$

Step 8. The axes and position of the best smell concentration are saved. The fruit fly population moves to this position by vision. It is shown as

$$\text{SmellBest} = \text{bestSmell} \tag{22}$$

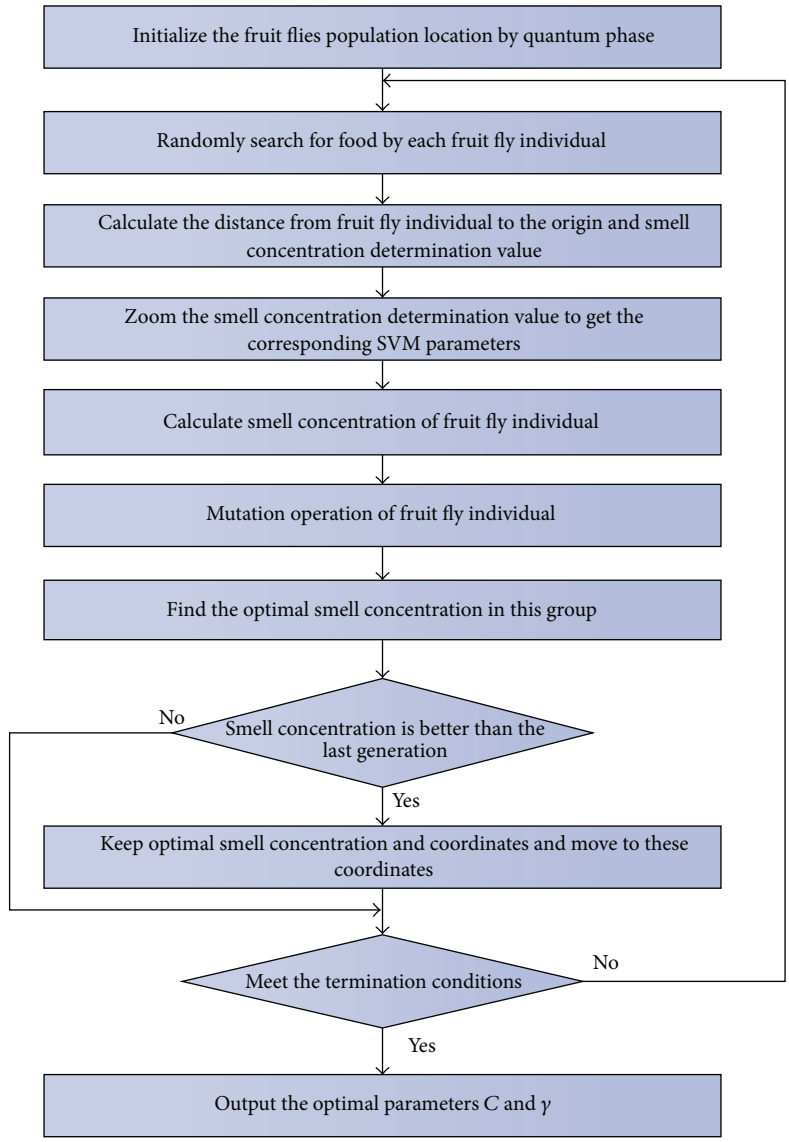


Figure 4: The flowchart of SVM parameter optimization based on improved FOA.

$$\text{Posbest} = \text{bestPos} \tag{23}$$

$$X\_\text{axis} = X\,(\text{bestIndex}) \tag{24}$$

$$Y\_\text{axis} = Y\,(\text{bestIndex})\,. \tag{25}$$

Step 9. Step 2~Step 7 are iterated. If the smell concentration is better than previous one, Step 8 is executed. If the termination condition is satisfied, the optimum parameters will be returned.

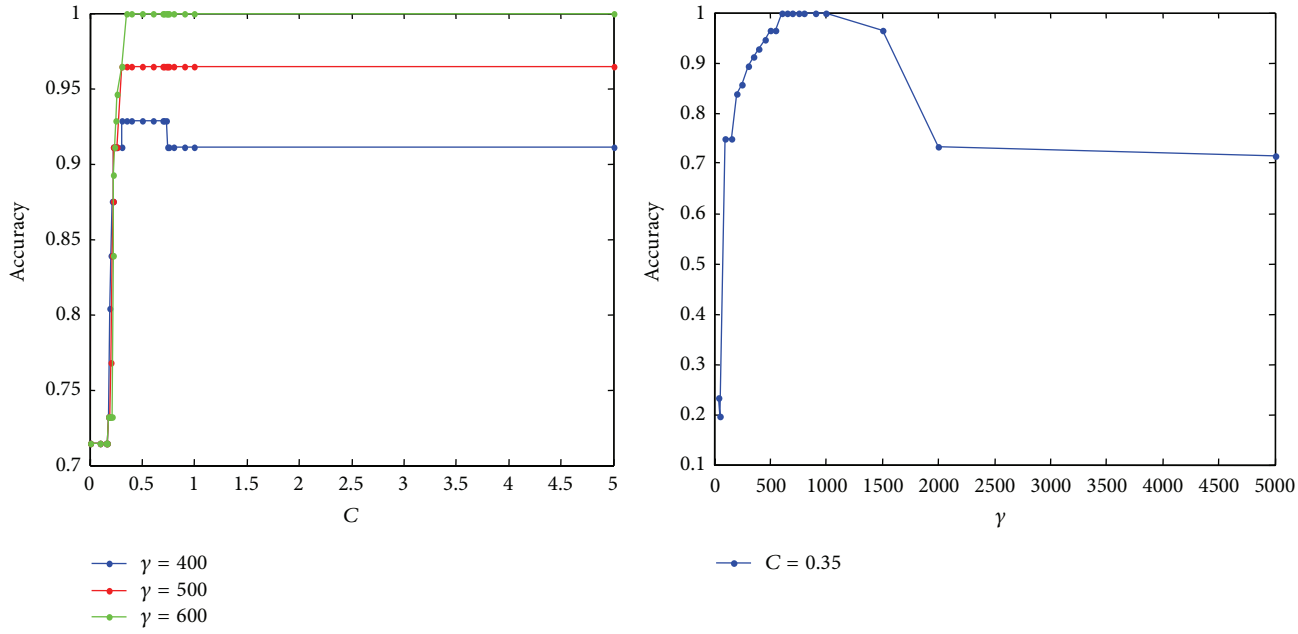
4. Results and Discussion

4.1. Experimental Data. In this paper, abdominal CT images are used in experiments, which are provided by the radiology department of a hospital in Shenyang, China. Their resolution is 512 × 512 pixels, the scan slice thickness is 2 mm, and the format is DICOM. For the purpose of algorithm simulation, the DICOM image is transformed into BMP image. The grayscale is 256 and the resolution is 128 × 128. The detailed information of dataset is shown in Table 1.

Table 1: Experimental data.

	Training samples	Testing samples	Total
Pancreatic cancer images (positive)	17	16	33
Normal images (negative)	41	40	81
Total	58	56	

4.2. Evaluation Criteria. According to the hybrid matrix, which is shown in Table 2, the evaluation criteria are calculated. In this paper, evaluation criteria consist of False Positive



(a) The influence of parameter C

(b) The influence of parameter γ

Figure 5: The effect of parameters C and γ on the classification accuracy.

Table 2: Hybrid matrix.

	Predicted positive example (P′)	Predicted negative example (N′)
Practical positive example (P)	True positive example (TP)	False negative example (FN)
Practical negative example (N)	False positive example (FP)	True negative example (TN)

Rate (FPR), False Negative Rate (FNR), Accuracy, Precision, $F1$ value, and the running time of the algorithms. The mean square errors of evaluation criteria in many experiments are also used to evaluate the stability of the algorithm:

$$\begin{aligned} \text{FPR} &= \frac{\text{FP}}{\text{N}} \\ \text{FNR} &= \frac{\text{FN}}{\text{P}} \\ \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \\ \text{Precision} &= \frac{\text{TP}}{\text{P}'} \\ F1 &= \frac{2}{1/\text{Precision} + 1/\text{Recall}}. \end{aligned} \tag{26}$$

4.3. Prior Knowledge. Because of the sensitivity of initial scope for parameters optimization, we make the statistical analysis for the penalty factor C and RBF kernel function parameter γ, which obtains the prior knowledge of the parameters. The result is shown as in Figure 5.

From Figure 5, we can obtain the initial scope of C and γ by QSA that is $[0.1, 1]$ and $[50, 2000]$, respectively. And the scope of optimal solution is $[0.3, 0.5]$ and $[500, 1500]$, respectively. The scaling of C and γ is 1 and 2000, respectively.

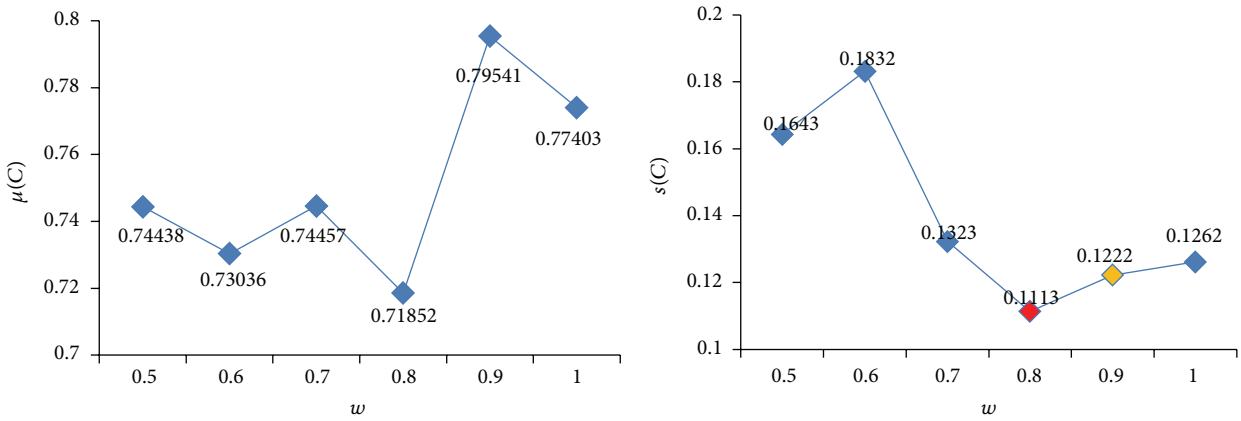
4.4. Determination of FNR Weight. In an actual treatment, a patient was ill, but he was diagnosed as disease-free; then the treatment progress would be delayed and the cure opportunity would be reduced. On the contrary, if a patient was disease-free and was diagnosed with illness, patient would undergo further examination to make up the mistake. Therefore, it is believed that FNR is more important than FPR. The weight of FNR should be greater than FPR; that is to say, $0.5 \leq w \leq 1$.

For different values of w, the experiment was run 10 times. Then we compared the mean value of evaluation criterions and their mean square error.

Figures 6 and 7 show the states of optimized parameters C and γ using different w. Figure 8 illustrates the states of FNR and FPR. Figure 9 demonstrates the states of Accuracy, Precision, and $F1$.

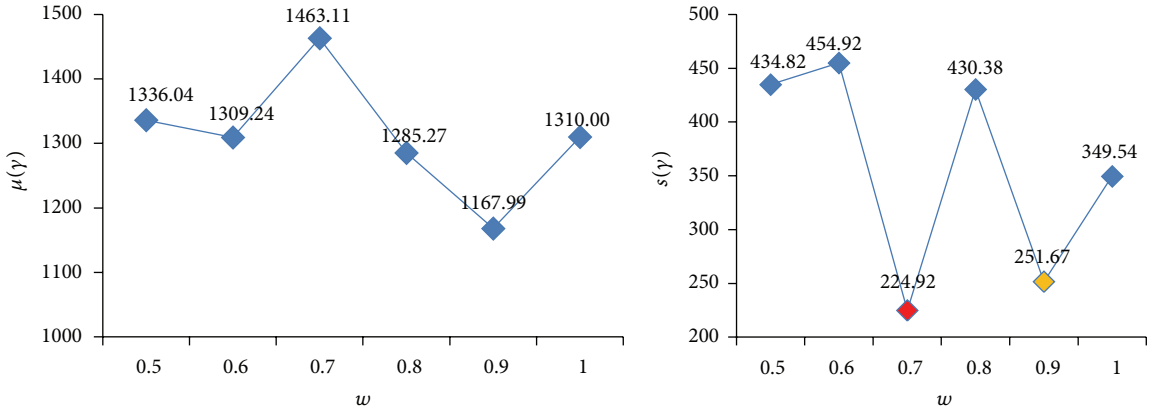
From Figures 6 and 7, it can be seen that the optimized parameters are found to be in line with prior knowledge. According to the mean square error of C, when $w = 0.8$, the algorithm is the most stable, and $w = 0.9$ is the second. According to the mean square error of γ, when $w = 0.7$, the algorithm is the most stable, and $w = 0.9$ is the second.

The better performance of the algorithm comes when the values of FNR and FPR are smaller. From Figure 8, for FPR, when w is 0.9, its mean value and mean square error are the



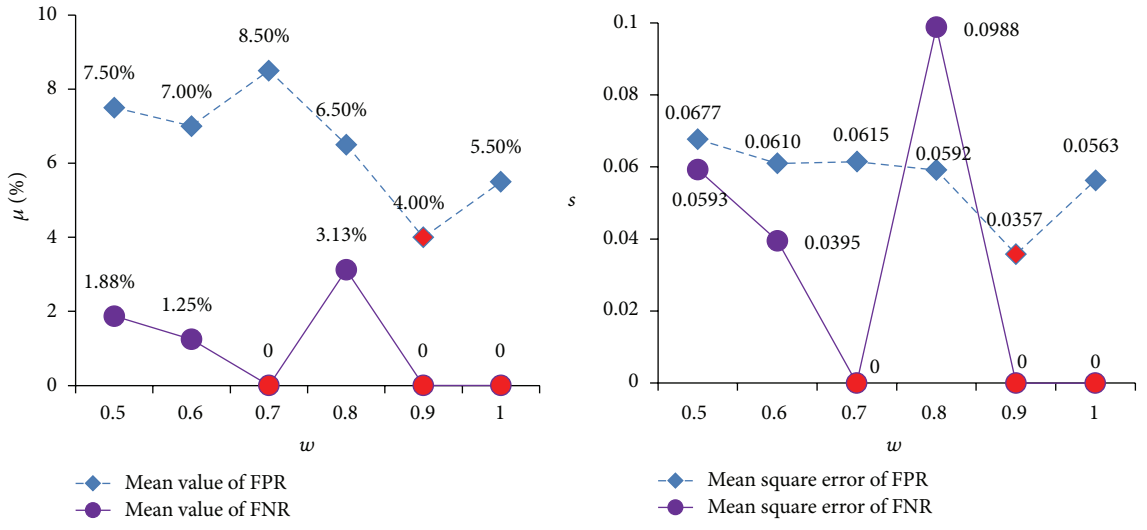
(a) Mean value of C (b) Mean square error of C

Figure 6: The status of parameter C for different w.



(a) Mean value of γ (b) Mean square error of γ

Figure 7: The status of parameter γ for different w.

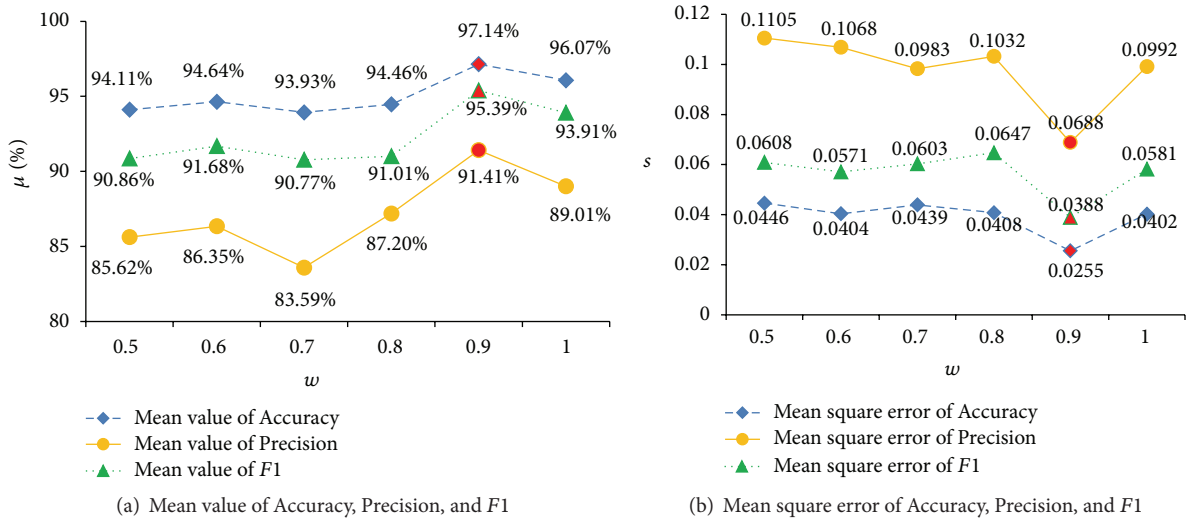


(a) Mean value of FPR and FNR (b) Mean square error of FPR and FNR

Figure 8: FPR and FNR status for different w.

Table 3: Experimental results of IFOA-SVM.

	C	γ	FPR	FNR	Accuracy	Precision	$F1$	Time (s)
1	0.8666	878.79	0	0	100.00%	100.00%	100.00%	31.78
2	0.7089	1003.4	2.50%	0	98.21%	94.12%	96.97%	31.64
3	0.7015	952.7641	2.50%	0	98.21%	94.12%	96.97%	31.35
4	0.8605	1401.3	5.00%	0	96.43%	88.89%	94.12%	30.49
5	0.9922	1314	5.00%	0	96.43%	88.89%	94.12%	31.09
6	0.8319	1364.5	5.00%	0	96.43%	88.89%	94.12%	31.28
7	0.6345	1050.6	2.50%	0	98.21%	94.12%	96.97%	30.7
8	0.9405	820.0642	0	0	100.00%	100.00%	100.00%	30.9
9	0.7671	1384.7	5.00%	0	96.43%	88.89%	94.12%	31.06
10	0.6504	1509.8	12.50%	0	91.07%	76.19%	86.49%	31.68
μ	0.79541	1167.99183	4.00%	0	97.14%	91.41%	95.39%	31.197
s	0.1222	251.6716	0.0357	0	0.0255	0.0688	0.0388	



(a) Mean value of Accuracy, Precision, and $F1$

(b) Mean square error of Accuracy, Precision, and $F1$

Figure 9: Accuracy, Precision, and $F1$ status for different w.

smallest; for FNR, when w is 0.7, 0.9, or 1, its mean value and mean square error are the smallest. Greater values of Accuracy, Precision, and $F1$ can lead to better performance of the algorithm. From Figure 8, when $w = 0.9$, its mean values of Accuracy, Precision, and $F1$ are the greatest, and the mean square error is the smallest. Therefore, the final value of w is determined as 0.9.

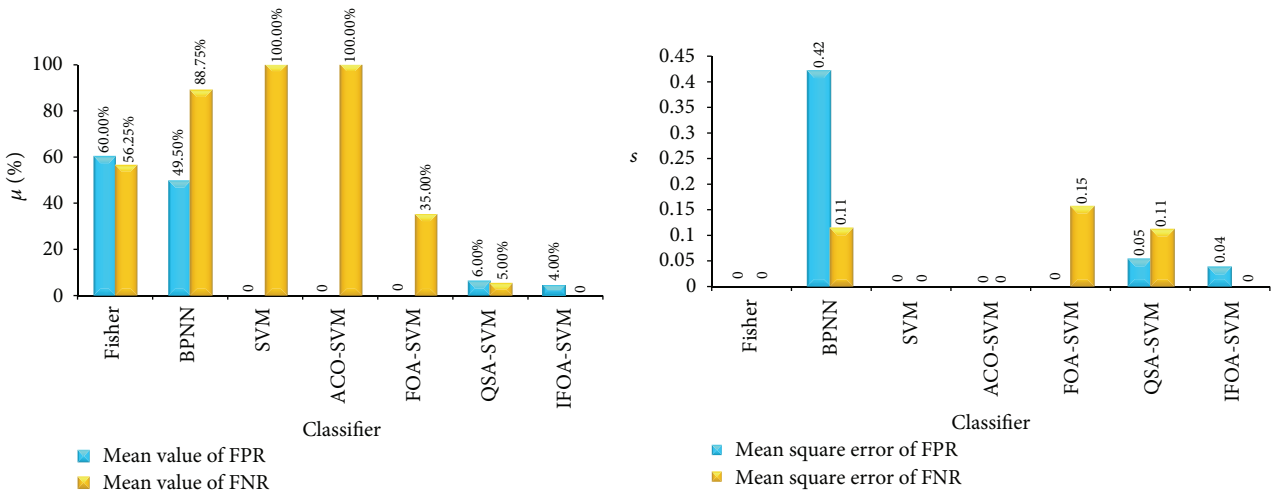
4.5. Experimental Results and Analysis. Ten experiments of SVM optimized by improved FOA (IFOA-SVM) are randomly done. The experimental result is shown in Table 3.

From Table 3, it is known that the mean values of C and γ are 0.79541 and 1167.99183, respectively. The average of FPR is 4%, FNR is 0, Accuracy is 97.14%, Precision is 91.41%, $F1$ is 95.39%, and the time is 31.197 s.

Compared with other classifiers, the performance of IFOA-SVM is better as shown in Figures 10 and 11. In Figure 12, the comparison of running time is shown. Classifier Fisher is the Fisher linear classifier, classifier BPNN is the BP neural network, SVM is the common SVM, and ACO-SVM, FOA-SVM, and QSA-SVM are the optimized classifier SVM using ant colony algorithm, fruit fly optimal algorithm, and quantum simulated annealing, respectively. IFOA-SVM is the proposed method.

In Figure 10, FNR achieved 100% and FPR is 0 by SVM and ACO-SVM; that is to say, all patients are diagnosed free from diseases. This situation is not allowed in actual diagnosis. FNR of BPNN and Fisher are 88.75% and 56.25%, and FPR are 60% and 49.5%. So BPNN and Fisher lack credibility. FPR and FNR of FOA-SVM are 0 and 35%. It sometimes occurs in missed diagnosis situation. FPR and FNR of QSA-SVM are 6% and 5%. It might occur in missed diagnosis to few patients. FPR and FNR of the proposed IFOA-SVM are 4% and 0. It is better than other methods. IFOA-SVM achieves the best sensibility and stability.

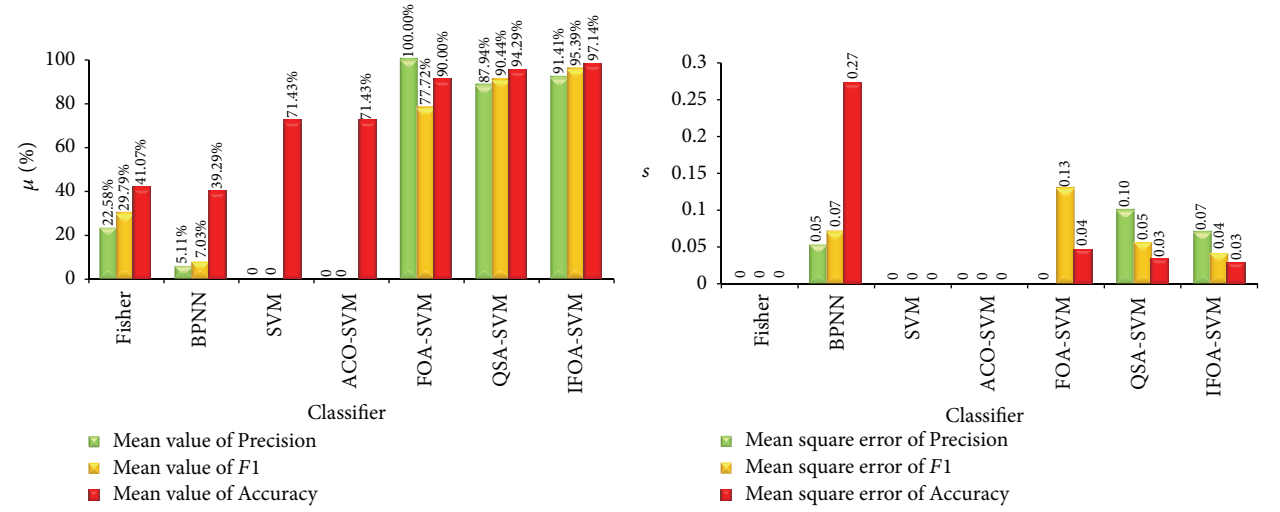
In Figure 11, for the average of Precision, FOA-SVM is the best, which is 100%. IFOA-SVM takes the second place, which is 91.41%. And from Figure 11(b), FOA-SVM is the most stable. For the average of $F1$, the proposed IFOA-SVM



(a) Mean value of FPR and FNR for different classifiers

(b) Mean square error of FPR and FNR for different classifiers

Figure 10: FPR and FNR status for different classifiers.



(a) Mean value of Precision, $F1$, and Accuracy for different classifiers

(b) Precision, $F1$, and Accuracy mean square errors for different classifiers

Figure 11: Accuracy, Precision, and $F1$ status for different classifiers.

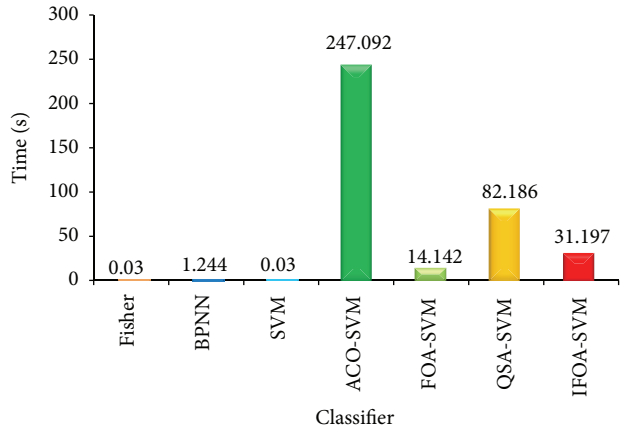
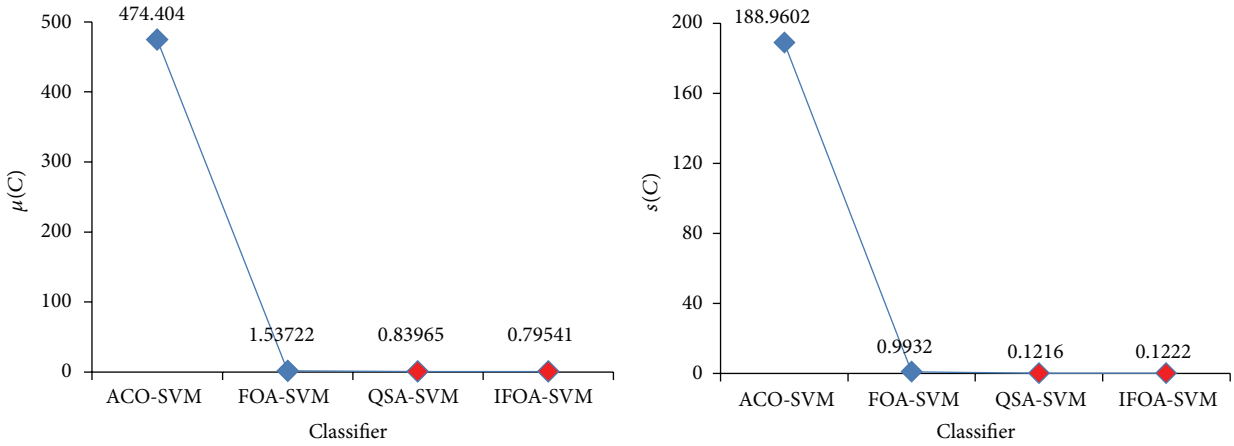


Figure 12: Running time for different classifiers.

achieves 95.39%, which is optimal. QSA-SVM is 90.44%, which takes the second place. But IFOA-SVM is more stable than QSA-SVM. For the average of Accuracy, IFOA-SVM is 97.14%, which is the best. QSA-SVM is 94.29%. FOA-SVM is 90%. SVM and ACO-SVM are 71.43%. Fisher and BPNN are less than 50%. Compared with mean square error of Accuracy, IFOA-SVM is the most stable.

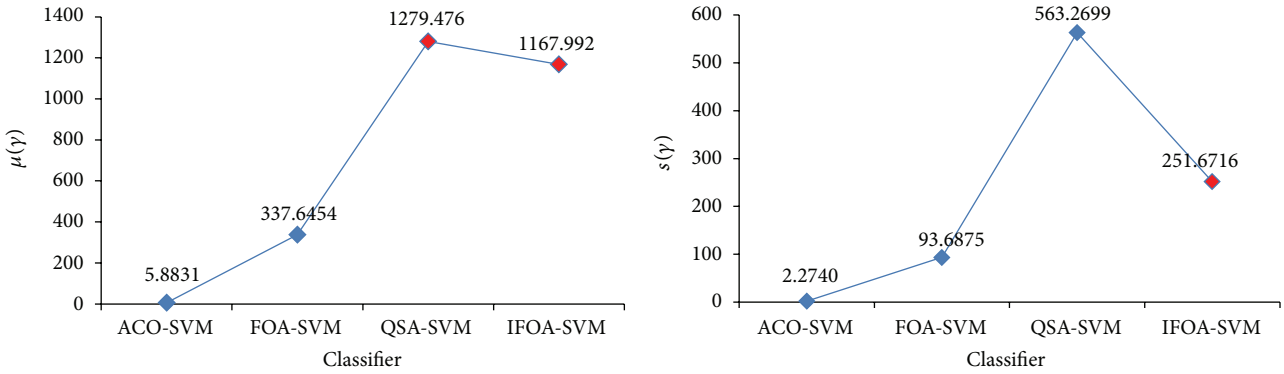
In Figure 12, Fisher and SVM cost the least time, which is 0.03 s. BPNN is 1.244 s. FOA-SVM is 14.142 s. IFOA-SVM is 31.197 s. QSA-SVM is 82.186 s. ACO-SVM cost the most time, which is 247.092 s. In actual diagnosis, less time is better. The proposed IFOA-SVM is not the best but is not out of the way.

ACO-SVM, FOA-SVM, QSA-SVM, and the proposed IFOA-SVM can be used to optimize SVM parameters. In Figures 13 and 14, comparative results of mean value and



(a) Mean value of C (b) Mean square error of C

Figure 13: Parameter C status for different method.



(a) Mean value of γ (b) Mean square error of γ

Figure 14: Parameter γ status for different method.

mean square error of optimal parameters, C and γ, are shown based on those methods.

In the pancreatic cancer classifier based on ACO-SVM, C is oversize and γ is undersize. By using FOA-SVM, optimal parameters are not in estimation interval of prior knowledge, but in terms of mean square error of optimal parameters FOA is stable. When QSA-SVM and IFOA-SVM are used to optimize SVM parameters, optimal parameters are in estimation interval, and the stability of two methods is similar from mean square error of C. And in terms of mean square error of γ, IFOA-SVM is more stable than QSA-SVM.

5. Conclusion

In this paper, we introduced the concept of quantum to FOA to improve it. A new smell concentration determination function was defined in the improved FOA. The improved FOA was used to optimize the parameters of SVM and a classifier was constructed based on the optimized SVM. As an application, pancreatic cancer classifier was established. The proposed method achieved better classification performance. The first reason is that quantum coding and quantum operation increased the diversity of the population and avoided precocity. The second reason is that the redefined smell concentration determination function was more suitable to the actual diagnosis requirements. The third reason is the advantages of FOA which are easy to set up, easy to implement, and fast to optimize. Therefore, the proposed method can improve the classification performance of pancreatic cancer images and then assist doctors in diagnosing diseases.

Conflict of Interests

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

Acknowledgment

The research is supported by the National Natural Science Foundation of China (no. 61272176).

References

[1] A. Jemal, R. Siegel, E. Ward, Y. Hao, J. Xu, and M. J. Thun, "Cancer statistics, 2009," *CA: Cancer Journal for Clinicians*, vol. 59, no. 4, pp. 225–249, 2009.

[2] Y. Wang and P. Zhao, "Advances in early diagnosis of pancreatic cancer," *Oncology Progress*, vol. 4, no. 4, pp. 327–332, 2006.

[3] F. Fraioli, G. Serra, and R. Passariello, "CAD (computed-aided detection) and CADx (computer aided diagnosis) systems in identifying and characterising lung nodules on chest CT: overview of research, developments and new prospects," *Radiologia Medica*, vol. 115, no. 3, pp. 385–402, 2010.

[4] H. Jiang, D. Zhao, T. Feng, S. Liao, and Y. Chen, "Construction of classifier based on MPCA and QSA and its application on classification of pancreatic diseases," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 713174, 7 pages, 2013.

[5] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[6] B. Kovalerchuk, E. Triantaphyllou, J. F. Ruiz, and J. Clayton, "Fuzzy logic in computer-aided breast cancer diagnosis: analysis of lobulation," *Artificial Intelligence in Medicine*, vol. 11, no. 1, pp. 75–85, 1997.

[7] P. C. Pendharkar, J. A. Rodger, G. J. Yaverbaum, N. Herman, and M. Benner, "Association, statistical, mathematical and neural approaches for mining breast cancer patterns," *Expert Systems with Applications*, vol. 17, no. 3, pp. 223–232, 1999.

[8] M. Antonie, O. Zaiane, and A. Coman, "Application of data mining techniques for medical image classification," in *Knowledge Discovery and Data Mining*, pp. 94–101, 2001.

[9] J. Zhang, Y. Wang, Y. Dong, and Y. Wang, "Ultrasonographic feature selection and pattern classification for cervical lymph nodes using support vector machines," *Computer Methods and Programs in Biomedicine*, vol. 88, no. 1, pp. 75–84, 2007.

[10] J. Ramírez, R. Chaves, J. Górriz et al., "Functional brain image classification techniques for early Alzheimer disease diagnosis," in *Bioinspired Applications in Artificial and Natural Computation: Third International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2009, Santiago de Compostela, Spain, June 22–26, 2009, Proceedings, Part II*, vol. 5602 of *Lecture Notes in Computer Science*, pp. 150–157, Springer, Berlin, Germany, 2009.

[11] D. Tsai and K. Kojma, "Enhancement of CT pancreatic features by a simple cascading filter," in *Proceedings of the Nuclear Science Symposium and Medical Imaging Conference, 1993 IEEE Conference Record*, San Francisco, Calif, USA, 1993.

[12] T. Takada, H. Yasuda, K. Uchiyama, H. Hasegawa, T. Iwagaki, and Y. Yamakawa, "A proposed new pancreatic classification system according to segments: operative procedure for a medial pancreatic segmentectomy," *Journal of Hepato-Biliary-Pancreatic Surgery*, vol. 1, no. 3, pp. 322–325, 1994.

[13] S. He, H. J. Cooper, D. G. Ward, X. Yao, and J. K. Heath, "Analysis of premalignant pancreatic cancer mass spectrometry data for biomarker selection using a group search optimizer," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 6, pp. 668–676, 2012.

[14] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[15] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[16] T. Wang, X. Ye, L. Wang, and H. Li, "Grid search optimized SVM method for dish-like underwater robot attitude prediction," in *Proceedings of the 5th International Joint Conference on Computational Sciences and Optimization (CSO '12)*, pp. 839–843, June 2012.

[17] M. Zhao, C. Fu, L. Ji, K. Tang, and M. Zhou, "Feature selection and parameter optimization for support vector machines: a new approach based on genetic algorithm with feature chromosomes," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5197–5204, 2011.

[18] F.-H. Yu and H.-B. Liu, "Structural damage identification by support vector machine and particle swarm algorithm," *Journal of Jilin University Engineering and Technology Edition*, vol. 38, no. 2, pp. 434–438, 2008.

[19] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[20] X. Xu, J. Jiang, J. Jie, H. Wang, and W. Wang, "An improved real coded quantum genetic algorithm and its applications," in *Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN '10)*, pp. 307–310, IEEE, Taiyuan, China, September 2010.

[21] P. K. Tiwari and D. P. Vidyarthi, "A variant of quantum genetic algorithm and its possible applications," *Advances in Intelligent and Soft Computing*, vol. 130, no. 1, pp. 797–811, 2012.

[22] S. Xu, H. Zhao, and Y. Xie, "Grey SVM with simulated annealing algorithms in patent application filings forecasting," in *Proceedings of the International Conference on Computational Intelligence and Security Workshops (CIS '07)*, pp. 850–853, IEEE, Harbin, China, December 2007.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 318217, 10 pages
http://dx.doi.org/10.1155/2015/318217

Research Article

OperomeDB: A Database of Condition-Specific Transcription Units in Prokaryotic Genomes

Kashish Chetal[1] and Sarath Chandra Janga[1,2,3]

[1]*Department of Biohealth Informatics, School of Informatics and Computing, Indiana University-Purdue University Indianapolis (IUPUI), 719 Indiana Avenue, Suite 319, Walker Plaza Building, Indianapolis, IN 46202, USA*
[2]*Center for Computational Biology and Bioinformatics, Indiana University School of Medicine, 5021 Health Information and Translational Sciences (HITS), 410 West 10th Street, Indianapolis, IN 46202, USA*
[3]*Department of Medical and Molecular Genetics, Indiana University School of Medicine, Medical Research and Library Building, 975 West Walnut Street, Indianapolis, IN 46202, USA*

Correspondence should be addressed to Sarath Chandra Janga; scjanga@iupui.edu

Received 26 November 2014; Accepted 2 March 2015

Academic Editor: Florencio Pazos

Background. In prokaryotic organisms, a substantial fraction of adjacent genes are organized into operons—codirectionally organized genes in prokaryotic genomes with the presence of a common promoter and terminator. Although several available operon databases provide information with varying levels of reliability, very few resources provide experimentally supported results. Therefore, we believe that the biological community could benefit from having a new operon prediction database with operons predicted using next-generation RNA-seq datasets. *Description*. We present operomeDB, a database which provides an ensemble of all the predicted operons for bacterial genomes using available RNA-sequencing datasets across a wide range of experimental conditions. Although several studies have recently confirmed that prokaryotic operon structure is dynamic with significant alterations across environmental and experimental conditions, there are no comprehensive databases for studying such variations across prokaryotic transcriptomes. Currently our database contains nine bacterial organisms and 168 transcriptomes for which we predicted operons. User interface is simple and easy to use, in terms of visualization, downloading, and querying of data. In addition, because of its ability to load custom datasets, users can also compare their datasets with publicly available transcriptomic data of an organism. *Conclusion*. OperomeDB as a database should not only aid experimental groups working on transcriptome analysis of specific organisms but also enable studies related to computational and comparative operomics.

1. Background

As the gap between the rate at which sequencing of complete genomes and the experimental characterization of transcriptional regulation in them increases, automated computational methods for unravelling the regulatory code are increasingly being sought after. Although accurate tools for gene identification encoded in a genome have been developed, our understanding on how the genes are expressed and regulated depends on our knowledge of how they are organized into operons-genes set that are cotranscribed to produce a single messenger RNA [1, 2]. Operons are the essential units of transcription in prokaryotic organisms and, as a result, identifying these structures is a main step in understanding transcriptional regulation. Knowing operon structure in a genome not only facilitates identifying sets of genes which are coregulated but also aids in other computational analyses, such as prediction of cis-regulatory elements, which often depend on accurate detection of operons. In addition, since operons often consist of genes that are related functionally and required by the cell for numerous biological processes, they are often good predictors of biological modules [3–5]. Therefore, deep understanding of operons will improve our knowledge of higher-order genomic associations and structures thereby expanding our understanding of various cellular networks composed of regulatory, structural, and functional pathways [5, 6]. Operons also provide insights into the cellular functions and also help in determining different

experimental designs. In various recent high-throughput RNA-sequencing studies across a number of prokaryotic organisms, it has been convincingly shown that the structure of operons changes with the environmental conditions [7, 8], thus suggesting a need to the discovery and a better understanding of the transcriptional units originating from operons (predicted or otherwise) across experimental conditions in bacterial genomes. For all these reasons, the characterization of condition-specific transcription unit structure on a genomic scale is an important origin point for microbial functional genomics.

Several operon databases are currently available and provide information with varying levels of reliability and emphasis [9–13]. However, it is important to note that, traditionally, definitions of operons and transcription units are synonymously used for computational predictions, mainly because each operon was believed to encode for a single transcription unit (single polycistronic unit). However, emerging evidence from several RNA-sequencing studies supports a more complex model, with several operons in a genome encoding for multiple transcription units depending on the condition [7, 8]. Databases such as RegulonDB [13], which are based on manual curation of experimentally reported polycistronic transcripts identified in at least one experimental condition in the literature in *Escherichia coli* K-12, define an operon as the ensemble of all the transcription units in a given genome loci which results in the longest stretch of codirectional transcript. In such frameworks, each transcription unit is governed by a promoter and terminator identified in at least one condition. In contrast, working definition for computational prediction of operons across most studies simply assumes the longest possible polycistronic transcript in a genomic locus as an operon. These differences in the working definition indicate that the current prediction pipelines and databases for operon prediction are perfect in predicting condition-specific transcription units/operons in bacterial genomes. In OperonDB, Pertea et al. employed a method to find and analyze gene pairs that are located on the same strand of DNA in two or more bacterial genomes [10]. The computational algorithm used in this database locates operons structure in microbial genomes using a method published earlier by the authors [14]. OperonDB currently contains 1059 genomes with prediction sensitivity of 30%–50% in *Escherichia coli* [10]. DOOR (database for prokaryotic operons) is another database, which contains predicted operons for 675 sequenced prokaryotic genomes. It provides similarity scores between operons by which user can search for related operons in different organisms [9]. ProOpDB (prokaryotic operon database) predicts operons in more than 1200 prokaryotic genomes using a neural network based approach. It provides several options for retrieving operon information. In ProOpDB, users can also visualize operons in their genomic context and their nucleotide or amino acid sequences [11].

MicrobesOnline is another operon database, which facilitates the phylogenetic analysis of genes from microbial genomes [12]. In principle, this database has two functionalities: (1) user can build a phylogenetic tree for every gene family as well as a species tree in a tree-based browser to assist in gene annotation and in reconstructing their history of evolution and (2) by using its tool one can analyze microarray data to find genes which exhibit similar expression profiles in an organism which can subsequently be used for identifying regulatory motifs and seeing if they are conserved. User can also compare the organization of a protein domain with genes of interest in a browser [12]. Finally, as mentioned above, RegulonDB is a database [13], which is curated and designed for *Escherichia coli* K-12 to facilitate the prediction of its transcriptional regulatory network and operon organization across growth conditions. It also provides extensive information about the evolutionary conservation of a number of regulatory elements in *Escherichia coli* genome. The method used for predicting operons in RegulonDB has a maximum accuracy of 88% for the identification of pairs of adjacent genes in operon and it also identifies 75% of the known transcription units when used to predict the transcription organization of *Escherichia coli* [13]. However, there is not a single database present, which uses data from RNA-sequencing experiments to predict the transcription unit organization in a broad range of bacterial genomes in a condition-specific manner. In this study, we present operomeDB (http://sysbio.informatics.iupui.edu/operomeDB/#/) to address this gap—a database dedicated to the identification and visualization of transcriptional units from publicly available RNA-seq data in microbial genomes.

High-throughput sequencing platforms like Illumina, ABI, and Roche are used to quantify the expression levels of RNA in a condition-specific manner in bacterial genomes—frequently referred to as an RNA-seq experiment. Such high-throughput technologies have several advantages compared to traditionally used microarray platforms like a low background signal, large dynamic range of expression level, and possibility of detecting novel transcripts. There are different tools for detection, management, and analysis of the eukaryotic RNA-seq data; however relatively very few tools are available for the analysis and processing of RNA-seq data in prokaryotes. Rockhopper is an open source computational algorithm implemented for the analysis of bacterial RNA-seq data [15]. It supports different stages of RNA-seq analysis and datasets from different sequencing platforms. The algorithm performs several functions such as aligning the sequence reads to a genome, constructing transcriptome maps, calculating the abundance of transcripts, differential gene expression, and predicting transcription unit structure. It also has the ability to detect novel small RNAs, operons, and transcription start sites with a high accuracy in a transcriptome-specific manner [15].

Although there are many tools and software available for the visualization and exploration of next-generation sequencing datasets for eukaryotic organisms, there is a lack of proper genome browsers to visualize prokaryotic organisms and transcriptomes in particular. The size of data generated by RNA-sequencing methods is usually large and makes data visualization a challenging task. IGV (Integrative Genomic Viewer) is a visualization tool that can visualize large data sets very smoothly with the main aim of helping the researchers to visualize and explore the results [16]. The UCSC and Ensembl

genome browsers are online tools that have been used to display different biological datasets, including genomic variants, expressed sequence tags, and functional genomic data with manually curated annotations [17, 18]. In this study, we used jBrowse to develop visualization of predicted transcription units for each RNA-seq dataset analyzed across genomes. jBrowse is an open source, portable, JavaScript based genome browser particularly suitable for prokaryotic genomes. The browser provides easy navigation of genome annotations on the web and has good track selection, zooming, panning, and navigation features [19].

We believe that biological community could benefit from having a new operon prediction database, which uses RNA-seq datasets to predict transcription units in a condition/transcriptome-specific manner. In our presented database (operomeDB) for bacterial genomes, we used an innovative approach to query operons. We predict operons for nine bacterial genomes for which at least few RNA-seq datasets are available in the public domain from the Sequence Read Archive (SRA) of NCBI [20]. We used Rockhopper [15] for the computational analysis of data. Using RNA-seq data for different bacterial genomes, the developed database, which to our knowledge is the largest of its kind to date, should facilitate to researchers navigating through operons predicted under different experimental conditions.

2. Construction and Content

2.1. Dataset Acquisition and Composition of the Datasets. We collected RNA-seq datasets for various bacterial species under a number of different conditions from the Sequence Read Archive (SRA) of NCBI [20] as described below. Descriptions of the bacterial genomes have been taken from Wikipedia and other cited resources as appropriate.

2.2. Escherichia coli K-12 MG1655. Escherichia coli is generally found in the colon and large intestine of the warm-blooded organisms. It belongs to a family of K-12 and B strains that is used in molecular biology for different experiments and also considered as a model organism. K-12 is the strain first confined from a sample of stool of the patient suffering from diphtheria. Different strains have emerged in years due to various treatment agents [21]. Expression profiling of wild type and SgrR mutant *E. coli* under aMG and 2-DG-induced strain was performed by Wadler and Vanderpool [22]. RNA-sequencing data available for illumine platform for this strain under 54 different conditions was analyzed using Rockhopper [15].

2.3. Eggerthella lenta DSM2243. Eggerthella lenta is an anaerobic, nonmotile, nonsporulating pathogenic gram-positive bacteria isolated from a rectal tumor. It is mostly found in blood and human intestine and can cause severe infections. Temperature favorable for growth of these bacteria is 37-degree Celsius [21]. Expression profiling study carried out for the generation of datasets is based on RNA-seq analysis of *Eggerthella lenta* cultured with or without digoxin. This dataset comprised 21 different transcriptomes in *Eggerthella lenta* DSM2243 strain from Haiser et al. [23].

2.4. Campylobacter jejuni RM1221. Campylobacter species are the prominent cause of gastroenteritis in countries on the path of development. An infection occurring due to *C. jejuni* is the most frequent preliminary cause for a neuromuscular paralysis, which is also known as Guillain-Barre syndrome. Healthy cattle and birds can carry *C. jejuni* [21]. For this study, data from Dugar et al. [24] where the authors did a comparative dRNA-seq analysis of multiple *Campylobacter jejuni* strains revealing conserved and specific transcriptional patterns to this strain was used. For 16 different conditions, RNA-seq data for *Campylobacter jejuni* RM1221 was obtained from this study [24].

2.5. Clostridium beijerinckii NCIMB 8052. C. beijerinckii NCIMB 8052 is anaerobic, motile, rod-shaped bacteria. The anatomy of the cell changes with the progression of growth cycle of the organism. *C. beijerinckii* species are present everywhere in nature and routinely segregated from soil samples [21]. Wang et al. carried out single-nucleotide resolution analysis of the transcriptomic structure of *Clostridium beijerinckii* NCIMB 8052 using RNA-seq technology [25]. This is comprised of expression quantification dataset for 6 different conditions in this organism [25].

2.6. Clostridium difficile 630. C. difficile is commonly found in water, air, human, and animal feces. Its genome reveals that the pathogen thrives in the gastrointestinal tract and some of its strains are more fatal than others. With the help of *C. difficile* genome we can understand the antimicrobial resistance and various treatment options available. After the sequencing of the whole genome, it was found from the whole genome that 11% of it consists of genetic elements such as conjugative transposons. These genetic elements contribute *Clostridium* with the genes subjected for their antimicrobial resistance, interaction to host, and surface structure production [26]. We used data from Fimlaid et al., where the authors conducted a global analysis of genes induced during sporulation of *Clostridium difficile* using Illumina HiSeq 1000 for 18 different conditions [27].

2.7. Mycobacterium tuberculosis H37rv. Mycobacterium is a causative agent of tuberculosis and has a waxy coating on its surface. Primary *Mycobacterium* affects respiratory system and lungs. H37rv strain of tuberculosis has 4 million base pairs with 3959 genes. The genome contains 250 genes that are involved in metabolism of fatty acids. Datasets for this genome are collected from experiments in which authors performed the high-resolution transcriptome and genome wide dynamics of RNA polymerase and NusA [28]. A total of 10 different transcriptomes were collected from this study for *Mycobacterium tuberculosis* [28].

2.8. Salmonella enterica subsp. enterica Serovar Typhimurium str. 14028S. Salmonella enterica serovar is a subspecies of *S. enterica*, which are in the shape of rod, flagellated, aerobic, and gram-negative. *Salmonella* serovar can have many

strains, which allows for accelerated increase in the total number of antigenically variable bacteria. In a study by Stringer et al. [29], authors used RNA-seq to conclude the effects of AraC and arabinose on RNA levels genomewide in *S. enterica*. Wild type or delta AraC mutant cells were developed in the presence and absence of 0.2% L-arabinose. The data for *Salmonella enterica* was collected for 8 different conditions [29].

2.9. Sinorhizobium meliloti 2011. *Sinorhizobium meliloti* is a nitrogen-fixing bacterium. Nitrogen fixation by *S. meliloti* is hampered by the plastic modifier bisphenol A. Dataset used in our database corresponded to a recent study where the authors performed RNA-sequencing of 18 samples corresponding to this bacteria in 3 different conditions [30]. For each condition, both short and long RNA fractions were analyzed, and three replicates per condition and per RNA fraction were performed. In this study next-generation annotation of prokaryotic genomes with Eugene-P was performed—applied to *Sinorhizobium meliloti* 2011 genome [30].

2.10. Synechococcus elongatus PCC 7942. *Synechococcus elongatus* are found in aquatic environments. They are called photosynthetic bacteria, as they are responsible for its production. *Synechococcus* consists of one circular chromosome and two plasmids. This particular strain contains a circular chromosome 2,700,000 bp long with GC content of 55%. For the generation of 17 datasets, three strains (7942, SE01, and SE02) were analyzed by Ruffing at two time points (100 h and 240 h) with three biological replicates [31].

2.11. Prediction of Operons Using Rockhopper. To predict transcription units (operons) in a condition/transcriptome-specific manner, we used Rockhopper, a computational algorithm which supports different stages of RNA-seq analysis for datasets originating from diverse sequencing platforms [15]. Rockhopper takes sequenced RNA reads as input in a number of formats including FASTQ, QSEQ, FASTA, SAM, and BAM files [15]. It allows the processing of next-generation RNA-seq data by permitting the user to specify different parameters to align sequence reads to a genome, such as number of mismatches allowed, orientation of mate-pair reads, and minimum seed length. For transcriptomic analysis in Rockhopper, some parameters specified include whether the dataset is strand specific, test for differential expression, prediction of operons and minimum expression of UTRs, and detection of ncRNAs. However, the authors recommend the use of default settings most of the time for best operon prediction performance and hence in this study we used the default parameters where possible [15]. Indeed, operon prediction by Rockhopper has been shown by the original authors to perform at ~90% accuracy when benchmarked against RegulonDB [13] and DOOR [9] databases. We also compared multigene operons predicted by Rockhopper for the nine genomes studied here using the RNA-seq data and the percentage of operons shared with the DOOR database predictions is shown in Table 1. We found that on average 93% of our predicted operons in a genome were independently confirmed to be operons by DOOR database suggesting the high quality of our operon predictions.

Table 1: Table showing the percentage of multigene operons in the operomeDB found to be overlapping with the operon predictions for the corresponding genomes in the DOOR database.

Bacterial genome	Percentage (overlap)
Clostridium difficile 630	89
Clostridium beijerinckii NCIMB 8052	95
Escherichia coli k-12 mg1655	94
Eggerthella lenta dsm2243	95
Mycobacterium tuberculosis h37rv	91
Salmonella enterica subsp.	91
Synechococcus elongatus pcc7942	92
Campylobacter jejuni rm1221	97

Each run of Rockhopper on a single RNA-seq dataset corresponding to a condition provides different files as output, such as summary file—which contains a summary analysis of successfully aligned reads to genomic regions, transcript file—which includes newly predicted transcripts, transcription start, and stop sites with expression levels. Finally, it provides operons file containing predicted operons in the condition. We ran Rockhopper in a batch mode to process and predict operons in each condition for each genomic dataset discussed above by selecting the appropriate reference sequence. We also ran operon prediction on the complete transcriptomic dataset for each genome to obtain a consensus set of operon predictions which was used to show as a reference operome (operon track) of the organism in jBrowse. In order to index the operons in our database, we numbered them by matching with the IDs of the predicted operons from DOOR database in order to easily know the novel operons. If our predicted operon shared at least one gene we gave the same operon ID as DOOR database and for operons which were not present in DOOR database we marked them as "NA."

3. Utility and Discussion

3.1. Implementation and Interface. We developed an interface using HTML, CSS, and JavaScript and also incorporated jBrowse, a genome viewer to display different tracks for building operomeDB (http://sysbio.informatics.iupui.edu/operomeDB/#/) presented here. User interface of our database is shown in Figure 1 which allows the selection of an organism using a drop-down list. User can select an organism and selected bacterial genome information is displayed. There are multiple view options available for users, like viewing as a table of predicted operons, viewing operons using jBrowse, or downloading the predicted operons as a table (Figure 1). Clicking on the tab with the option of "view in jBrowse" will display data in jBrowse and user can view reference sequence of the genome, gene list in the particular bacterial genome, and a list of operons predicted. User can also select to show the operon data for different

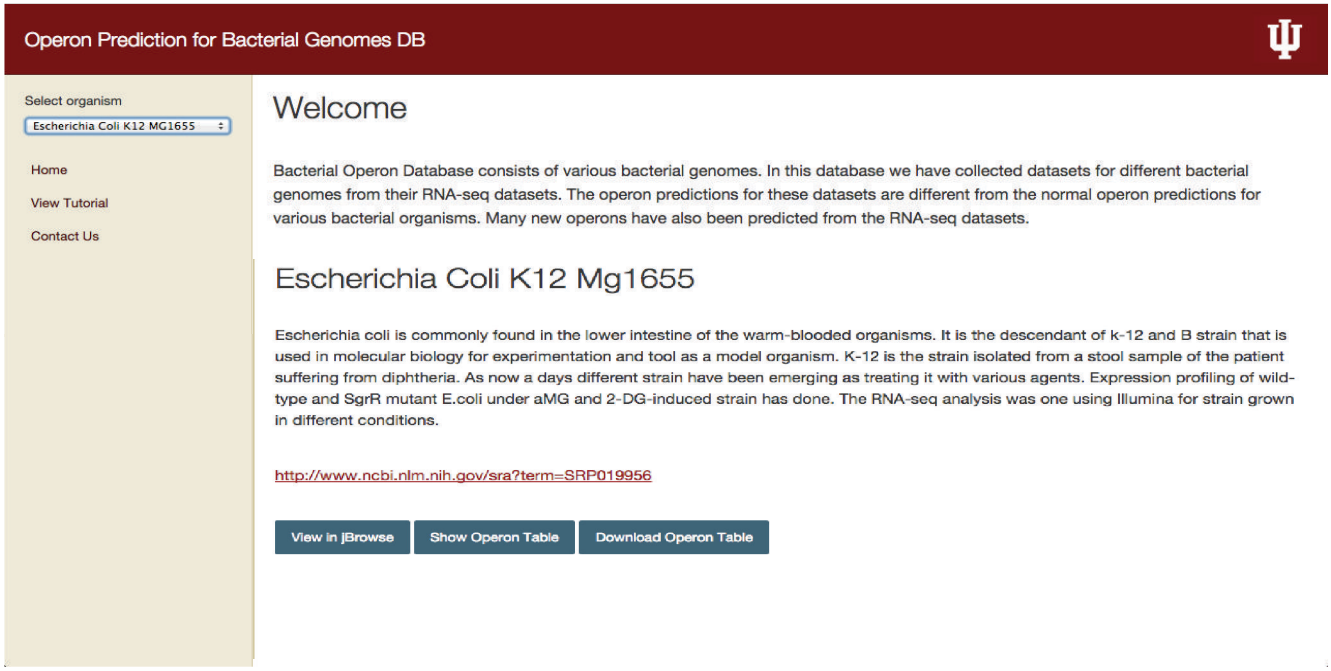


Figure 1: Web interface for operomeDB showing a screenshot of a selected bacterial genome to facilitate the browsing and download of predicted operons. The left panel of the webpage allows user to select an organism of interest. Once the user selects a bacterial organism the interface will provide information about the organism, experimental conditions under which RNA-seq datasets are available, and SRA link for experimental conditions and options to visualize in jBrowse, show operon table, and download the complete set of operon predictions across all the conditions as a table.

conditions from which RNA-seq datasets have been taken, with reference to their SRA IDs. Also, using SRA ID, user can search for a specific condition of each bacterial genome in the NCBI SRA database (http://www.ncbi.nlm.nih.gov/sra) [20].

For a selected operon or gene, jBrowse will provide detailed information such as genomic position of that particular operon, its length in base pairs (bp), and its primary attributes such as IDs, associated gene names, and source and sequence region in FASTA format (Figure 2). For a selected gene in the gene track, additional attributes such as Dbxref (reference id) and Gbkey (CDS, gene) are also displayed. Our database can generate a FASTA file containing user-specified operons and associated information and can be downloaded to the user's local computer for further analysis.

3.2. Visualization Using jBrowse. In our database we incorporated jBrowse, which supports different file formats, and in our specific implementation we use FASTA files to display the reference sequence and BED, GFF, or BAM format files for displaying the list of genes and other discrete features such as operons [32]. User can select the particular operon and selecting that particular operon can display the length of the operon and genes constituting the specific operon as well as sequence for that particular operon (Figure 3). From jBrowse panel, user can also select any number of experimental conditions for which operon predictions using RNA-seq data are available, and it will display the operons for selected location. Users have the choice to display any number of tracks and visually compare them for downstream analysis.

For instance, Figure 4 shows examples of predicted presence and absence of operons for different experimental conditions in *Escherichia coli* K-12 *MG1655* and *Campylobacter jejuni RM1221*. It was found that certain operons in microbial genomes studied here were missing for a few experimental conditions. In our database we represent this variability of tracks with respect to the reference genome. For example, in *Escherichia coli* K-12 *MG1655*, "3025" operon encoding for the genes speD (S-adenosylmethionine decarboxylase) and speE (spermidine synthase) was found to be missing in the experimental condition SRX254733 (Figure 4(a)). Such observations could be contributed due to the specific experimental condition which the experimentalists interested in the operon can explore further on a case-by-case basis. Another example shown in Figure 4(b) from the *Campylobacter jejuni RM1221* transcriptome also exhibits variability in operon organization. In this organism we found that "61693" operon (encoding for the poorly annotated ORFs CJE0054 and CJE0055) is missing in SRX155620. In our database multigene operons are predicted based on the cotranscription occurring in genes. Hence, the operons with a lack of occurrence of cotranscription would be identified as missing operons suggesting either a functional relevance of their absence or in few cases for very low abundant genes due to the lack of sequencing depth, in certain experimental conditions under study. We anticipate that, with increase in the depth and number of conditions for which RNA-seq datasets will become available, it will become easy to tease functionally important condition-specific transcription units via operomeDB.

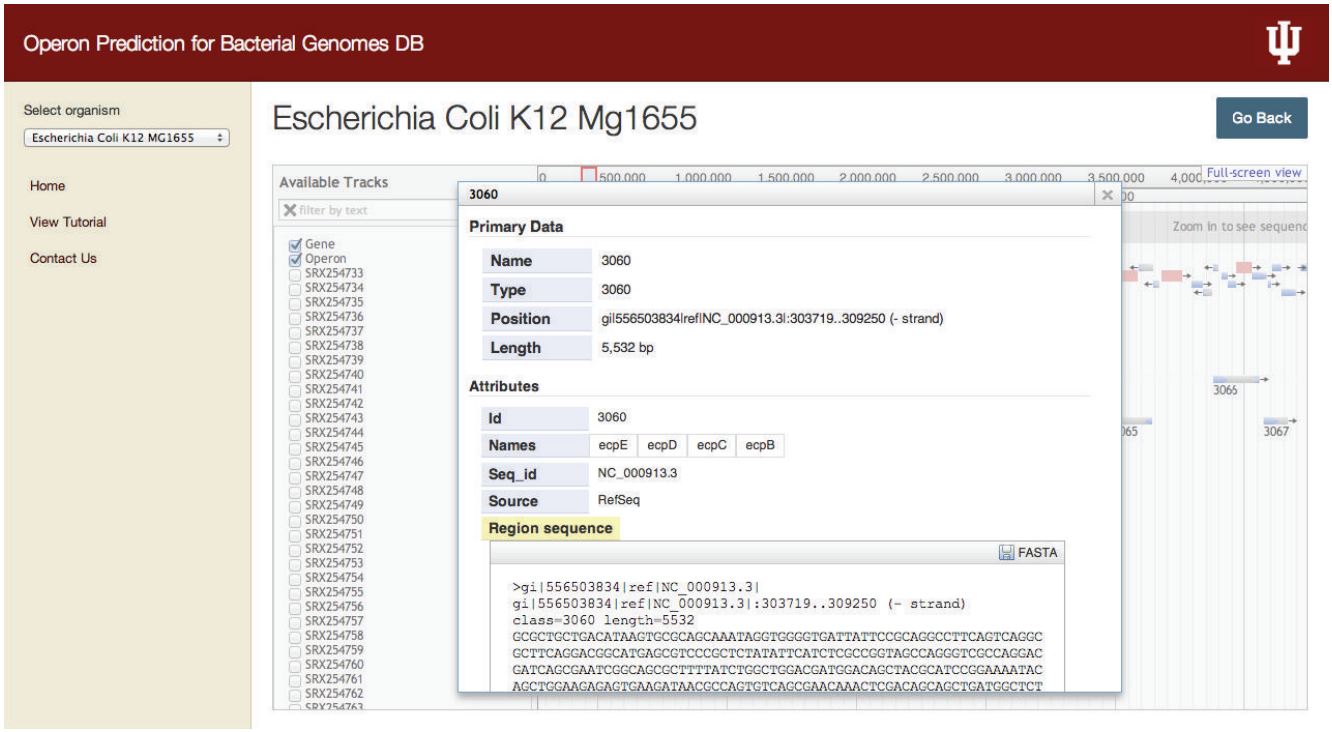


FIGURE 2: Screenshot showing the selection of an operon in the operons track. Highlighted is the ecpBCDE operon in *Escherichia coli* K-12 genome encoding for the membrane and fimbria formation proteins. This view provides the name (database generated ID), position, type, and length of the operon. It also gives information such as the number of genes present in the operon and sequence of the region for the selected operon.

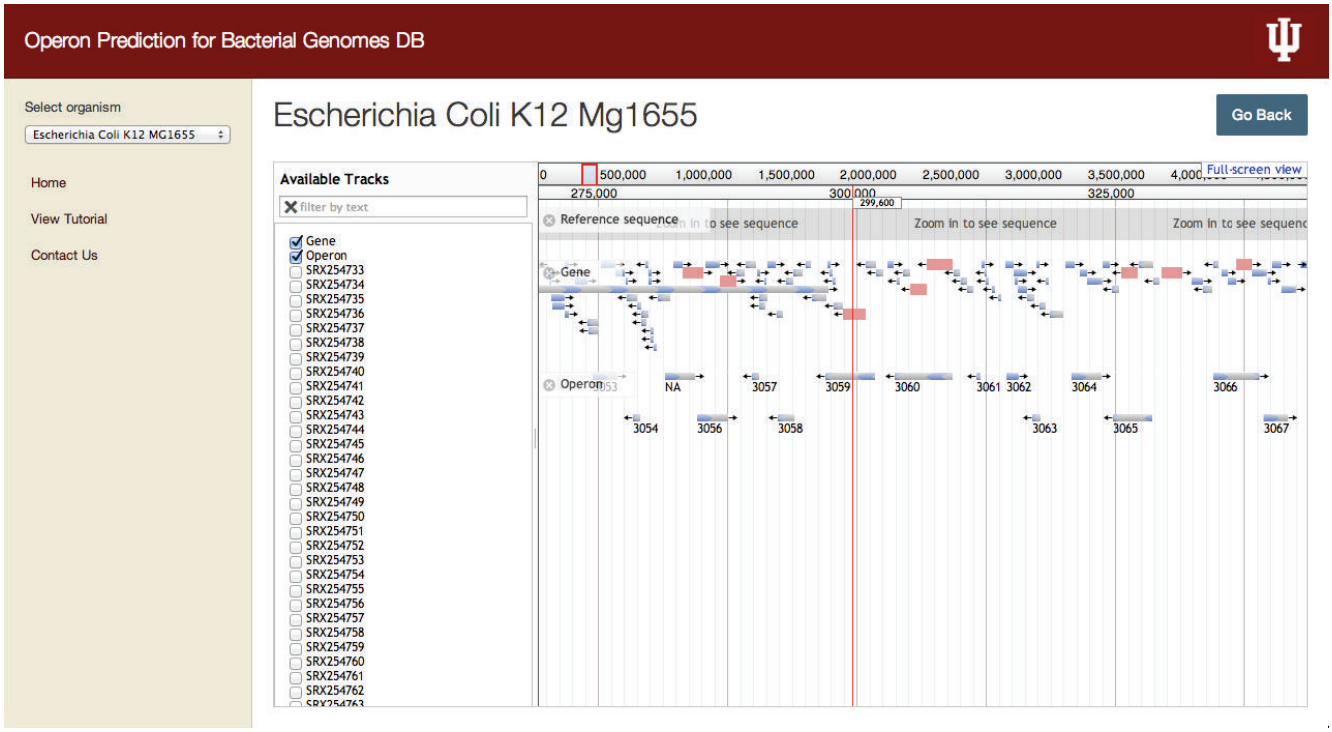
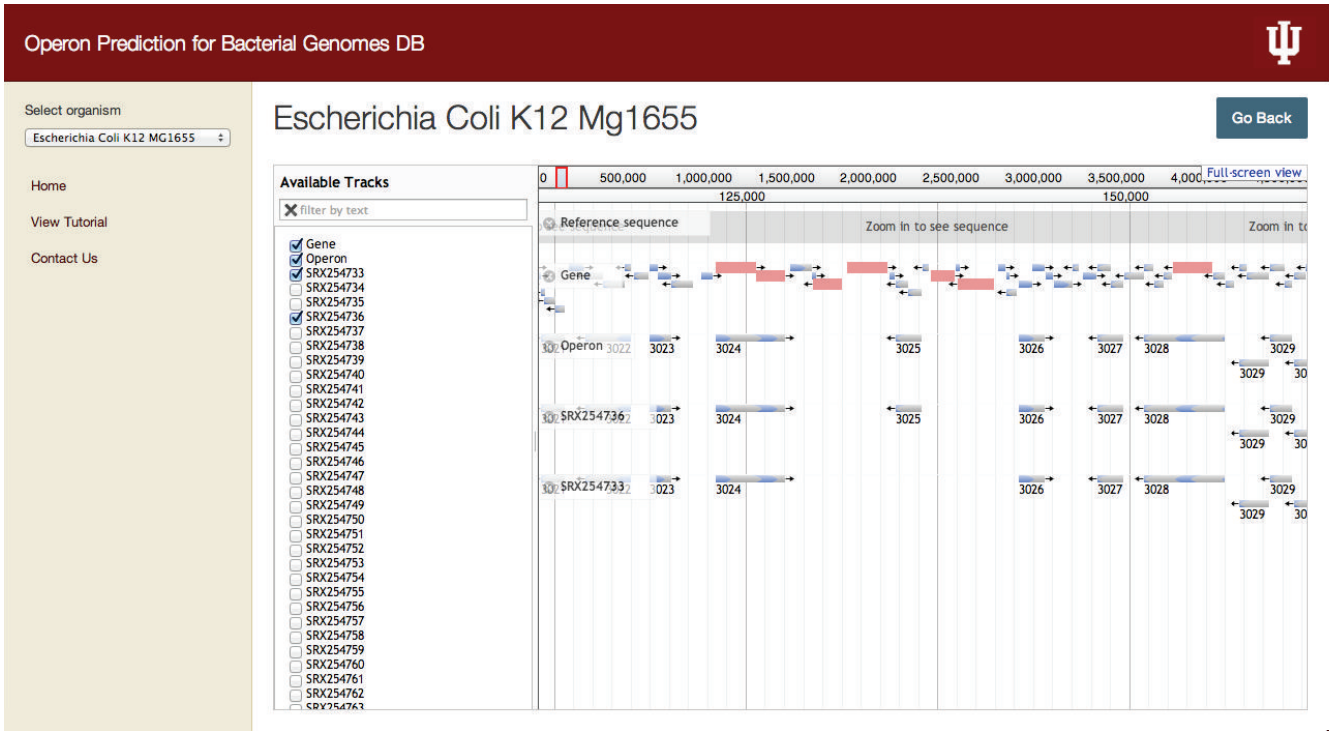
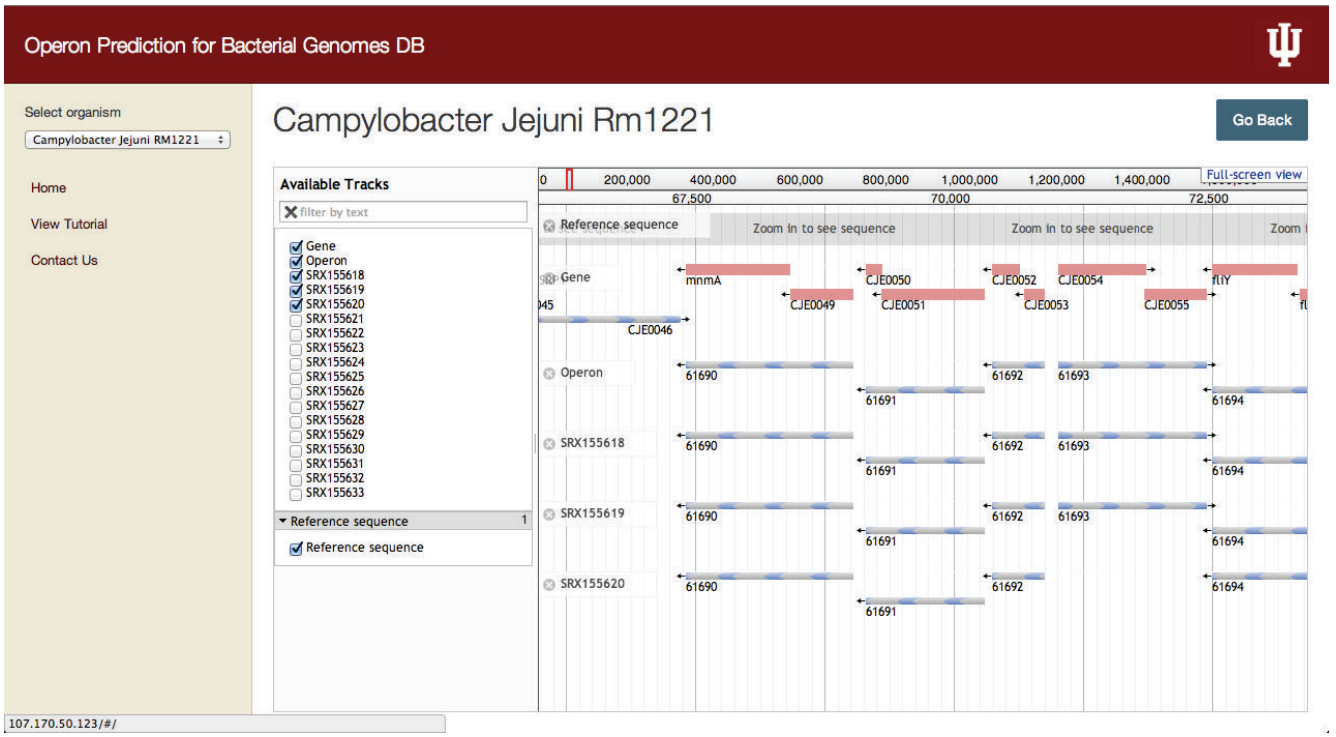


FIGURE 3: Snapshot of the jBrowse visualization showing the ensemble of all the operons predicted for a bacterial organism. User can select the reference sequence, genes present in the organism, and operons predicted from all the datasets and select the individual dataset to get the operons predicted for a particular experimental condition.



(a)



(b)

FIGURE 4: Presence and absence of operons for different experimental conditions in two different bacterial genomes. (a) For *Escherichia coli* K-12 *MG1655* we have displayed the information for the missing operon "3025" in one of the experimental conditions SRX254733. (b) Another example is from *Campylobacter jejuni Rm1221* where we have displayed the information for the condition SRX155620 with missing operon "61693."

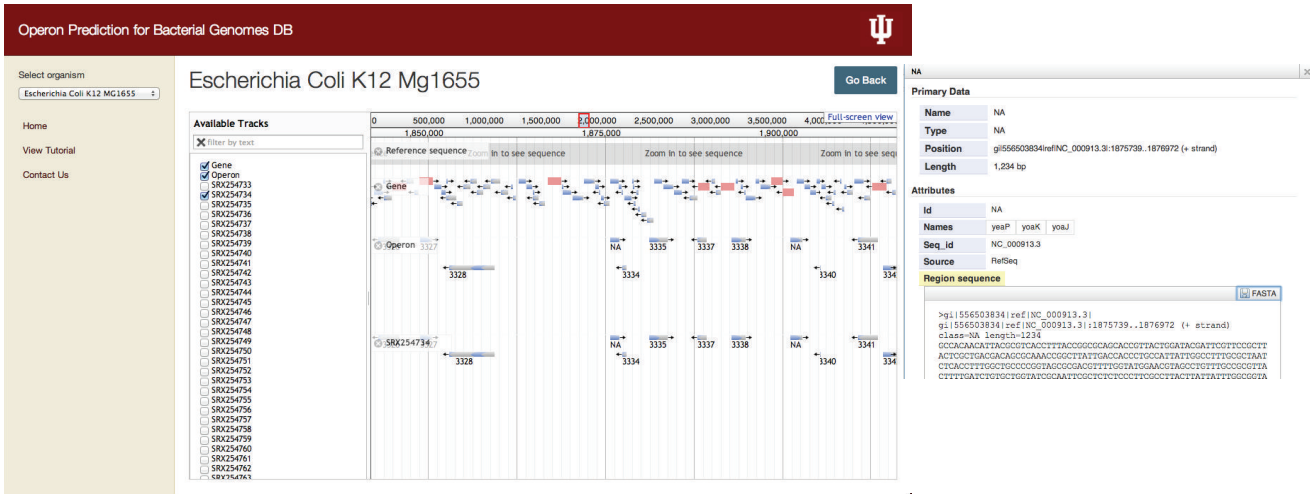


FIGURE 5: Operonic view showing a newly identified operon (yeaP-yoaK-yoaJ) in the genome of *Escherichia coli* K-12. In operomeDB, newly identified operons compared to other databases such as DOOR are marked as "NA" and user can further click on these to get the relevant information.

Our system also allows a user to submit their own sequence in specific file format and database will display its contents as an additional track. Using options in jBrowse, user can easily upload their data files to jBrowse or paste URLs, where data is present to display its contents. Various file formats such as GFF3, BigWig, BAM index, BAM, and VCF are supported. User can also visualize and compare different tracks and hence analyze if there are similarities/dissimilarities between tracks. This feature will enable the comparison of new RNA-seq data for a given organism with already available public data for various experimental conditions available in operomeDB. Additionally, custom tracks will also enable comparison of operon tracks of different closely related organisms to study the variations in transcript architecture across the length of the genome.

In comparison to earlier resources of bacterial operons, our database offers high quality single-nucleotide resolution bacterial operon predictions based on high-throughput data sets.

4. Using the Database: An Example

Below we provide an example illustrating the functionality of operomeDB. The presented example (Figure 5) is from *Escherichia coli* K-12 *MG1655* genome for a newly identified three-gene operon yeaP-yoaK-yoaJ which has not been annotated in other databases such as DOOR [9] highlighting novel predicted operons that can be identified and visualized using our database.

(1) A user can go to the main page of our Graphical User Interface (GUI) and click on "Select Organism" and then it will provide the list of the entire bacterial organisms present in the database. For instance, selecting the query genome as *Escherichia coli* K-12 *MG1655* will display the page showing the operon predictions in various formats for *E. coli*.

(2) On the query result page, it will display the information regarding *E. coli* and other possible options available. By clicking on the link "View in jBrowse" will enable the user to navigate the data via genome browser through different tracks.

(3) In genome browser on the left panel user can select any number of available tracks and selected tracks will be displayed in the browser window. The user can now go through each track and query different operons predicted in our database.

(4) Using the download button user can download the FASTA sequence file for a particular operon.

(5) By selecting "file" option in upper panel, users can also upload/add their own sequence or dataset for visualization or comparison in the genome browser.

(6) User can also look for predicted operons in each bacterial organism marked as "NA." These are the operons that are newly predicted in our study compared to the DOOR operon database [9] (Figure 5).

5. Conclusion and Future Directions

Characterizing operon structures in a genome is one of the first and fundamental steps towards improving our understanding on transcriptional regulation in bacterial genomes. OperomeDB represents one of the first attempts to provide a comprehensive resource for operon structures in microbial genomes based on RNA-sequencing data, providing a one stop portal for understanding the genome organization in the context of transcriptional regulation in a condition-specific manner. OperomeDB as a database should not only aid experimental groups working on transcriptome analysis of specific organisms but also enable studies related to computational and comparative operomics.

In our study each SRA ID for which the operon prediction was performed corresponds to a different condition

or perturbation to the cell in which RNA was sequenced to quantitate the expression levels of genes. Therefore, this database will be helpful for researchers not only to browse through each condition and analyze operons predicted for that particular condition but also to add their own new RNA-seq datasets corresponding to their experiments to uncover novel operon signatures specific to their condition of interest. Researchers can also compare operons predicted in our database with other databases under various conditions. Comparing operons under experimental and normal conditions will provide insight into the mechanism and effect of the particular condition on bacterial regulation at specific genomic loci. In the future, we will add more bacterial organisms with RNA-seq datasets to our database and we will also increase the number of datasets/conditions for already existing bacterial organisms in our database.

6. Availability and Requirements

OperomeDB can be accessed from the following URL: http://sysbio.informatics.iupui.edu/operomeDB/#/.

Conflict of Interests

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

References

[1] F. Jacob, D. Perrin, C. Sanchez, and J. Monod, "Operon: a group of genes with the expression coordinated by an operator," *Comptes Rendus Hebdomadaires des Séances de l'Académie des sciences*, vol. 250, pp. 1727–1729, 1960.

[2] F. Jacob, D. Perrin, C. Sánchez, J. Monod, and S. Edelstein, "The operon: a group of genes with expression coordinated by an operator. C.R.Acad. Sci. Paris 250 (1960) 1727–1729]," *Comptes Rendus Biologies*, vol. 328, no. 6, pp. 514–520, 2005.

[3] T. Dandekar, B. Snel, M. Huynen, and P. Bork, "Conservation of gene order: a fingerprint of proteins that physically interact," *Trends in Biochemical Sciences*, vol. 23, no. 9, pp. 324–328, 1998.

[4] R. Overbeek, M. Fonstein, M. D'Souza, G. D. Push, and N. Maltsev, "The use of gene clusters to infer functional coupling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 6, pp. 2896–2901, 1999.

[5] S. C. Janga, J. Collado-Vides, and G. Moreno-Hagelsieb, "Nebulon: a system for the inference of functional relationships of gene products from the rearrangement of predicted operons," *Nucleic Acids Research*, vol. 33, no. 8, pp. 2521–2530, 2005.

[6] W. C. Lathe III, B. Snel, and P. Bork, "Gene context conservation of a higher order than operons," *Trends in Biochemical Sciences*, vol. 25, no. 10, pp. 474–479, 2000.

[7] M. Güell, V. van Noort, E. Yus et al., "Transcriptome complexity in a genome-reduced bacterium," *Science*, vol. 326, no. 5957, pp. 1268–1271, 2009.

[8] R. Sorek and P. Cossart, "Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity," *Nature Reviews Genetics*, vol. 11, no. 1, pp. 9–16, 2010.

[9] F. Mao, P. Dam, J. Chou, V. Olman, and Y. Xu, "DOOR: a database for prokaryotic operons," *Nucleic Acids Research*, vol. 37, no. 1, pp. D459–D463, 2009.

[10] M. Pertea, K. Ayanbule, M. Smedinghoff, and S. L. Salzberg, "OperonDB: a comprehensive database of predicted operons in microbial genomes," *Nucleic Acids Research*, vol. 37, no. 1, pp. D479–D482, 2009.

[11] B. Taboada, R. Ciria, C. E. Martinez-Guerrero, and E. Merino, "ProOpDB: prokaryotic operon database," *Nucleic Acids Research*, vol. 40, no. 1, pp. D627–D631, 2012.

[12] D. Chivian, P. S. Dehal, K. Keller, and A. P. Arkin, "MetaMicrobesOnline: phylogenomic analysis of microbial communities," *Nucleic Acids Research*, vol. 41, no. 1, pp. D648–D654, 2013.

[13] H. Salgado, M. Peralta-Gil, S. Gama-Castro et al., "RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more," *Nucleic Acids Research*, vol. 41, no. 1, pp. D203–D213, 2013.

[14] M. D. Ermolaeva, O. White, and S. L. Salzberg, "Prediction of operons in microbial genomes," *Nucleic Acids Research*, vol. 29, no. 5, pp. 1216–1221, 2001.

[15] R. McClure, D. Balasubramanian, Y. Sun et al., "Computational analysis of bacterial RNA-Seq data," *Nucleic Acids Research*, vol. 41, no. 14, article e140, 2013.

[16] H. Thorvaldsdóttir, J. T. Robinson, and J. P. Mesirov, "Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration," *Briefings in Bioinformatics*, vol. 14, no. 2, pp. 178–192, 2013.

[17] M. Goldman, B. Craft, T. Swatloski et al., "The UCSC cancer genomics browser: update 2013," *Nucleic Acids Research*, vol. 41, no. 1, pp. D949–D954, 2013.

[18] P. Flicek, M. R. Amode, D. Barrell et al., "Ensembl 2014," *Nucleic Acids Research*, vol. 42, no. 1, pp. D749–D755, 2014.

[19] M. E. Skinner, A. V. Uzilov, L. D. Stein, C. J. Mungall, and I. H. Holmes, "JBrowse: a next-generation genome browser," *Genome Research*, vol. 19, no. 9, pp. 1630–1638, 2009.

[20] Y. Kodama, M. Shumway, and R. Leinonen, "The sequence read archive: explosive growth of sequencing data," *Nucleic Acids Research*, vol. 40, no. 1, pp. D54–D56, 2012.

[21] P. Stothard, G. van Domselaar, S. Shrivastava et al., "BacMap: an interactive picture atlas of annotated bacterial genomes," *Nucleic Acids Research*, vol. 33, no. supplement 1, pp. D317–D320, 2005.

[22] C. S. Wadler and C. K. Vanderpool, "Characterization of homologs of the small rna sgrs reveals diversity in function," *Nucleic Acids Research*, vol. 37, no. 16, pp. 5477–5485, 2009.

[23] H. J. Haiser, D. B. Gootenberg, K. Chatman, G. Sirasani, E. P. Balskus, and P. J. Turnbaugh, "Predicting and manipulating cardiac drug inactivation by the human gut bacterium *Eggerthella lenta*," *Science*, vol. 341, no. 6143, pp. 295–298, 2013.

[24] G. Dugar, A. Herbig, K. U. Förstner et al., "High-resolution transcriptome maps reveal strain-specific regulatory features of multiple *Campylobacter jejuni* isolates," *PLoS Genetics*, vol. 9, no. 5, Article ID e1003495, 2013.

[25] Y. Wang, X. Li, Y. Mao, and H. P. Blaschek, "Single-nucleotide resolution analysis of the transcriptome structure of *Clostridium beijerinckii* NCIMB 8052 using RNA-Seq," *BMC Genomics*, vol. 12, article 479, 2011.

[26] M. Sebaihia, B. W. Wren, P. Mullany et al., "The multidrug-resistant human pathogen *Clostridium difficile* has a highly mobile, mosaic genome," *Nature Genetics*, vol. 38, no. 7, pp. 779–786, 2006.

[27] K. A. Fimlaid, J. P. Bond, K. C. Schutz et al., "lobal analysis of the sporulation pathway of *Clostridium difficile*," *PLoS Genetics*, vol. 9, no. 8, Article ID e1003660, 2013.

[28] S. Uplekar, J. Rougemont, S. T. Cole, and C. Sala, "High-resolution transcriptome and genome-wide dynamics of RNA polymerase and NusA in *Mycobacterium tuberculosis*," *Nucleic Acids Research*, vol. 41, no. 2, pp. 961–977, 2013.

[29] A. M. Stringer, S. Currenti, R. P. Bonocora et al., "Genome-scale analyses of *Escherichia coli* and *Salmonella enterica* AraC reveal noncanonical targets and an expanded core regulon," *Journal of Bacteriology*, vol. 196, no. 3, pp. 660–671, 2014.

[30] E. Sallet, B. Roux, L. Sauviac et al., "Next-generation annotation of prokaryotic genomes with EuGene-P: application to sinorhizobium meliloti 2011," *DNA Research*, vol. 20, no. 4, pp. 339–353, 2013.

[31] A. M. Ruffing, "RNA-Seq analysis and targeted mutagenesis for improved free fatty acid production in an engineered cyanobacterium," *Biotechnology for Biofuels*, vol. 6, no. 1, article 113, 2013.

[32] O. Westesson, M. Skinner, and I. Holmes, "Visualizing next-generation sequencing data with JBrowse," *Briefings in Bioinformatics*, vol. 14, no. 2, pp. 172–177, 2013.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 857327, 9 pages
http://dx.doi.org/10.1155/2015/857327

Research Article

How to Choose In Vitro Systems to Predict In Vivo Drug Clearance: A System Pharmacology Perspective

Lei Wang,[1,2] ChienWei Chiang,[3,4] Hong Liang,[1,2] Hengyi Wu,[3,4] Weixing Feng,[1,5] Sara K. Quinney,[3,6] Jin Li,[1,2] and Lang Li[3,7]

[1]*Bioinformatics Research Center, College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China*
[2]*Biomedical Engineering Institute, College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China*
[3]*Center for Computational Biology and Bioinformatics, School of Medicine, Indiana University, Indianapolis, IN 46202, USA*
[4]*School of Informatics and Computing, Indiana University, Indianapolis, IN 46202, USA*
[5]*Pattern Recognition and Intelligent System Institute, College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China*
[6]*Department of Obstetrics and Gynecology, School of Medicine, Indiana University, Indianapolis, IN 46202, USA*
[7]*Department of Medical and Molecular Genomics, School of Medicine, Indiana University, Indianapolis, IN 46202, USA*

Correspondence should be addressed to Jin Li; lijin@hrbeu.edu.cn

Received 13 November 2014; Revised 23 January 2015; Accepted 4 February 2015

Academic Editor: Stelvio M. Bandiera

The use of in vitro metabolism data to predict human clearance has become more significant in the current prediction of large scale drug clearance for all the drugs. The relevant information (in vitro metabolism data and in vivo human clearance values) of thirty-five drugs that satisfied the entry criteria of probe drugs was collated from the literature. Then the performance of different in vitro systems including *Escherichia coli* system, yeast system, lymphoblastoid system and baculovirus system is compared after in vitro-in vivo extrapolation. Baculovirus system, which can provide most of the data, has almost equal accuracy as the other systems in predicting clearance. And in most cases, baculovirus system has the smaller CV in scaling factors. Therefore, the baculovirus system can be recognized as the suitable system for the large scale drug clearance prediction.

1. Introduction

In vivo drug clearance is a very significant pharmacokinetic parameter, which largely determines the drug exposure in human body [1, 2]. Predicting the clinical in vivo drug clearance from the preclinical in vitro experiments is essential during the drug development. Specifically, hepatic clearance (CL_H) is the most important clearance parameter as the majority of the drugs are metabolized in human liver [3].

The most common in vitro drug clearance methods include the use of human liver microsomes (HLMs) or hepatocytes [4], which are well documented in the literature [5–8]. The advantage of HLMs and human hepatocytes is that they are physiologically closer to human liver [4, 9, 10]. Their disadvantages include enormous problems between sample variations with unknown causes and relative high expense [11, 12]. In particular, the large variation of in vitro experiments in HLMs or hepatocytes causes the doubts in reproducibility. On the other hand, the commercial availability of recombinant human enzyme expression systems makes the prediction of human drug clearance cheaper and more reproducible [13, 14]. The advantages and limitations of each in vitro approach are well documented [15–21].

In order to predict in vivo clearance from in vitro experiments, system pharmacology model, such as the physiologically based pharmacokinetic (PBPK) model have been developed rapidly [22–25]. Yap et al. [26] present statistical learning models based on mixed physicochemical and topological descriptors. Demir-Kavuk et al. [27] develop a single application called DemQSAR. Simcyp [28] and Gastroplus [29] are developed originally in collaboration with major

pharmaceutical companies to simulate and predict drug clearance and drug interaction in virtual patient populations.

Despite the previously described advances in both technology and system pharmacology modeling, in vitro drug clearance prediction still faces some new challenges [25, 30]. There are a number of in vitro recombinant enzyme systems available, but it is not clear whether they all perform similarly or differently. The performance of different recombinant systems can also be enzyme dependent, but little was known about it [30]. Finally and most importantly, unlike traditional physiologically based pharmacokinetics modeling that investigates one or a few drugs a time, current translational bioinformatics desires a system model that can conduct large scale drug clearance for all the drugs [31]. This is a new challenge that poses not only the accuracy of the in vitro-in vivo clearance prediction, but also the completeness and variations of the annotated in vitro recombinant experiment data on drug metabolisms. As the in vitro-in vivo clearance prediction has been well documented in the literature, this paper will address the completeness and variations of various in vitro recombinant experiments.

2. Methods

2.1. In Vitro Experimental Data Collection. All the in vitro data of selected drugs were collated from the published literature after identifying sources using PubMed. The criteria for drug selection were that they were recognized as probes for specific cytochrome P450s (CYP) or metabolized mostly by a single CYP enzyme [49, 50]. Probe drugs refer to drugs whose plasma AUC values had been shown to increase 5-fold or higher when coadministered with a known CYP inhibitor or AUC ratio in poor metabolizers versus 1280 extensive metabolizers is greater than 5-fold [50]. These literature data including $V_{\max}$ (pmol $\min^{-1}$ pmol^{-1}CYP), K_m (pM) were obtained from various systems for heterologous expression of recombinant P450 enzymes containing bacterial expression in *Escherichia coli*, expression in yeast cells, lymphoblastoid expression systems, and baculovirus-driven expression in insect cells. Fraction unbound in plasma (fu) of drugs was also collected. If intervals of the fraction of drug unbound in plasma parameters were collected, the mean of an interval was the acceptable value.

2.2. In Vivo Data Collection. Human clearance values were taken from published original work and in part reported by Obach et al. [32]. Both intravenous data and oral data were accepted. In the case of oral clearance, the clearance was taken as a product of oral clearance and absolute bioavailability of the drug, in order to calculate drugs' intravenous clearance. The bioavailability was got through Drug Bank [39] and published original literature with a single point or the mean of an interval. At the end, only the intravenous clearance was used to assess in vitro-in vivo clearance prediction.

2.3. In Vitro-In Vivo Extrapolation. Prediction of drug hepatic clearance using in vitro recombinant P450 enzyme kinetic parameters was performed in three main steps. Initially, intrinsic clearance per unit enzyme ($\text{CL}_{\text{int,rec}}$) was calculated by the following:

$$\text{CL}_{\text{int,rec}} = \frac{v_{\max}}{k_m}. \tag{1}$$

The median $\text{CL}_{\text{int,rec}}$ value of the same recombinant P450 enzymes expression systems for each drug was taken, respectively.

After that, $\text{CL}_{\text{int,rec}}$ was converted to a whole organ intrinsic metabolic clearance (CL_{int}) using enzyme abundance, MPPGL, and the liver weight as shown in the following:

$$\text{CL}_{\text{int}} = \left(\sum_{j=1}^{m} \text{CL}_{\text{int,rec}} \cdot \text{enzyme abundance} \right) \cdot \text{MPPGL} \cdot \text{liver weight}, \tag{2}$$

where there were m CYPs with corresponding $\text{CL}_{\text{int,rec}}$ values for different pathways in each recombinant system; enzyme abundance refers to the amount (pmol P450) per milligram of microsomal protein; MPPGL means the amount (mg) of microsomal protein per gram of liver; and the liver weight means the weight (g) of human liver. Enzyme abundance, MPPGL, liver weight, and liver blood flow were generated by Simcyp with 1000 Sim-healthy volunteers (age: 20–50), female/male ratio 1, and 100% of extensive metabolizer for all major CYP enzymes [51].

At last, the value of CL_{int} was combined with binding parameters (f_{ub}) and liver blood flow (Q_{H}) to extrapolate to whole organ clearance by well-stirred model by the following:

$$\text{CL}_{\text{H}} = \frac{\text{CL}_{\text{int}} \times Q_{\text{H}} \times f_{\text{ub}}}{\text{CL}_{\text{int}} + Q_{\text{H}} \times f_{\text{ub}}}, \tag{3}$$

where f_{ub} is the fraction of drug unbound in blood. So, it could be calculated by $f_{\text{u}}/B/P$ ratios. While B/P ratios were not all available from the literature, a default value of 0.55 was used. Meanwhile, nonspecific microsomal binding was ignored.

2.4. Scaling Factor. The scaling factor of each probe drug was assessed from the difference between predicted and observed in vivo values as described in the following:

$$\text{Scaling Factor} = \log_2 \left(\frac{\text{CL}_{\text{H,in vivo}}}{\text{CL}_{\text{H,predicted}}} \right), \tag{4}$$

where $\text{CL}_{\text{H,in vivo}}$ is the observed in vivo clearance and $\text{CL}_{\text{H,predicted}}$ is the predicted value. Then, the scaling factor for different enzymes was determined by averaging scaling factor of probe drugs with the same recombinant P450 enzymes expression systems. This value also could assess the accuracy of clearance predicting. For one drug, if the scaling factor in one system was identical to the others, they had the same accuracy in predicting.

2.5. Statistical Analysis. All data were presented as mean ± S.E., unless stated otherwise. To measure the variability of prediction, the coefficient of variation (CV) was utilized. This CV measures the technical variations of in vitro metabolism experiments published from different labs.

Table 1: Drug set.

Drug	Expression systems	In vivo clearance (L/h)	References
Caffeine	Baculovirus	5.88	Obach et al. (2008) [32]
Melatonin	Baculovirus	57.96	Mallo et al. (1990) [33]
Tacrine	Yeast	235.2	Obach et al. (2008) [32]
Theophylline (1,3-DMX)	*E. coli* and lymphoblastoid	3.612	Obach et al. (2008) [32]
Bupropion	Baculovirus	5.415	Lei et al. (2010) [34], Hill et al. (2007) [35]
Efavirenz	Baculovirus	5.483	Gengiah et al. (2012) [36], Chiappetta et al. (2010) [37]
Repaglinide	Baculovirus	32.76	Obach et al. (2008) [32]
Paclitaxel	Baculovirus	26.88	Obach et al. (2008) [32]
(R)-Warfarin	Baculovirus, *E. coli*, and lymphoblastoid	0.231	Obach et al. (2008) [32]
Phenytoin	Baculovirus and lymphoblastoid	3.906	Hayes et al. (1975) [38]
Celecoxib	Lymphoblastoid and yeast	21.05	Drug Bank [39], Paulson et al. (2001) [40]
Clobazam	Baculovirus	2.49	Drug Bank [39]
(R)-Lansoprazole (dexlansoprazole)	Baculovirus	18.48	Obach et al. (2008) [32]
(R)-Omeprazole	Baculovirus and lymphoblastoid	35.28	Obach et al. (2008) [32]
Atomoxetine	Baculovirus	15.435	Drug Bank [39]
Dextromethorphan	Baculovirus, *E. coli*, yeast, and lymphoblastoid	40.59	Moghadamnia et al. (2003) [41], Kukanich and Papich (2004) [42]
Metoprolol	Lymphoblastoid	54.6	Obach et al. (2008) [32]
Perphenazine	Baculovirus	113.4	Obach et al. (2008) [32]
Tolterodine	Baculovirus	10.5	Brynne et al. (1997) [43]
Venlafaxine	Lymphoblastoid and yeast	40.95	Drug Bank [39]
Alfentanil	Baculovirus	16.38	Obach et al. (2008) [32]
Astemizole	Yeast	82.6	Lefebvre et al. (1997) [44]
Cisapride	Baculovirus	14.20	Lowry et al. (2003) [45]
Cyclosporine	Baculovirus	31.5	Obach et al. (2008) [32]
Felodipine	Baculovirus and lymphoblastoid	46.2	Obach et al. (2008) [32]
Indinavir	Baculovirus	75.6	Obach et al. (2008) [32]
Maraviroc	Baculovirus	44	Abel et al (2008) [46]
Midazolam	Baculovirus, *E. coli*, and lymphoblastoid	22.26	Obach et al. (2008) [32]
Pimozide	Baculovirus	0.042	Desta et al. (1999) [47]
Quinidine	Lymphoblastoid	16.8	Obach et al. (2008) [32]
Sildenafil	Baculovirus	38.22	Obach et al. (2008) [32]
Sirolimus	Baculovirus	2.73	Brattstram et al. (2000) [48]
Tacrolimus	Baculovirus	4.63	Obach et al. (2008) [32]
Triazolam	Baculovirus and lymphoblastoid	12.6	Obach et al. (2008) [32]
Vardenafil	Baculovirus	54.6	Obach et al. (2008) [32]

3. Results

3.1. Literature Data Collection. Thirty-five drugs were considered as probe drugs for various enzymes, CYP1A2, CYP2B6, CYP2C9, CYP2C19, CYP2D6, and CYP3A, from different expression systems as they had relatively adequate kinetic data, as shown in Table 1 [49, 50].

3.2. Comparison of Clearance Predictions for Different Enzyme Probe Drugs from the Same Expression System. Since most drugs had baculovirus system data, they were used to predict probe drugs' clearance. The predicted clearance was within 3-fold of the observed in vivo value for 6 of the 15 (40%) drugs for CYP3A probe drugs. While for CYP 2D6, none of the predicted values was within 3-fold the observed in vivo

Table 2: Predicted value and observed in vivo value of probe drugs.

Drug	Expression systems	The predicted value (L/h)	The observed in vivo value (L/h)	Scaling factor
Caffeine	Baculovirus	1.21	5.88	2.28
(R)-Lansoprazole (dexlansoprazole)	Baculovirus	81.74	18.48	−2.12
(R)-Omeprazole	Baculovirus	31.81	35.28	0.15
(R)-Omeprazole	Lymphoblastoid	15.56	35.28	1.18
(R)-Warfarin	Baculovirus	0.66	0.231	−1.51
(R)-Warfarin	*E.coli*	0.0041	0.231	5.82
(R)-Warfarin	Lymphoblastoid	0.018	0.1512	3.07
7-Epi-10-deacetyl-paclitaxel	Baculovirus	1.77	26.88	3.93
Alfentanil	Baculovirus	56.32	16.38	−1.79
Astemizole	Yeast	40.24	82.6	1.04
Atomoxetine	Baculovirus	1.65	20.99875	3.67
Bupropion	Baculovirus	7.27	5.415	−0.43
Celecoxib	Lymphoblastoid	41.26	21.05	−0.97
Celecoxib	Yeast	5.66	21.05	1.90
Cisapride	Baculovirus	20.76	14.1975	−0.56
Clobazam	Baculovirus	23.02	2.49	−3.18
Cyclosporine	Baculovirus	29.07	31.5	0.11
Dextromethorphan	Baculovirus	11.53	15.435	0.42
Dextromethorphan	*E.coli*	0.26	15.435	5.89
Dextromethorphan	Lymphoblastoid	11.87	15.435	0.38
Dextromethorphan	Yeast	21.66195	15.435	−0.49
Efavirenz	Baculovirus	0.19	5.483	4.85
Felodipine	Lymphoblastoid	0.83	46.2	5.80
Indinavir	Baculovirus	89.11	75.6	−0.23
Maraviroc	Baculovirus	33.78	44	0.38
Melatonin	Baculovirus	35.52	57.96	0.70
Metoprolol	Lymphoblastoid	44.05	54.6	0.31
Midazolam	Baculovirus	54.9	22.26	−1.29
Midazolam	*E.coli*	44.97	22.26	−1.03
Midazolam	Lymphoblastoid	46.51	22.26	−1.06
Perphenazine	Baculovirus	34.35	113.4	1.72
Phenytoin	Baculovirus	0.37	3.906	3.40
Phenytoin	Lymphoblastoid	1.07	3.906	1.87
Pimozide	Baculovirus	6.87	0.042471	−6.64
Quinidine	Lymphoblastoid	76.93	16.8	−2.18
Repaglinide	Baculovirus	27.29	32.76	0.26
Sildenafil	Baculovirus	5.17	38.22	2.89
Sirolimus	Baculovirus	19.47	2.73	−2.84
Tacrine	Yeast	16.96	235.2	3.79
Tacrolimus	Baculovirus	89.11	4.634	−4.32
Theophylline (1,3-DMX)	*E. coli*	6.6	3.612	−0.86
Theophylline (1,3-DMX)	Lymphoblastoid	8.04	3.612	−1.15
Tolterodine	baculovirus	2.22	10.5	2.24
Triazolam	Baculovirus	46.65	12.6	−1.89
Triazolam	Lymphoblastoid	4.65	12.6	1.44
Vardenafil	Baculovirus	27.75	54.6	0.98
Venlafaxine	Lymphoblastoid	13.91	40.95	1.56
Venlafaxine	Yeast	2.76	40.95	3.89

Table 3: Scaling factor with different enzymes and expression systems.

Enzymes	Expression systems	Scaling factor (mean ± SD)	CV
1A2	Baculovirus	1.493 ± 1.112	74.48%
1A2	*E. coli*	−0.869	—
1A2	Lymphoblastoid	−1.154	—
1A2	Yeast	3.794	—
2B6	Baculovirus	2.232 ± 3.756	168.28%
2C8	Baculovirus	2.093 ± 2.588	123.65%
2C9	Baculovirus	0.947 ± 3.490	368.53%
2C9	*E. coli*	5.828	—
2C9	Lymphoblastoid	1.530 ± 2.351	153.66%
2C9	Yeast	1.894	—
2C19	Baculovirus	−1.735 ± 1.716	98.90%
2C19	Lymphoblastoid	1.181	—
2D6	Baculovirus	2.394 ± 0.601	25.10%
2D6	Lymphoblastoid	1.213 ± 0.790	65.13%
2D6	*E. coli*	7.290	—
2D6	Yeast	2.399 ± 2.111	87.95%
3A	Baculovirus	−1.320 ± 2.653	200.98%
3A	*E. coli*	−1.014	—
3A	Lymphoblastoid	0.993 ± 3.541	356.60%
3A	Yeast	1.038	—

value. Only one drug was within 3-fold the observed value for CYP1A2, 2B6, 2C8, 2C9, and 2C19, which accounted for 50%, 50%, 50%, 50%, and 33% of the total. These results were illustrated in Figure 1 and Table 2.

3.3. Comparisons of Clearance Predictions in the Different Expression Systems. Dextromethorphan and midazolam were selected to compare different expression systems, because these two drugs were investigated and published under all these systems. For dextromethorphan, the predicted values from yeast system were only within 3-fold the in vivo value. And baculovirus system and lymphoblastoid system had almost the same prediction accuracy (Figure 2, Table 2).

For midazolam, all of the predicted clearance values were within 3-fold the in vivo clearance values. The most accurate predicted value was from *E. coli* system. And the three expression systems had almost the same prediction accuracy (Figure 2, Table 2).

3.4. Comparison of Data Availability from Different Expression Systems. All the in vitro recombinant enzyme expression system data were collated from the published literature. The total number of data points was 293. Figure 3 showed the proportion of data from different expression systems. In general, baculovirus and lymphoblastoid system were more abundant than the others. Baculovirus system has the largest proportion, 67%. Lymphoblastoid system was the second one, 20%. Only 8% and 5% of the data came from *E. coli* system and yeast system.

If we mapped all the data to different drugs, the majority of the drugs (28/35) were tested in the baculovirus expression system; part (12/35) of the selected drugs were test in the lymphoblastoid expression systems and only 4/35 were from *E. coli* and yeast systems, respectively.

3.5. Comparisons of Scaling Factors. Scaling factors of different enzymes based on CYP expression systems were calculated and shown in Table 3. These scaling factor ranged from −1.735 to 3.794. In the baculovirus expression system, the values of scaling factors varied a lot across the enzymes (Figure 4). And 71.4% (5/7) of the values, whose range was −1.735 to 2.394, were positive.

The variability in the same enzyme between CYP expression systems was also different. In CYP2D6, yeast system and lymphoblastoid system had higher variability than baculovirus system with the coefficient of variation (CV) 65.13%, 87.95%, and 25.10%, respectively. In CYP3A, lymphoblastoid system (CV = 356.60%) had higher variability than baculovirus system (CV = 200.98%) similarly. However, in CYP2C9 the coefficient of variation in lymphoblastoid (CV = 153.66%) was smaller than baculovirus system (CV = 368.53%).

4. Discussion and Conclusion

In this paper, we compare the performance of different recombinant human enzyme expression systems (including *Escherichia coli* system, yeast system, lymphoblastoid system, and baculovirus system) for predicting hepatic clearance in human body. And we attempt to find out the most suitable one for the large scale drug clearance prediction. After collecting the in vitro pharmacokinetic parameters of thirty-five probe drugs, we use in vitro-in vivo extrapolation to predict the clearance. The experimental results (Table 2) show

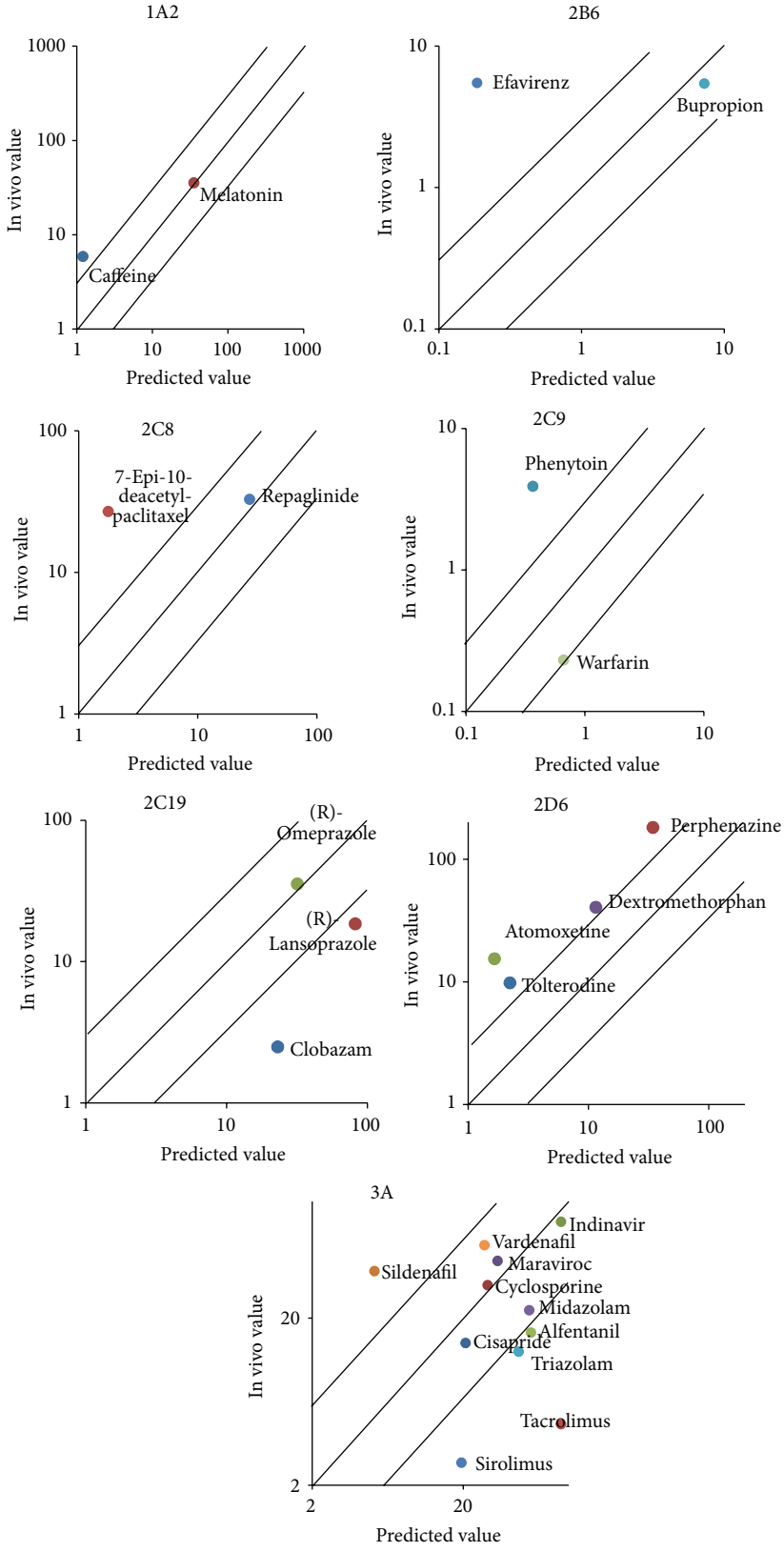


Figure 1: Predicted versus observed clearances of 28 drugs of baculovirus expression system.

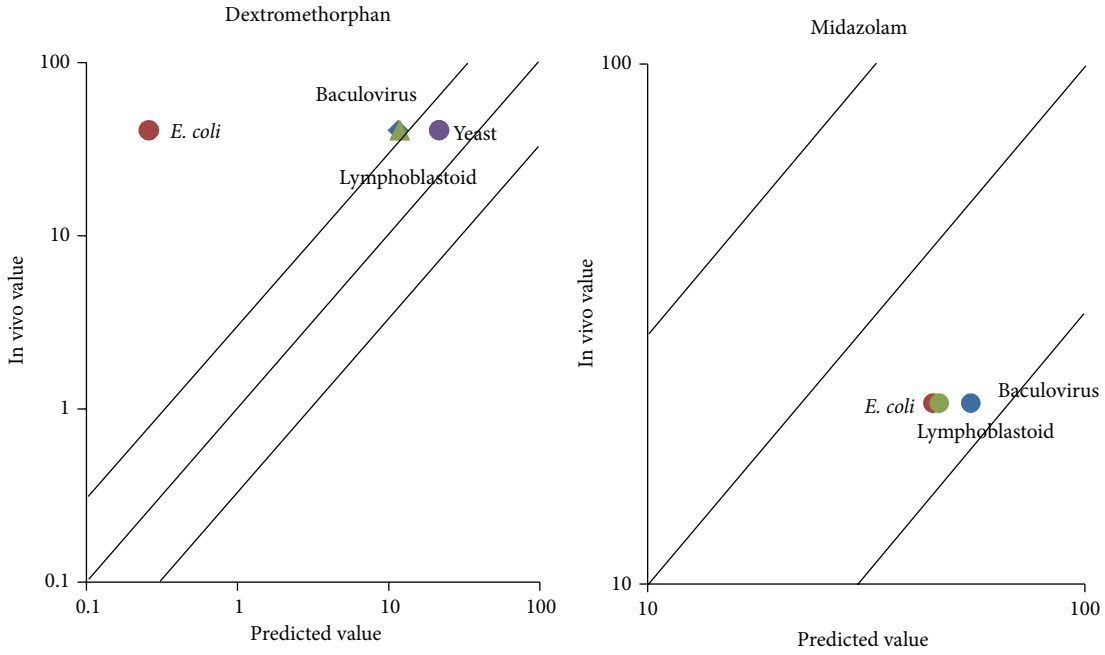


Figure 2: Predicted versus observed clearances of 2 drugs with different expression systems.

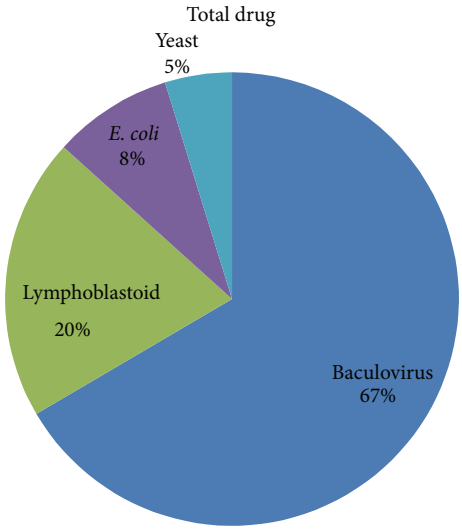


Figure 3: Proportion of data from different expression systems.

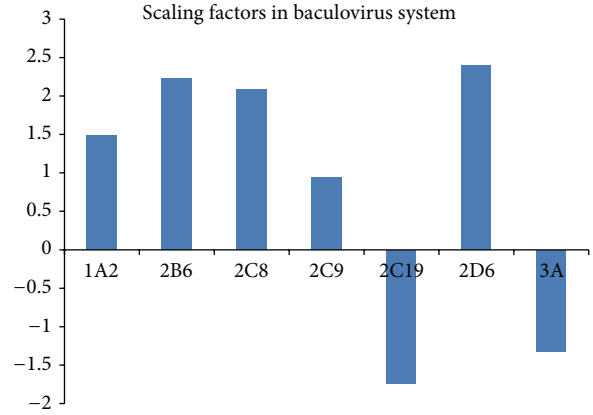


Figure 4: Comparisons of scaling factors in the baculovirus system.

that half (24/48) of the predicted values in different in vitro systems are within 3-fold the observed in vivo clearance values.

The comparisons of clearance predictions for different enzyme probe drugs from the same expression system and in different expression systems, data availability from different expression systems, and scaling factors are further analyzed. Figure 2 shows that baculovirus system has almost equal accuracy as the other systems in predicting clearance. Meanwhile, it can provide more and sufficient data for prediction than the others (Figure 3). We should note that the scaling factor will be enzyme dependent as shown in Table 3 and in most cases baculovirus systems have the smaller CV in scaling factors. Therefore, we shall use data of the baculovirus system for the large scale drug clearance prediction.

Nevertheless, there are a few more caveats. Most important of all is that in vivo clearance of some probe drugs we collected contains the renal clearance. Some of the in vivo clearance is obtained as the systemic clearance. And the proportion of hepatic metabolism was not clear. Hence, the scaling factor estimation may have some bias.

In most closely related studies, the combination of HLM and recombinant enzymes is implemented to predict in vivo clearance for high accuracy of in vitro-in vivo extrapolation [52–54]. But most of them only focused on one drug, and the choice of in vitro systems was not taken into consideration.

We are fully aware that some drugs are metabolized through non-CYP pathways, such as oxidases, reductases,

and other phase II metabolism enzymes. Our preliminary research on these enzymes revealed very limited in vitro experiment data on only a handful of drugs. Therefore, these data cannot be scaled up to do large scale in vitro-in vivo prediction and to evaluate their variations.

To our knowledge, this is the first study to compare the performance of different in vitro systems and make a decision. With the assistance of our work, the large scale drug clearance prediction should be more effective and efficient.

Conflict of Interests

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

Acknowledgments

This work was supported in part by a grant from National Natural Science Foundation of China (61071174) and Fundamental Research Funds for the Central Universities (HEUCFT1102, HEUCFT1302, and HEUCFX41303), and United States National Institute of Health Grants R01GM10448301-A1 and R01LM011945-01.

References

[1] K. Abduljalil, T. Cain, H. Humphries, and A. Rostami-Hodjegan, "Deciding on success criteria for predictability of pharmacokinetic parameters from in vitro studies: an analysis based on in vivo observations," *Drug Metabolism & Disposition*, vol. 42, no. 9, pp. 1478–1484, 2014.

[2] F. Lombardo, R. S. Obach, M. V. Varma, R. Stringer, and G. Berellini, "Clearance mechanism assignment and total clearance prediction in human based upon in silico models," *Journal of Medicinal Chemistry*, vol. 57, no. 10, pp. 4397–4405, 2014.

[3] M. Shou, "Prediction of pharmacokinetics and drug-drug interactions from in vitro metabolism data," *Current Opinion in Drug Discovery and Development*, vol. 8, no. 1, pp. 66–77, 2005.

[4] D. Zhang, G. Luo, X. Ding, and C. Lu, "Preclinical experimental models of drug metabolism and disposition in drug discovery and development," *Acta Pharmaceutica Sinica B*, vol. 2, no. 6, pp. 549–561, 2012.

[5] H. Tang and M. Mayersohn, "A novel model for prediction of human drug clearance by allometric scaling," *Drug Metabolism and Disposition*, vol. 33, no. 9, pp. 1297–1303, 2005.

[6] R. A. Stringer, C. Strain-Damerell, P. Nicklin, and J. B. Houston, "Evaluation of recombinant cytochrome p450 enzymes as an in vitro system for metabolic clearance predictions," *Drug Metabolism & Disposition*, vol. 37, no. 5, pp. 1025–1034, 2009.

[7] U. Zanelli, N. P. Caradonna, D. Hallifax, E. Turlizzi, and J. B. Houston, "Comparison of cryopreserved HepaRG cells with cryopreserved human hepatocytes for prediction of clearance for 26 drugs," *Drug Metabolism and Disposition*, vol. 40, no. 1, pp. 104–110, 2012.

[8] K. Ito and J. B. Houston, "Prediction of human drug clearance from in vitro and preclinical data using physiologically based and empirical approaches," *Pharmaceutical Research*, vol. 22, no. 1, pp. 103–112, 2005.

[9] J. Sahi, S. Grepper, and C. Smith, "Hepatocytes as a tool in drug metabolism, transport and safety evaluations in drug discovery," *Current Drug Discovery Technologies*, vol. 7, no. 3, pp. 188–198, 2010.

[10] D. F. McGinnity, M. G. Soars, R. A. Urbanowicz, and R. J. Riley, "Evaluation of fresh and cryopreserved hepatocytes as in vitro drug metabolism tools for the prediction of metabolic clearance," *Drug Metabolism and Disposition*, vol. 32, no. 11, pp. 1247–1253, 2004.

[11] P. Zhao, K. L. Kunze, and C. A. Lee, "Evaluation of time-dependent inactivation of CYP3A in cryopreserved human hepatocytes," *Drug Metabolism & Disposition*, vol. 33, no. 6, pp. 853–861, 2005.

[12] N. Hariparsad, R. S. Sane, S. C. Strom, and P. B. Desai, "In vitro methods in human drug biotransformation research: implications for cancer chemotherapy," *Toxicology in Vitro*, vol. 20, no. 2, pp. 135–153, 2006.

[13] X. Wu, J. Wang, L. Tan et al., "*In Vitro* ADME profiling using high-throughput rapidfire mass spectrometry: cytochrome P450 inhibition and metabolic stability assays," *Journal of Biomolecular Screening*, vol. 17, no. 6, pp. 761–772, 2012.

[14] O. V. Trubetskoy, J. R. Gibson, and B. D. Marks, "Highly miniaturized formats for in vitro drug metabolism assays using vivid fluorescent substrates and recombinant human cytochrome P450 enzymes," *Journal of Biomolecular Screening*, vol. 10, no. 1, pp. 56–66, 2005.

[15] N. J. Hewitt, M. J. G. Lechón, J. B. Houston et al., "Primary hepatocytes: current understanding of the regulation of metabolic enzymes and transporter proteins, and pharmaceutical practice for the use of hepatocytes in metabolism, enzyme induction, transporter, clearance, and hepatotoxicity studies," *Drug Metabolism Reviews*, vol. 39, no. 1, pp. 159–234, 2007.

[16] P. Chao, A. S. Uss, and K. Cheng, "Use of intrinsic clearance for prediction of human hepatic clearance," *Expert Opinion on Drug Metabolism and Toxicology*, vol. 6, no. 2, pp. 189–198, 2010.

[17] M. Chiba, Y. Ishii, and Y. Sugiyama, "Prediction of hepatic clearance in human from *in vitro* data for successful drug development," *The AAPS Journal*, vol. 11, no. 2, pp. 262–276, 2009.

[18] R. Niro, J. P. Byers, R. L. Fournier, and K. Bachmann, "Application of a convective-dispersion model to predict in vivo hepatic clearance from in vitro measurements utilizing cryopreserved human hepatocytes," *Current Drug Metabolism*, vol. 4, no. 5, pp. 357–369, 2003.

[19] Y. Naritomi, S. Terashita, A. Kagayama, and Y. Sugiyama, "Utility of hepatocytes in predicting drug metabolism: comparison of hepatic intrinsic clearance in rats and humans *in vivo* and *in vitro*," *Drug Metabolism & Disposition*, vol. 31, no. 5, pp. 580–588, 2003.

[20] R. A. Stringer, C. Strain-Damerell, P. Nicklin, and J. B. Houston, "Evaluation of recombinant cytochrome p450 enzymes as an in vitro system for metabolic clearance predictions," *Drug Metabolism and Disposition*, vol. 37, no. 5, pp. 1025–1034, 2009.

[21] Y. Chen, L. Liu, K. Nguyen, and A. J. Fretland, "Utility of intersystem extrapolation factors in early reaction phenotyping and the quantitative extrapolation of human liver microsomal intrinsic clearance using recombinant cytochromes P450," *Drug Metabolism and Disposition*, vol. 39, no. 3, pp. 373–382, 2011.

[22] M. Rowland, C. Peck, and G. Tucker, "Physiologically-based pharmacokinetics in drug development and regulatory science," *Annual Review of Pharmacology and Toxicology*, vol. 51, pp. 45–73, 2011.

[23] L. E. Gerlowski and R. K. Jain, "Physiologically based pharmacokinetic modeling: principles and applications," *Journal of Pharmaceutical Sciences*, vol. 72, no. 10, pp. 1103–1127, 1983.

[24] A. Rostami-Hodjegan, "Physiologically based pharmacokinetics joined with in vitro-in vivo extrapolation of ADME: a marriage under the arch of systems pharmacology," *Clinical Pharmacology and Therapeutics*, vol. 92, no. 1, pp. 50–61, 2012.

[25] A. Rostami-Hodjegan and G. T. Tucker, "Simulation and prediction of in vivo drug metabolism in human populations from in vitro data," *Nature Reviews Drug Discovery*, vol. 6, no. 2, pp. 140–148, 2007.

[26] C. W. Yap, Y. Xue, H. Li et al., "Prediction of compounds with specific pharmacodynamic, pharmacokinetic or toxicological property by statistical learning methods," *Mini-Reviews in Medicinal Chemistry*, vol. 6, no. 4, pp. 449–459, 2006.

[27] O. Demir-Kavuk, J. Bentzien, I. Muegge, and E.-W. Knapp, "DemQSAR: predicting human volume of distribution and clearance of drugs," *Journal of Computer-Aided Molecular Design*, vol. 25, no. 12, pp. 1121–1133, 2011.

[28] Simcyp, http://www.simcyp.com/.

[29] Gastroplus, http://www.simulations-plus.com/.

[30] O. Pelkonen, M. Turpeinen, J. Uusitalo, A. Rautio, and H. Raunio, "Prediction of drug metabolism and interactions on the basis of in vitro investigations," *Basic and Clinical Pharmacology and Toxicology*, vol. 96, no. 3, pp. 167–175, 2005.

[31] N. S. Buchan, D. K. Rajpal, Y. Webster et al., "The role of translational bioinformatics in drug discovery," *Drug Discovery Today*, vol. 16, no. 9-10, pp. 426–434, 2011.

[32] R. S. Obach, F. Lombardo, and N. J. Waters, "Trend analysis of a database of intravenous pharmacokinetic parameters in humans for 670 drug compounds," *Drug Metabolism & Disposition*, vol. 36, no. 7, pp. 1385–1405, 2008.

[33] C. Mallo, R. Zaidan, G. Galy et al., "Pharmacokinetics of melatonin in man after intravenous infusion and bolus injection," *European Journal of Clinical Pharmacology*, vol. 38, no. 3, pp. 297–301, 1990.

[34] H. P. Lei, X. Y. Yu, H. T. Xie et al., "Effect of St. John's wort supplementation on the pharmacokinetics of bupropion in healthy male Chinese volunteers," *Xenobiotica*, vol. 40, no. 4, pp. 275–281, 2010.

[35] S. Hill, H. Sikand, and J. Lee, "A case report of seizure induced by bupropion nasal insufflation," *Primary Care Companion to the Journal of Clinical Psychiatry*, vol. 9, no. 1, pp. 67–69, 2007.

[36] T. N. Gengiah, N. H. G. Holford, J. H. Botha, A. L. Gray, K. Naidoo, and S. S. A. Karim, "The influence of tuberculosis treatment on efavirenz clearance in patients co-infected with HIV and tuberculosis," *European Journal of Clinical Pharmacology*, vol. 68, no. 5, pp. 689–695, 2012.

[37] D. A. Chiappetta, C. Hocht, C. Taira, and A. Sosnik, "Oral pharmacokinetics of the anti-HIV efavirenz encapsulated within polymeric micelles," *Biomaterials*, vol. 32, no. 9, pp. 2379–2387, 2011.

[38] M. J. Hayes, M. J. S. Langman, and A. H. Short, "Changes in drug metabolism with increasing age: 2. Phenytoin clearance and protein binding," *British Journal of Clinical Pharmacology*, vol. 2, no. 1, pp. 73–79, 1975.

[39] Drug Bank, http://www.drugbank.ca.

[40] S. K. Paulson, M. B. Vaughn, S. M. Jessen et al., "Pharmacokinetics of celecoxib after oral administration in dogs and humans: effect of food and site of absorption," *Journal of Pharmacology and Experimental Therapeutics*, vol. 297, no. 2, pp. 638–645, 2001.

[41] A. A. Moghadamnia, A. Rostami-Hodjegan, R. Abdul-Manap, C. E. Wright, A. H. Morice, and G. T. Tucker, "Physiologically based modelling of inhibition of metabolism and assessment of the relative potency of drug and metabolite: dextromethorphan vs. dextrorphan using quinidine inhibition," *British Journal of Clinical Pharmacology*, vol. 56, no. 1, pp. 57–67, 2003.

[42] B. KuKanich and M. G. Papich, "Plasma profile and pharmacokinetics of dextromethorphan after intravenous and oral administration in healthy dogs," *Journal of Veterinary Pharmacology and Therapeutics*, vol. 27, no. 5, pp. 337–341, 2004.

[43] N. Brynne, M. M. Stahl, B. Hallen et al., "Pharmacokinetics and pharmacodynamics of tolterodine in man: a new drug for the treatment of urinary bladder overactivity," *International Journal of Clinical Pharmacology and Therapeutics*, vol. 35, no. 7, pp. 287–295, 1997.

[44] R. A. Lefebvre, A. Van Peer, and R. Woestenborghs, "Influence of itraconazole on the pharmacokinetics and electrocardiographic effects of astemizole," *British Journal of Clinical Pharmacology*, vol. 43, no. 3, pp. 319–322, 1997.

[45] J. A. Lowry, G. L. Kearns, S. M. Abdel-Rahman et al., "Cisapride: a potential model substrate to assess cytochrome P4503A4 activity in vivo," *Clinical Pharmacology & Therapeutics*, vol. 73, no. 3, pp. 209–222, 2003.

[46] S. Abel, D. Russell, L. A. Whitlock, C. E. Ridgway, A. N. Nedderman, and D. K. Walker, "Assessment of the absorption, metabolism and absolute bioavailability of maraviroc in healthy male subjects," *British Journal of Clinical Pharmacology*, vol. 65, supplement 1, pp. 60–67, 2008.

[47] Z. Desta, T. Kerbusch, and D. A. Flockhart, "Effect of clarithromycin on the pharmacokinetics and pharmacodynamics of pimozide in healthy poor and extensive metabolizers of cytochrome P450 2D6 (CYP2D6)," *Clinical Pharmacology & Therapeutics*, vol. 65, no. 1, pp. 10–20, 1999.

[48] C. Brattstram, J. Salve, B. Jansson et al., "Pharmacokinetics and safety of single oral doses of sirolimus (rapamycin) in healthy male volunteers," *Therapeutic Drug Monitoring*, vol. 22, no. 5, pp. 537–544, 2000.

[49] H.-Y. Wu, S. Karnik, A. Subhadarshini et al., "An integrated pharmacokinetics ontology and corpus for text mining," *BMC Bioinformatics*, vol. 14, no. 1, article 35, 2013.

[50] FDA, *Drug Interaction Studies-Study Design, Data Analysis, Implications for Dosing, and Labeling Recommendations*, 2012.

[51] M. Jamei, S. Marciniak, K. Feng, A. Barnett, G. Tucker, and A. Rostami-Hodjegan, "The Simcyp population-based ADME simulator," *Expert Opinion on Drug Metabolism and Toxicology*, vol. 5, no. 2, pp. 211–223, 2009.

[52] N. Wattanachai, W. Tassaneeyakul, A. Rowland et al., "Effect of albumin on human liver microsomal and recombinant CYP1A2 activities: impact on in vitro-in vivo extrapolation of drug clearance," *Drug Metabolism and Disposition*, vol. 40, no. 5, pp. 982–989, 2012.

[53] H. T'jollyn, J. Snoeys, P. Colin et al., "Physiology-based IVIVE predictions of tramadol from *in vitro* metabolism data," *Pharmaceutical Research*, vol. 32, no. 1, pp. 260–274, 2015.

[54] J. O. Miners, P. I. MacKenzie, and K. M. Knights, "The prediction of drug-glucuronidation parameters in humans: UDP-glucuronosyltransferase enzyme-selective substrate and inhibitor probes for reaction phenotyping and in vitroin vivo extrapolation of drug clearance and drug-drug interaction potential," *Drug Metabolism Reviews*, vol. 42, no. 1, pp. 189–201, 2010.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 456479, 11 pages
http://dx.doi.org/10.1155/2015/456479

Research Article

A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference

Adam Cornish[1] and Chittibabu Guda[1,2,3,4,5]

[1]*Department of Genetics, Cell Biology and Anatomy, University of Nebraska Medical Center, Omaha, NE 68198, USA*
[2]*Bioinformatics and Systems Biology Core, University of Nebraska Medical Center, Omaha, NE 68198, USA*
[3]*Department of Biochemistry and Molecular Biology, University of Nebraska Medical Center, Omaha, NE 68198, USA*
[4]*Fred and Pamela Buffet Cancer Center, University of Nebraska Medical Center, Omaha, NE 68198, USA*
[5]*Eppley Institute for Research in Cancer and Allied Diseases, University of Nebraska Medical Center, Omaha, NE 68198, USA*

Correspondence should be addressed to Chittibabu Guda; babu.guda@unmc.edu

Received 17 November 2014; Accepted 17 December 2014

Academic Editor: Aparup Das

High-throughput sequencing, especially of exomes, is a popular diagnostic tool, but it is difficult to determine which tools are the best at analyzing this data. In this study, we use the NIST Genome in a Bottle results as a novel resource for validation of our exome analysis pipeline. We use six different aligners and five different variant callers to determine which pipeline, of the 30 total, performs the best on a human exome that was used to help generate the list of variants detected by the Genome in a Bottle Consortium. Of these 30 pipelines, we found that Novoalign in conjunction with GATK UnifiedGenotyper exhibited the highest sensitivity while maintaining a low number of false positives for SNVs. However, it is apparent that indels are still difficult for any pipeline to handle with none of the tools achieving an average sensitivity higher than 33% or a Positive Predictive Value (PPV) higher than 53%. Lastly, as expected, it was found that aligners can play as vital a role in variant detection as variant callers themselves.

1. Background

In the past few years there have been many advances made to high-throughput sequencing technologies. Due to these advances, it is now possible to detect a great number of potential disease-causing variants [1], and, in a few cases, next generation sequencing (NGS) data has even been used for diagnostic purposes [2–4]. This is partially due to the developments in sequencing technologies over the past few years but also due to the number of improvements made to the various bioinformatic tools used to analyze the mountains of data produced by NGS instruments [5].

When searching for mutations in a patient, a typical workflow is to sequence their exome with an Illumina sequencer, align the raw data to the human reference genome, and then identify single nucleotide variants (SNVs) or short insertions and deletions (indels) that could possibly cause or influence the phenotype of interest [6]. While this is fairly straightforward, deciding on the best tools to use at each stage of the analysis pipeline is not. There are a large number of tools that are used in various intermediate steps, but the two most important steps in the entire process are aligning the raw reads to the genome and then searching for variants (i.e., SNVs and indels) [7]. In this study, we aim to help today's bioinformatician by elucidating the correct combination of short read alignment tool and variant calling tool for processing exome sequencing data produced by NGS instruments.

A number of these studies have been performed in the past, but they all had drawbacks of some form or another. Ideally one should have a list of every known variant contained in a sample so that when a pipeline of analysis tools is run, you can test it to know with certainty that it is performing correctly. However, in the past no such list existed, so validation had to be performed by less complete methods. In some instances, validation was performed by

generating simulated data so as to create a set of known true positives (TP) and true negatives (TN) [8–10]. While this conveniently provides a list of every TP and TN in the dataset, it does a poor job of accurately representing biology. Other methods of validating variant calling pipelines include using genotyping arrays or Sanger sequencing to obtain a list of TPs and false positives (FP) [11]. These have the upside of providing biologically validated results, but they also have the downside of not being comprehensive due to the limited number of spots on genotyping arrays and the prohibitive cost of Sanger validation when performed thousands of times. Lastly, none of these studies aimed at looking at the effect the short read aligner had on variant calling. Consequently, the upstream effect of aligner performance could not be assessed independently.

In this study, we have the advantage of a list of variants for an anonymous female from Utah (subject ID: NA12878, originally sequenced for the 1000 Genomes project [12]) that was experimentally validated by the NIST-led Genome in a Bottle (GiaB) Consortium. This list of variants was created by integrating 14 different datasets from five different sequencers, and it allows us to validate any list of variants generated by our exome analysis pipelines [7]. The novelty of this work is to validate the right combination of aligners and variant callers against a comprehensive and experimentally determined variant dataset: NIST-GiaB.

To perform our analysis we will be using one of the exome datasets originally used to create the NIST-GiaB list. We chose only one of the original Illumina TruSeq-generated exomes because we wanted to provide a standard use case scenario for someone who wishes to perform NGS analysis, and while whole genome sequencing is continuing to drop in price, exome sequencing is still a popular and viable alternative [1]. It is also important to note that, per Bamshad et al., currently the expected number of SNVs per European-American exome is 20,283 ± 523 [13]. Despite this, the total number of SNVs found in the NIST-GiaB list with the potential to exist in TruSeq exome dataset was 34,886, which is significantly higher than expected. This is likely due to the fact that while the exome kit was used to generate NIST-GiaB data it was also supplemented by whole genome sequencing.

Lastly, we considered a large number of aligners [14–21] and variant callers [22–29] but ultimately chose the 11 tools based on prevalence, popularity, and relevancy to our dataset (e.g., SNVMix, VarScan2, and MuTect were not used as they are intended for use on tumor-derived samples). Our analysis itself involves comparing six aligners (Bowtie2 [14], BWA sampe [15], BWA mem [16], CUSHAW3 [17], MOSAIK [18], and Novoalign) and five variant callers (FreeBayes [22], GATK HaplotypeCaller, GATK UnifiedGenotyper [23], SAMtools mpileup [24], and SNPSVM [25]). In this study we also try to determine how much of an effect, if any, the aligner has on variant calling and which aligners perform best when using a normal Illumina exome sample. To our knowledge, this is the first report which validates all possible combinations (total of 30 pipelines) of a wide array of aligners and variant callers.

Table 1: Alignment percentages for filtered reads and unfiltered reads. The average depth of coverage is for the alignment files created with the filtered reads.

Aligner	% reads aligned (unfiltered)	% reads aligned (filtered)	Average depth of coverage
Bowtie2	89.73	98.73	47.97
BWA mem	92.91	99.85	46.89
BWA sampe	85.95	97.49	46.67
CUSHAW3	85.00	99.81	47.69
MOSAIK	85.68	96.22	45.14
Novoalign	82.21	94.20	45.62

2. Methods

2.1. Datasets. Human reference genome hg19 was downloaded from the UCSC browser (http://hgdownload.soe.ucsc.edu/goldenPath/hg19/chromosomes/) and was used to perform the alignments. The human exome, SRR098401, was downloaded from the Sequence Read Archive (SRA) (http://www.ncbi.nlm.nih.gov/sra). For annotation and calibration purposes, dbSNP137 without sites after version 129, HapMap 3.3, Human Omni 2.5 BeadChip, and Mills and 1000 G gold standard indel set lists were used (all from ftp://ftp.broadinstitute.org/distribution/gsa/gatk_resources.tgz).

2.2. The Pipeline. Figure 1 shows the workflow used in this study, which is similar to the one outlined in the Best Practices guide produced by The Broad Institute [30]. This involves a number of steps to ensure that the alignment files produced are of the highest quality as well as several more to guarantee the variants are called correctly. First, raw reads were aligned to hg19, and then PCR duplicates were removed from the alignment. Next, to help with indel identification later in the pipeline, read realignment was performed around indels. The last step of alignment processing was to perform a base quality score recalibration step, which helps to ameliorate the inherent bias and inaccuracies of scores issued by sequencers. Unfortunately, despite these steps, the alignment rate of each aligner was significantly lower than expected, so to offset this, the fastx toolkit was used to filter out low quality reads (Table 1). Low quality reads were defined as those reads that had at least half of their quality scores below 30. Following alignment processing, variant calling and variant filtering were performed.

The six tools used to generate alignments were Bowtie2, BWA mem, BWA sampe, CUSHAW3, MOSAIK, and Novoalign, and the five tools used to generate variants were FreeBayes, GATK HaplotypeCaller, GATK UnifiedGenotyper, SAMtools mpileup, and SNPSVM, as can be seen in Table 2.

2.3. Filtering. Raw data was acquired from the SRA (SRR098401), split with fastq-dump, and filtered using the fastx toolkit. Specifically, fastq-dump used the `--split_files` and `--split_spot` flags, and

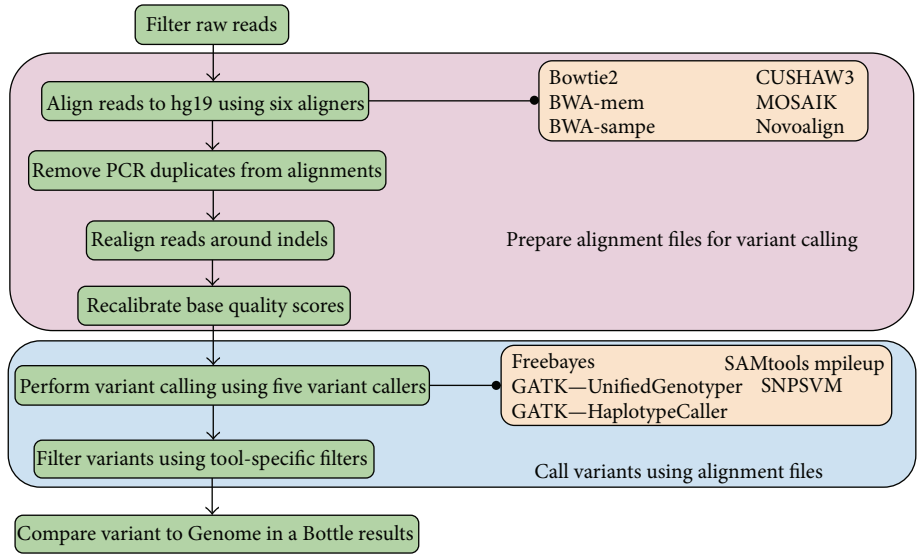


Figure 1: Schematic of the data analysis pipeline used. To ensure that the highest quality alignments are created, reads are first filtered and then aligned to the human reference genome, hg19. Next, PCR duplicates are removed, reads are aligned around putative indels, and base quality scores are recalibrated. Finally, variants are called and validated against the NIST-GiaB list of variants.

Table 2: These are the 11 different tools used that made up the 30 (six aligners $*$ five variant callers) different pipelines. Software versions are also included to ensure reproducibility.

Tool	Type	Version	Reference
Bowtie2	Aligner	2.1.0	[14]
BWA sampe	Aligner	0.7.5a	[15]
BWA mem	Aligner	0.7.5a	[16]
CUSHAW3	Aligner	3.0.3	[17]
MOSAIK	Aligner	2.2.3	[18]
Novoalign	Aligner	3.02.07	N/A
FreeBayes	Genotyper	v9.9.2-19-g011561f	[22]
GATK HaplotypeCaller	Genotyper	2.7-2	N/A
GATK UnifiedGenotyper	Genotyper	2.7-2	[23]
SAMtools mpileup	Genotyper	0.1.19	[24]
SNPSVM	Genotyper	0.01	[25]

fastq_quality_filter was run with the following arguments: -Q 33 -q 30 -p 50. Then reads were properly paired with a custom script.

2.4. Aligning. Aligners used default arguments except when a threads argument was used where available. The commands used are as follows.

2.4.1. Bowtie2

(1) `bowtie2 -p 10 -x $INDEX -1 raw_data/read_1_filtered.fastq -2 raw_data/read_2_filtered.fastq -S alignments/NA12878.bt2.sam`

2.4.2. BWA Sampe

(1) `bwa aln -t 10 genome/hg19.fa raw_data/read_1_filtered.fastq > alignments/NA12878.R1.sai`

(2) `bwa aln -t 10 genome/hg19.fa raw_data/read_2_filtered.fastq > alignments/NA12878.R2.sai`

(3) `bwa sampe genome/hg19.fa alignments/NA12878.R1.sai alignments/NA12878.R2.sai raw_data/read_1_filtered.fastq raw_data/read_2_filtered.fastq > alignments/NA12878.bwa-sampe.sam`

2.4.3. BWA Mem

(1) `bwa mem -t 10 genome/hg19.fa raw_data/read_1_filtered.fastq raw_data/read_2_filtered.fastq > alignments/NA12878.bwa-mem.sam`

2.4.4. CUSHAW3

(1) `cushaw3 align -r $INDEX -t 10 -o alignments/NA12878.CUSHAW3.sam -q raw_data/read_1_filtered.fastq raw_data/read_2_filtered.fastq`

2.4.5. MOSAIK

(1) `MosaikBuild -q raw_data/read_1_filtered.fastq -q2 raw_data/read_2_filtered.fastq -st illumina -out alignments/NA12878.MOSAIK.mkb`

(2) `MosaikAligner -in alignments/NA12878 .MOSAIK.mkb -out alignments/NA12878 .MOSAIK -p 10 -ia genome/hg19.dat -j genome/hg19_15 -annpe tools/MOSAIK/src/ networkFile/2.1.78.pe.ann -annse tools/MOSAIK/src/networkFile/ 2.1.78.se.ann`

2.4.6. Novoalign

(1) `novoalign -d $INDEX -f raw_data/ read_1_filtered.fastq raw_data/ read_2_filtered.fastq -o SAM -c 10 > alignments/NA12878.novoalign.sam`

2.5. Alignment Depth of Coverage Calculation. To ensure proper depth of coverage calculation, the Picard Tools module CalculateHsMetrics was used with the following arguments:

(1) `java -jar CalculateHsMetrics.jar I=NA12878.ALN.BQSR.bam O=ALN.O.log R=genome/hg19.fa TI=genome/ truseq_exome.bed BI=genome/ truseq_exome.bed VALIDATION_STRINGENCY= SILENT PER_TARGET_COVERAGE=ALN.ptc.bed`

It is important to note that the TruSeq exome bed file must have the header from the SAM alignment file prepended to it for this module to function. Further, column 6 must be moved to column 4, and column 5 needs to be removed from the TruSeq bed file.

2.6. Alignment File Processing. Processing the alignment files (SAM/BAM files) required the following steps for all aligners:

(1) SAM to BAM conversion with SAMtools view:

(a) `samtools view -bS alignments/ NA12878.ALN.sam -o alignments/ NA12878.ALN.bam`

(2) BAM file sorting using the Picard Tools module, SortSam:

(a) `java -jar bin/SortSam.jar VALIDATION_STRINGENCY=SILENT I=alignments/NA12878.ALN.bam OUTPUT=alignments/NA12878.ALN.sorted .bam SORT_ORDER=coordinate`

(3) PCR duplicate removal using the Picard Tools module, MarkDuplicates:

(a) `java -jar bin/MarkDuplicates.jar VALIDATION_STRINGENCY=SILENT I=alignments/NA12878.ALN.sorted.bam O=alignments/NA12878.ALN.dups_ removed.bam REMOVE_DUPLICATES= true M=alignments/metrics`

(4) Read Group added to alignment files using the Picard Tools module, AddOrReplaceReadGroups:

(a) `java -jar bin/AddOrReplaceReadGroups .jar VALIDATION_STRINGENCY=SILENT I=alignments/NA12878.ALN.dups_ removed.bam O=alignments/NA12878 .ALN.RG.bam SO=coordinate RGID=NA12878 RGLB=NA12878 RGPL=illumina RGPU=NA12878 RGSM=NA12878 CREATE_INDEX=true`

(5) Realignment around indels using the GATK modules RealignerTargetCreator and IndelRealigner:

(a) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T RealignerTargetCreator -R genome/hg19.fa -I alignments/ NA12878.ALN.RG.bam -known genome/mills.vcf -o tmp/ALN .intervals`

(b) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T IndelRealigner -R genome/hg19.fa -I alignments/NA12878.ALN.RG.bam -known genome/mills.vcf -o alignments/NA12878.ALN.indels.bam --maxReadsForRealignment 100000 --maxReadsInMemory 1000000 -targetIntervals tmp/ALN .intervals`

(6) Base recalibration using the GATK modules BaseRecalibrator and PrintReads:

(a) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T BaseRecalibrator -R genome/hg19.fa -I alignments/NA12878.ALN.indels.bam -knownSitesgenome/dbsnp_137 .hg19.excluding_sites_after_129 .only_standard_chroms.vcf -o tmp/NA12878.ALN.grp`

(b) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T PrintReads -R genome/hg19.fa -I alignments/NA12878.ALN.indels .bam -BQSR tmp/NA12878.ALN.grp -o alignments/NA12878.ALN.BQSR.bam`

2.7. Variant calling. Default arguments were used for each variant caller unless it contained a "threads" or "parallel" flag in which case that was used as well. Additionally, indels were called separately from SNVs where possible. Specifically, the commands used are as follows.

2.7.1. FreeBayes

(1) `freebayes -f genome/hg19.fa -i -X -u -v vcf_files/NA12878.ALIGNER.freebayes .raw.snv.vcf alignments/NA12878.ALIGNER .BQSR.bam`

(2) `freebayes -f genome/hg19.fa -I -X -u -v vcf_files/NA12878.ALIGNER.freebayes .raw.indel.vcf alignments/NA12878 .ALIGNER.BQSR.bam`

2.7.2. GATK HaplotypeCaller

(1) `java -XX:-DoEscapeAnalysis -jar bin/ GenomeAnalysisTK.jar -T HaplotypeCaller -R genome/hg19.fa -I alignments/NA12878 .ALIGNER.BQSR.bam --dbsnp $DBSNP -o vcf_files/NA12878.ALIGNER.HC .raw.vcf -stand_call_conf 50`

2.7.3. GATK UnifiedGenotyper

(1) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T UnifiedGenotyper -R genome/hg19.fa -nt 10 -I alignments/NA12878.ALIGNER .BQSR.bam -o vcf_files/NA12878.ALIGNER .UG.raw.snv.vcf -glm SNP -D $DBSNP`

(2) `java -XX:-DoEscapeAnalysis -jar bin/GenomeAnalysisTK.jar -T UnifiedGenotyper -R genome/hg19.fa -nt 10 -I alignments/NA12878.ALIGNER .BQSR.bam -o vcf_files/NA12878.ALIGNER .UG.raw.indel.vcf -glm INDEL -D $MILLS`

2.7.4. SAMtools Mpileup

(1) `samtools mpileup -uf genome/hg19.fa alignments/NA12878.ALIGNER.BQSR .bam | bcftools view -bvcg - > vcf_files/NA12878.ALIGNER .mpileup.bcf && bcftools view vcf_files/ NA12878.ALIGNER.mpileup.bcf > vcf_files/ NA12878.ALIGNER.mpileup.raw.vcf`

2.7.5. SNPSVM

(1) `java -XX:ParallelGCThreads=10 -jar tools/SNPSVM/snpsvm.jar predict -R genome/hg19.fa -B alignments/ NA12878.ALIGNER.BQSR.bam -M tools/ SNPSVM/models/default.model -V vcf_files/NA12878.ALIGNER.SNPSVM.raw.vcf`

Due to the nonexistence of requisite CIGAR flags in the alignment file, SNPSVM failed to call variants for CUSHAW3, and SAMtools mpileup could not call variants on MOSAIK alignments for the same reason. Also, due to the fact that SNPSVM only detects SNVs, no indels were reported for this program.

2.8. Variant Filtration. Filtration varied depending on the variant caller being used. In the cases of GATK HaplotypeCaller and GATK UnifiedGenotyper, the GATK modules, VariantRecalibrator and ApplyRecalibration, were used to filter SNVs using HapMap 3.3, the Omni 2.5 SNP BeadChip, and dbSNP 137 without 1000 Genome data as training sets. For SNPSVM, QUAL scores ≥ 4 and DP values ≥ 6 were used. For FreeBayes and SAMtools, QUAL scores ≥ 20 and DP values ≥ 6 were used.

2.9. Variant Comparison. For variant comparison, USeq 8.8.1 was used to compare SNVs shared between all datasets. To compare indels, the vcflib tool vcfintersect was used. The TruSeq hg19 exome bed file truseq_exome_targeted_regions.hg19.bed.chr, obtained in December 11, 2013, was used to restrict comparisons to locations that could be captured by the exome pull down kit used in the sequencing of SRR098401. This file can be obtained from Illumina here: http://support.illumina.com/sequencing/sequencing_kits/truseq_exome_enrichment_kit/downloads.ilmn. To ensure that variants were represented identically between different call sets, the vcflib tool vcfallelicprimitives was used to preprocess vcf files.

2.10. Statistical Calculations

True Positive (TP). It is a mutation that was detected by the pipeline being tested and is one that exists in the NIST-GiaB list.

False Positive (FP). It is a mutation that was detected by the pipeline being tested but is one that does not exist in the NIST-GiaB list.

True Negative (TN). It is a mutation that was not detected by the pipeline being tested and is one that does not exist in the NIST-GiaB list.

False Negative (FN). It is a mutation that was not detected by the pipeline being tested but is one that does exist in the NIST-GiaB list:

$$\begin{aligned} \text{PPV} &= \frac{\text{TP}}{(\text{TP} + \text{FP})}, \\ \text{Sensitivity} &= \frac{\text{TP}}{(\text{TP} + \text{FN})}. \end{aligned} \tag{1}$$

3. Results and Discussion

3.1. Prefiltering Variants. When performing variant analysis, one of the many pitfalls that must be taken into consideration is the exome sequence space (as defined by the exome capture kit) and how it can affect the analysis results. In this case, we had a single exome (SRR098401) that was extracted using the Illumina TruSeq exome kit and sequenced on a HiSeq 2000.

Table 3: Raw variant statistics for the 30 pipelines, including SNVs and indels.

Aligner	Genotyper	Raw TP SNVs	Raw FP SNVs	Raw TP indels	Raw FP indels
Bowtie2	FreeBayes	23,985	73,473	806	2,482
Bowtie2	GATK HC	21,631	273	771	1,103
Bowtie2	GATK UG	25,136	2,276	418	420
Bowtie2	mpileup	21,930	1,030	734	1,414
Bowtie2	SNPSVM	17,613	47	—	—
BWA mem	FreeBayes	23,857	18,256	785	2,088
BWA mem	GATK HC	21,707	367	779	1,348
BWA mem	GATK UG	21,925	213	402	408
BWA mem	mpileup	25,081	2,129	761	1,772
BWA mem	SNPSVM	17,920	65	—	—
BWA sampe	FreeBayes	23,789	27,143	737	1,872
BWA sampe	GATK HC	21,878	263	758	1,161
BWA sampe	GATK UG	22,153	321	394	385
BWA sampe	mpileup	25,206	2,205	684	1,401
BWA sampe	SNPSVM	18,017	78	—	—
CUSHAW3	FreeBayes	23,191	53,525	624	3,310
CUSHAW3	GATK HC	19,673	14,814	751	4,727
CUSHAW3	GATK UG	19,113	13,184	360	1,005
CUSHAW3	mpileup	22,171	9,694	681	1,983
CUSHAW3	SNPSVM	—	—	—	—
MOSAIK	FreeBayes	23,373	39,203	783	3,359
MOSAIK	GATK HC	13,528	111	500	458
MOSAIK	GATK UG	17,147	76	392	284
MOSAIK	mpileup	—	—	—	—
MOSAIK	SNPSVM	14,586	8	—	—
Novoalign	FreeBayes	22,794	2,970	678	1,554
Novoalign	GATK HC	21,407	473	779	1,370
Novoalign	GATK UG	21,113	144	387	365
Novoalign	mpileup	24,512	1,861	773	1,781
Novoalign	SNPSVM	17,109	164	—	—

With this in mind, we wanted to make sure that we were measuring the ability of the bioinformatic tools to do their jobs and not how well the Illumina TruSeq exome capture kit worked. That is, we only want to try to call variants that are supposed to be present in the exons as defined by the pull down kit. For this reason, we use the bed file provided by Illumina, not a generic annotation bed file, for example, RefSeq for hg19. We found that for this particular individual, according to the NIST-GiaB list, there should be a total of 34,886 SNVs and 1,473 indels within the regions defined by the TruSeq bed file.

Once we filtered out variants that were not located in the regions defined by the Illumina TruSeq exome bed file, we went from hundreds of thousands of putative variants (data not shown) to, on average, about 23,000 variants (SNVs and indels) per pipeline (Table 3). This is an important step for researchers to begin with, as it significantly reduces the search space for potentially interesting variants.

3.2. Raw Variant Results Compared to GiaB. One aspect we wanted to understand when doing this comparison was the importance of filtering variants detected by these tools. The reason for this is that ideally one would like to have as high a level of sensitivity as possible so that the mutations of interest do not get lost in the filtering process. It therefore behooves us to determine whether or not this step is necessary and to what degree it is necessary, since it is clear from the NIST-GiaB results and the Bamshad et al. [13] review that sensitivity could be an issue.

As we can see in Table 3, filtering is needed more for some variant callers than for others when it comes to SNVs, and it is absolutely necessary for indels. In most cases, the number of TP variants is close to or higher than our expected number of about 20,000 [13], but, on the other hand, in some cases the number of FPs is very high.

Clearly there is a lot of variation in the numbers generated by each pipeline. However, one can find some commonalities

in the numbers that likely stem from the algorithmic origins of each tool. FreeBayes produces both the largest number of unfiltered variants and the highest number of FPs. It is likely that we only see this kind of performance from this tool due to the fact that while it is not the only variant caller based on Bayesian inference it is unique in its interpretation of alignments. That is, it is a haplotype-based caller that identifies variants based on the sequence of the reads themselves instead of the alignment, the latter of which is how GATK's UnifiedGenotyper operates.

Additionally we see the Burrows-Wheeler based aligners perform very similarly to each other: Bowtie2, BWA mem, and BWA sampe achieve similar results across the board. One might surmise that this is likely due to the fact that all of these tools utilize similar algorithms when performing their designated task. This observation is supported by the fact that MOSAIK (gapped alignments using the Smith-Waterman algorithm) and CUSHAW3 (a hybrid seeding approach) both have very different underlying algorithms and subsequently produce very different results.

This difference in results correlating with different algorithms is seen best in the SNPSVM results. Of the variant callers, it is the only one that utilizes support vector machines and model building to generate SNV calls. It would appear that while it has the disadvantage of not being as sensitive as other methods it does benefit from being extremely accurate regardless of the aligner being used. This suggests that one is able to skip the filtering step altogether when using this variant caller.

With regard to indels, no aligner seems to stand out among the rest as one that handles this type of mutation well. In fact, when looking at the number of FPs, it is clear that it is the variant caller that plays the largest role in the accuracy of indel identification. Additionally, there are data for neither CUSHAW3 plus SNPSVM nor MOSAIK plus SAMtools mpileup pipelines due to the alignment files not containing the necessary CIGAR strings for the variant callers to function downstream. Lastly, the reason there are no indel data for SNPSVM is because this tool is solely used for identification of SNVs.

3.3. Filtered Results Compared to GiaB. As in Table 4, standard filtering practices manage to remove a large number of FP SNVs for each pipeline; however it seems that these filters are a bit too aggressive in most cases for SNV detection, but not strict enough for indels. This is made obvious when looking at the differences in the number of FPs reported in each dataset. For example, Bowtie2 with Freebayes sees a removal of 72,570 FP SNVs (a reduction of 98%) but only a removal of 1,736 FP indels (a reduction of 70%). It should also be noted that the filters used were pipeline-dependent and, for the most part, within each pipeline produced similar reductions in SNV and indel FPs. The one exception here is the number of variants identified from the CUSHAW3 alignments when compared to other alignments: overall the number of TP SNVs is lower, the number of FP SNVs is higher, and it is the only aligner that produces more than 1,000 FP indels after filtering.

Given the fact that filtering significantly reduces the number of TP variants, it might be wise to, with the exception of pipelines using CUSHAW3 and FreeBayes, skip this step when searching for rare, high-impact variants. Instead, one might spend more time on a filtering process that is based on biology rather than statistics. For example, it may make more sense to invest time identifying a small list of variants that are likely to be high-impact: splice site mutations, indels that cause frameshifts, truncation mutations, stop-loss mutations, or mutations in genes that are known to be biologically relevant to the phenotype of interest.

3.4. Average TPR and Sensitivity. As can be seen in Table 5, the Positive Predictive Value (PPV) for each tool, with the exception of CUSHAW3, ranges from 91% to 99.9% for SNVs, but the average sensitivity is very low (around 50%). This discrepancy could be due to a number of reasons, but the most likely one is variable depth of coverage across exons. We can see that, in addition to low SNV sensitivity, indel sensitivity is low (around 30%); however the PPV for indels is considerably lower (35.86% to 52.95%). This could be due to any of the following reasons: very short indels are hard to detect by conventional NGS [31], the representation of indels by different variant callers can cause tools to incorrectly claim that two indels are different, or alignment tools produce different representations of the same indel [7].

Perhaps the most likely explanation for both types of mutations is the issue of depth. As is the case with any variant analysis study, an increase in depth of coverage leads to an increase in sensitivity, but it is impossible to guarantee good depth of coverage due to the inability of exome capture kits to uniformly pull down exons [32–34]. Additionally, no single exome capture kit covers every exon. Indeed, it has been shown that variant analysis of whole genome sequencing at an average depth lower than an exome performs better due to the uniformity of said depth. Thus, it is likely that a large number of variants are missing due to the fact that the NIST-GiaB list was created from a compilation of exomic and genomic sequencing data. Ultimately, to achieve proper sensitivity one will eventually need to perform whole genome sequencing, but that is currently cost-prohibitive for most labs. Fortunately, this cost is continuing to drop, and we will soon see a gradual shift from exome analysis to the more complete whole genome analysis.

3.5. Sensitivity as a Function of Depth. Because sensitivity reflects one of the most important performance metrics of a tool and most of the tools struggle to achieve sensitivity higher than 50%, we would like to further explore how depth affected variant calling sensitivity. We looked at a number of different combinations of tools to determine what the best pipelines, variant callers, and aligners were. For Figure 2, we took the five best combinations of variant callers and aligners as determined by their sensitivity and false positive rate (FPR). That is, we selected those which had the highest number of TP SNVs called in addition to the lowest number of FP SNVs. Upon inspection, the thing that stands out immediately is that the sensitivity is lower than expected.

Table 4: Filtered variant statistics for the 30 pipelines, including SNVs and indels.

Aligner	Genotyper	Filtered TP SNVs	Filtered FP SNVs	Filtered TP indels	Filtered FP indels
Bowtie2	FreeBayes	17,504	903	481	746
Bowtie2	GATK HC	17,330	29	648	687
Bowtie2	GATK UG	19,937	49	395	338
Bowtie2	mpileup	17,049	153	402	541
Bowtie2	SNPSVM	13,983	8	—	—
BWA mem	FreeBayes	17,376	347	461	739
BWA mem	GATK HC	19,388	302	689	860
BWA mem	GATK UG	20,000	48	397	355
BWA mem	mpileup	17,070	57	403	606
BWA mem	SNPSVM	15,060	10	—	—
BWA sampe	FreeBayes	17,435	450	443	647
BWA sampe	GATK HC	19,438	214	630	725
BWA sampe	GATK UG	19,557	27	384	336
BWA sampe	mpileup	17,049	111	387	518
BWA sampe	SNPSVM	15,218	10	—	
CUSHAW3	FreeBayes	16,620	7,627	362	1,294
CUSHAW3	GATK HC	16,590	2,195	665	1,551
CUSHAW3	GATK UG	17,939	2,202	357	545
CUSHAW3	mpileup	15,942	4,029	368	796
CUSHAW3	SNPSVM	—	—	—	—
MOSAIK	FreeBayes	17,177	679	458	645
MOSAIK	GATK HC	11,616	33	426	255
MOSAIK	GATK UG	16,423	42	381	224
MOSAIK	mpileup	—	—	—	—
MOSAIK	SNPSVM	4,727	3	—	—
Novoalign	FreeBayes	16,658	219	384	559
Novoalign	GATK HC	19,406	385	702	872
Novoalign	GATK UG	20,521	46	386	315
Novoalign	mpileup	16,493	62	396	579
Novoalign	SNPSVM	14,451	18	—	—

Table 5: Average Positive Predictive Value (PPV) and sensitivity for each tool.

Tool	Average SNV PPV	Average SNV sensitivity	Average indel PPV	Average indel sensitivity
Bowtie2	98.69%	49.19%	45.45%	32.69%
BWA mem	99.15%	50.96%	43.24%	33.10%
BWA sampe	99.09%	50.85%	45.31%	31.30%
CUSHAW3	80.69%	48.08%	29.50%	29.74%
MOSAIK	98.51%	35.79%	52.95%	28.63%
Novoalign	99.17%	50.18%	44.55%	31.70%
FreeBayes	90.95%	51.00%	35.86%	32.79%
GATK HaplotypeCaller	97.05%	51.03%	43.17%	31.79%
GATK UnifiedGenotyper	97.93%	50.77%	52.12%	31.57%
SAMtools mpileup	94.99%	50.76%	39.15%	31.30%
SNPSVM	99.92%	50.85%	N/A	N/A

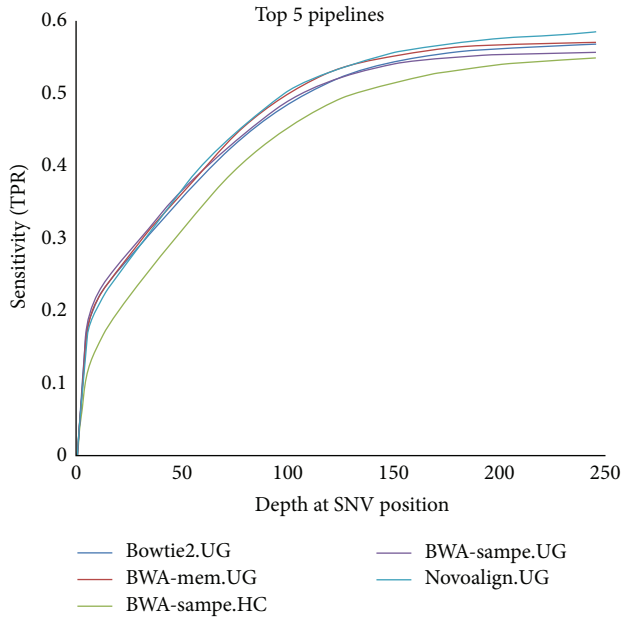


Figure 2: Sensitivity as a function of depth for the top five pipelines. The top five pipelines are shown here with the depth of every SNV plotted against sensitivity.

All of the pipelines perform at roughly the same level: they identify most of their variants by the time a depth of about 150x has been reached, which indicates that this depth is likely sufficient and that the number of missing variants is probably due to certain exons having lower than average coverage. Note that four out of the five best performing pipelines have GATK UnifiedGenotyper as their variant caller, demonstrating its superior performance irrespective of the aligner used as shown in Figure 3(b).

In addition to looking at the top five pipelines, we determined it would be useful to perform the same analysis on the best aligner coupled with every variant caller (Figure 3(a)), as well as the best variant caller coupled with every aligner (Figure 3(b)). As with the pipelines, the best aligner was identified as that which produced the highest number of TP SNVs and the lowest number of FP SNVs—in this case BWA mem. Despite having the best alignment to work with, we still see a fairly large difference between the variant callers, which is likely attributable to the different algorithms they employ (Figure 3(a)). However, in the case of the best performing variant caller, GATK UnifiedGenotyper, there seems to be less variation among the top four aligners indicating that it performs fairly well in most situations with the exceptions being CUSAHW3 and MOSAIK.

3.6. Shared Variants among the Top Pipelines. Lastly, we wanted to know just how unique the variant call sets were between the different pipelines. To do this, we again focused on the top five variant calling pipelines: Bowtie2 plus UnifiedGenotyper, BWA mem plus UnifiedGenotyper, BWA sampe plus HaplotypeCaller, BWA sampe plus UnifiedGenotyper, and Novoalign plus UnifiedGenotyper. As can be seen in Figure 4, there is a large amount of overlap between

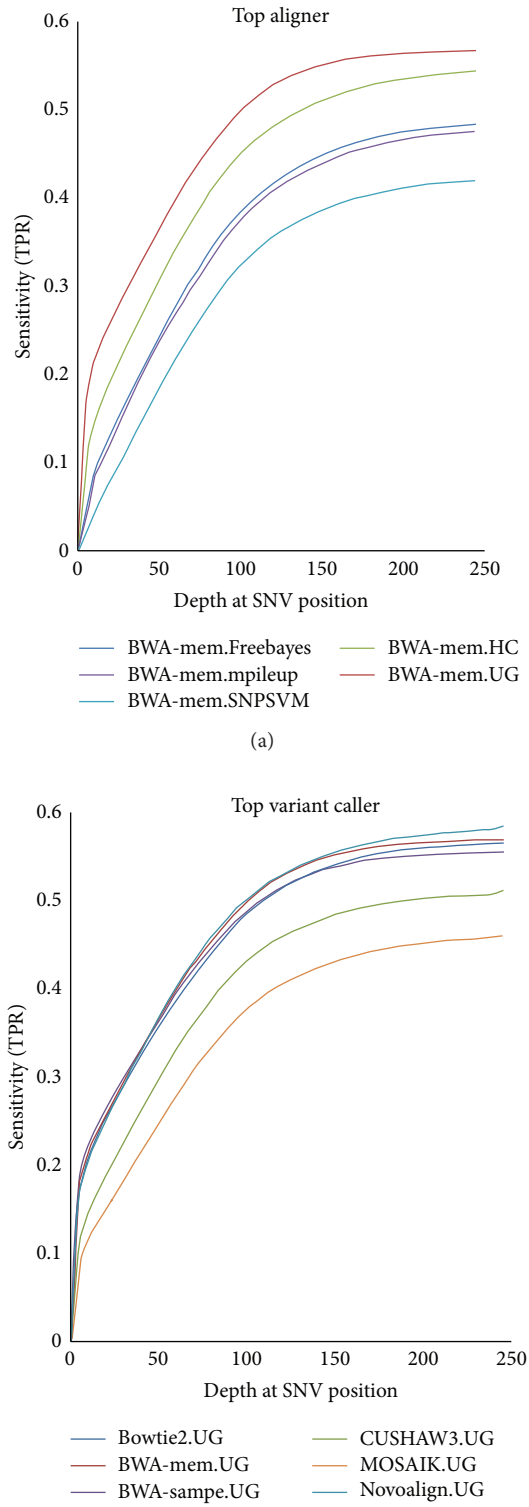


Figure 3: Sensitivity as a function of depth for the top aligner and top variant caller. (a) Results for the depth of every SNV plotted against sensitivity for the top aligner, BWA mem, paired with every variant caller. (b) Results for the depth of every SNV plotted against sensitivity for the top variant caller, GATK UnifiedGenotyper, paired with every aligner.

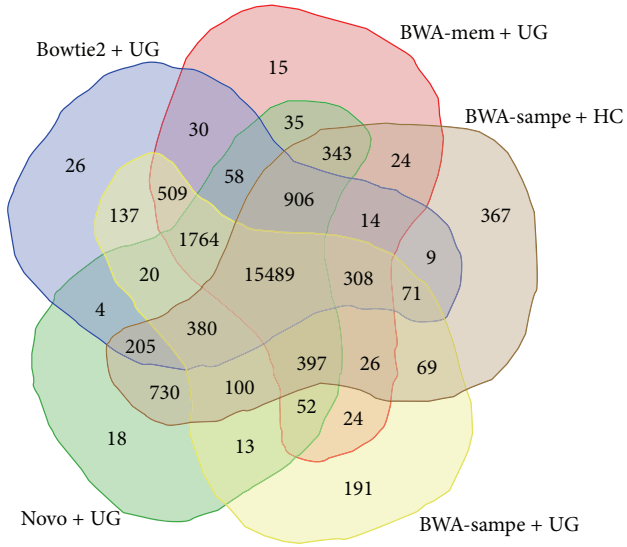


Figure 4: The intersection of the SNVs identified by the top five pipelines.

the five different pipelines in question, with 15,489 SNVs (70%) shared out of a total of 22,324 distinct variants. However, one could also argue that this is largely due to four of the five pipelines using the UnifiedGenotyper as their variant caller. This notion is corroborated by the fact that the largest number of variants unique to a pipeline, 367, belongs to the BWA sampe plus HaplotypeCaller combination. It is also worth noting that the second highest number of unique SNVs also belongs to the BWA sampe aligner, so it is possible that the high number of unique SNVs is better attributed to the aligner than the variant caller.

4. Conclusions

We found that among the thirty different pipelines tested Novoalign plus GATK UnifiedGenotyper exhibited the highest sensitivity while maintaining a low number of FP for SNVs. Of the aligners, BWA mem consistently performed the best, but results still varied greatly depending on the variant caller used. Naturally, it follows that the best variant caller, GATK UnifiedGenotyper, mostly produced similar results regardless of the aligner used. However, it is readily apparent that indels are still difficult for any pipeline to handle with none of the pipelines achieving an average sensitivity higher than 33% or a PPV higher than 53%. In addition to the low overall performance we see in detecting indels, sensitivity, regardless of mutation type, is a problem for every pipeline outlined in this paper. The expected number of SNVs for NA12878's exome is 34,886, but even when using the union of all the variants identified by the top five pipelines, the greatest number identified was very low (22,324). It seems that while still very useful exome analysis has its limitations even when it comes to something as seemingly simple as SNV detection.

Disclosure

Adam Cornish is a graduate student in Chittibabu Guda's lab with training in computer science and genomics. Chittibabu Guda (Associate Professor) has an interdisciplinary background in molecular and computational biology. He has published a number of computational methods with a variety of applications in biomedical research, since 2001.

Conflict of Interests

The authors are unaware of any competing interests.

Authors' Contribution

Adam Cornish designed the study, performed all analyses, made the figures, and wrote the paper. Chittibabu Guda provided essential feedback on improvements to the paper and input on the analyses themselves and thoroughly edited the paper. All authors read and approved the final paper.

Acknowledgments

This work was supported by the development funds to Chittibabu Guda from the University of Nebraska Medical Center (UNMC). The authors would like to thank the creators of Novoalign for making the software available as the trial version and the Bioinformatics and Systems Biology Core facility at UNMC for the infrastructure support facilitating this research.

References

[1] N. Paria, L. A. Copley, J. A. Herring et al., "Whole-exome sequencing: discovering genetic causes of orthopaedic disorders," *Journal of Bone and Joint Surgery—Series A*, vol. 95, no. 23, pp. e1851–e1858, 2013.

[2] L. G. Biesecker and R. C. Green, "Diagnostic clinical genome and exome sequencing," *The New England Journal of Medicine*, vol. 370, no. 25, pp. 2418–2425, 2014.

[3] N. Delanty and D. B. Goldstein, "Diagnostic exome sequencing: a new paradigm in neurology," *Neuron*, vol. 80, no. 4, pp. 841–843, 2013.

[4] M. E. Shanks, S. M. Downes, R. R. Copley et al., "Next-generation sequencing (NGS) as a diagnostic tool for retinal degeneration reveals a much higher detection rate in early-onset disease," *European Journal of Human Genetics*, vol. 21, no. 3, pp. 274–280, 2013.

[5] H. P. Buermans and J. T. den Dunnen, "Next generation sequencing technology: advances and applications," *Biochimica et Biophysica Acta (BBA)—Molecular Basis of Disease*, vol. 1842, no. 10, pp. 1932–1941, 2014.

[6] X. Liu, S. Han, Z. Wang, J. Gelernter, and B.-Z. Yang, "Variant callers for next-generation sequencing data: a comparison study," *PLoS ONE*, vol. 8, no. 9, Article ID e75619, 2013.

[7] J. M. Zook, B. Chapman, J. Wang et al., "Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls," *Nature Biotechnology*, vol. 32, no. 3, pp. 246–251, 2014.

[8] P. Krawitz, C. Rödelsperger, M. Jäger, L. Jostins, S. Bauer, and P. N. Robinson, "Microindel detection in short-read sequence data," *Bioinformatics*, vol. 26, no. 6, Article ID btq027, pp. 722–729, 2010.

[9] Z. Wei, W. Wang, P. Hu, G. J. Lyon, and H. Hakonarson, "SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data," *Nucleic Acids Research*, vol. 39, no. 19, article e132, 2011.

[10] R. Nielsen, T. Korneliussen, A. Albrechtsen, Y. Li, and J. Wang, "SNP calling, genotype calling, and sample allele frequency estimation from new-generation sequencing data," *PLoS ONE*, vol. 7, no. 7, Article ID e37558, 2012.

[11] J. Kim, Y. Lee, and N. Kim, "Bioinformatics interpretation of exome sequencing: blood cancer," *Genomics & Informatics*, vol. 11, no. 1, pp. 24–33, 2013.

[12] G. R. Abecasis, D. Altshuler, A. Auton et al., "A map of human genome variation from population-scale sequencing," *Nature*, vol. 467, no. 7319, pp. 1061–1073, 2010.

[13] M. J. Bamshad, S. B. Ng, A. W. Bigham et al., "Exome sequencing as a tool for Mendelian disease gene discovery," *Nature Reviews Genetics*, vol. 12, no. 11, pp. 745–755, 2011.

[14] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2," *Nature Methods*, vol. 9, no. 4, pp. 357–359, 2012.

[15] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[16] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," http://arxiv.org/abs/1303.3997.

[17] Y. Liu, B. Popp, and B. Schmidt, "CUSHAW3: sensitive and accurate base-space and color-space short-read alignment with hybrid seeding," *PLoS ONE*, vol. 9, no. 1, Article ID e86869, 2014.

[18] W.-P. Lee, M. P. Stromberg, A. Ward, C. Stewart, E. P. Garrison, and G. T. Marth, "MOSAIK: a hash-based algorithm for accurate next-generation sequencing short-read mapping," *PLoS ONE*, vol. 9, no. 3, Article ID e90581, 2014.

[19] F. Hach, F. Hormozdiari, C. Alkan, I. Birol, E. E. Eichler, and S. C. Sahinalp, "mrsFAST: a cache-oblivious algorithm for short-read mapping," *Nature Methods*, vol. 7, no. 8, pp. 576–577, 2010.

[20] M. David, M. Dzamba, D. Lister, L. Ilie, and M. Brudno, "SHRiMP2: sensitive yet practical short read mapping," *Bioinformatics*, vol. 27, no. 7, pp. 1011–1012, 2011.

[21] G. Lunter and M. Goodson, "Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads," *Genome Research*, vol. 21, no. 6, pp. 936–939, 2011.

[22] E. Garrison and G. Marth, "Haplotype-based variant detection from short-read sequencing," http://arxiv.org/abs/1207.3907.

[23] M. A. DePristo, E. Banks, R. Poplin et al., "A framework for variation discovery and genotyping using next-generation DNA sequencing data," *Nature Genetics*, vol. 43, no. 5, pp. 491–501, 2011.

[24] H. Li, B. Handsaker, A. Wysoker et al., "The sequence alignment/map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[25] B. D. O'Fallon, W. Wooderchak-Donahue, and D. K. Crockett, "A support vector machine for identification of single-nucleotide polymorphisms from next-generation sequencing data," *Bioinformatics*, vol. 29, no. 11, pp. 1361–1366, 2013.

[26] K. Cibulskis, M. S. Lawrence, S. L. Carter et al., "Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples," *Nature Biotechnology*, vol. 31, no. 3, pp. 213–219, 2013.

[27] R. Goya, M. G. F. Sun, R. D. Morin et al., "SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors," *Bioinformatics*, vol. 26, no. 6, pp. 730–736, 2010.

[28] D. C. Koboldt, Q. Zhang, D. E. Larson et al., "VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing," *Genome Research*, vol. 22, no. 3, pp. 568–576, 2012.

[29] R. Li, Y. Li, X. Fang et al., "SNP detection for massively parallel whole-genome resequencing," *Genome Research*, vol. 19, no. 6, pp. 1124–1132, 2009.

[30] G. A. Van der Auwera, M. O. Carneiro, C. Hartl et al., "From fastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline," *Current Protocols in Bioinformatics*, vol. 43, 2013.

[31] C. A. Albers, G. Lunter, D. G. MacArthur, G. McVean, W. H. Ouwehand, and R. Durbin, "Dindel: accurate indel calls from short-read data," *Genome Research*, vol. 21, no. 6, pp. 961–973, 2011.

[32] V. N. Rykalina, A. A. Shadrin, V. S. Amstislavskiy et al., "Exome sequencing from nanogram amounts of starting DNA: comparing three approaches," *PLoS ONE*, vol. 9, no. 7, Article ID e101154, 2014.

[33] A. M. Meynert, M. Ansari, D. R. FitzPatrick, and M. S. Taylor, "Variant detection sensitivity and biases in whole genome and exome sequencing," *BMC Bioinformatics*, vol. 15, article 247, 2014.

[34] C. S. Chilamakuri, S. Lorenz, M.-A. Madoui et al., "Performance comparison of four exome capture systems for deep sequencing," *BMC Genomics*, vol. 15, article 449, 2014.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 196082, 10 pages
http://dx.doi.org/10.1155/2015/196082

Research Article

Assessing Computational Steps for CLIP-Seq Data Analysis

Qi Liu,[1,2] Xue Zhong,[1] Blair B. Madison,[3] Anil K. Rustgi,[4,5,6] and Yu Shyr[1,7,8]

[1]*Center for Quantitative Sciences, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*
[2]*Department of Biomedical Informatics, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*
[3]*Department of Medicine, Washington University School of Medicine, St. Louis, MO 63110, USA*
[4]*Division of Gastroenterology, University of Pennsylvania Perelman School of Medicine, Philadelphia, PA 19104, USA*
[5]*Department of Medicine, University of Pennsylvania Perelman School of Medicine, Philadelphia, PA 19104, USA*
[6]*Abramson Cancer Center, University of Pennsylvania Perelman School of Medicine, Philadelphia, PA 19104, USA*
[7]*Department of Cancer Biology, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*
[8]*Department of Biostatistics, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*

Correspondence should be addressed to Yu Shyr; yu.shyr@vanderbilt.edu

Received 19 November 2014; Accepted 7 January 2015

Academic Editor: Cheol Yong Choi

RNA-binding protein (RBP) is a key player in regulating gene expression at the posttranscriptional level. CLIP-Seq, with the ability to provide a genome-wide map of protein-RNA interactions, has been increasingly used to decipher RBP-mediated posttranscriptional regulation. Generating highly reliable binding sites from CLIP-Seq requires not only stringent library preparation but also considerable computational efforts. Here we presented a first systematic evaluation of major computational steps for identifying RBP binding sites from CLIP-Seq data, including preprocessing, the choice of control samples, peak normalization, and motif discovery. We found that avoiding PCR amplification artifacts, normalizing to input RNA or mRNAseq, and defining the background model from control samples can reduce the bias introduced by RNA abundance and improve the quality of detected binding sites. Our findings can serve as a general guideline for CLIP experiments design and the comprehensive analysis of CLIP-Seq data.

1. Background

RNA-binding proteins (RBPs) are the primary regulator of posttranscriptional gene expression [1]. As soon as RNAs are transcribed, they are associated with RBPs to form ribonucleoprotein (RNP) complexes. The RBP-RNA associations modulate the biogenesis, stability, cellular localization, and transport of the RNA and determine the fate and function of RNA molecules. Therefore, a high resolution and precise map of protein-RNA interactions is essential for deciphering posttranscriptional regulation under various biological processes.

CLIP (cross-linking and immunoprecipitation) is the main technology for studying protein-RNA interactions *in vivo* [2–4]. CLIP uses ultraviolet irradiation to form covalent crosslinks only at direct sites between RBP and RNAs *in situ*, followed by immunoprecipitation of the protein-RNA complex with an antibody specific to the RBP of interest. The copurified RNA fragments are amplified and sequenced and mapped back to the reference genome to reveal RBP binding sites. CLIP has been successfully applied to study protein-RNA interactions in bacteria, fungi, yeast, and mammals [4–10]. To obtain a more comprehensive view of protein-RNA interactions, recently, CLIP coupled with high throughput sequencing technology (CLIP-Seq or HITS-CLIP) [11–15] and several alternative approaches, such as photoactivatable ribonucleoside-enhanced CLIP (PAR-CLIP) [16–20] and individual nucleotide resolution CLIP (iCLIP) [21–26], has been developed. Compared with the low-throughput CLIP data, these approaches allow genome wide mapping of protein-RNA interactions and have demonstrated their power for a number of proteins [12–14, 27–34]. For example, in contrast to only 34 Nova-bound transcripts detected in the original CLIP experiments, the first application of HITS-CLIP identified 2,481 Nova-bound RNAs [12].

The genome-wide insights provide a robust and unbiased means to study RBP function and predict RBP action *de novo*, leading to a tremendous progress in these areas.

Given the large amount of data generated by CLIP-Seq, considerable computational effort is required to generate reliable quantitative information of protein-RNA interactions [35]. A series of computational steps is involved in CLIP-Seq analysis, including data preprocessing, reads mapping, peak calling, normalization and annotation, and motif discovery [35]. Although more approaches and tools have been developed to address the challenges of peak calling by considering the global and local background [36–40], there has been no clear consensus on the appropriate way to implement each computational step or the impact of a chosen step on the downstream analysis. For example, some studies used all reads to call peaks, while others employed reads after duplication removal thinking that reads mapped to the same location are due to amplification bias [12, 29, 34]. As another example, early studies simply used the read counts to rank peaks [12], while recent methods ranked the sites by the relative enrichment of CLIP-data to the average CLIP count within the transcript for RBPs binding pre-mRNAs or to individual gene expression for RBPs binding mRNAs, trying to correct the bias introduced by RNA abundance [28–30, 32, 41]. Here we performed a comprehensive evaluation of different strategies to preprocess the data, normalize the peaks, and choose background sequences in the motif discovery. We generated CLIP data for Lin28b in three different colon cancer cell lines (Caco-2, DLD1, and Lovo) and mouse colon tissues with input RNA or corresponding RNAseq as controls [31]. We compared different strategies on the accuracy of identifying LIN28B binding sites. Our findings can serve as a general guideline on the appropriate way to implement each computational step, which will enable the design of improved computational and experimental protocols for CLIP-Seq analysis to further investigate posttranscriptional regulatory networks.

2. Materials and Methods

2.1. CLIP, Input RNA, and mRNAseq. CLIP samples were prepared from Caco-2 cells (three replicates), DLD1, and Lovo cell lines (one replicate each) with a doxycycline-inducible LIN28B and colonic epithelia of Vil-Lin28b^{Med} mice (two replicates) using a modified CLIP-Seq protocol [31]. For crosslinking at 254 nm, cells were irradiated on ice using stratalinker 2400 (Stratagene). Immunoprecipitation was performed overnight at 4°C using anti-FLAG M2 magnetic beads (M8823, Sigma) for Flag-HA-tagged mouse or human LIN28B. RNase T1 digestion, adapter ligations, and RNP isolation were described in [31]. Caco-2 cells and colon epithelium were sequenced by Illumina Hiseq 2000 as multiplexed samples, while DLD1 and Lovo were each sequenced on a single lane.

We used two basic methods to produce control samples, input RNA, and mRNAseq. Caco-2 used input RNA as control samples, while DLD1, Lovo, and mouse colonic epithelium had mRNAseq (Figure 1). Input RNA samples were prepared from total RNA extracted from UV crosslinked Caco-2 cells by digestion for 30 minutes in Proteinase-K (Roche). RNA was depleted of ribosomal RNA using the RiboMinus Transcriptome Isolation Kit (K1550-02, Life Technologies) and then digested with 2 units of RNase-T1 (Fermentas). Total RNA samples from DLD1, Lovo, and mice colonic epithelium were depleted of ribosomal RNA and libraries were prepared using the NEBNext mRNA Library Prep Master Mix Set for Illumina (E6110S, New England Biolabs).

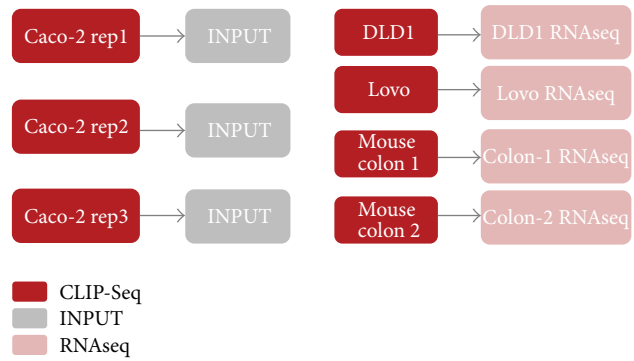


Figure 1: Experimental design of LIN28B CLIP-Seq. There are three replicates in CLIP Caco-2 samples with input RNA (no antibody) as control samples, one replicate each for DLD1 and Lovo CLIP samples with mRNAseq as control, and two replicates for mouse colon epithelium CLIP with mRNAseq as control.

2.2. Reads Trimming and Mapping. CLIP reads for Caco-2 (50 bp), DLD1 (36 bp), Lovo (36 bp), and input RNA reads for Caco-2 (50 bp) were trimmed to remove adaptor sequences and mapped to human reference genome (hg19) using Novoalign (parameters: *-l* 18 *-t* 85 *-h* 90) (http://www.novocraft.com/), which require unambiguous mapping to the genome with ≤2 substitutions, insertions or deletions in ≥18 nt and homopolymer score ≥90. CLIP reads for mouse colonic epithelium (50 bp) were mapped to mouse reference genome (mm9) using Novoalign. mRNAseq reads for DLD1 and Lovo cell lines (101 and 100 bp) were mapped to human reference genome (hg19) using TopHat [42] and mRNAseq reads for mouse colonic epithelium (50 bp) were mapped to mouse reference genome (mm9) using Novoalign.

There were ~33–48 million reads for each CLIP Caco-2 sample and ~30% of reads could be uniquely mapped to the genome. In contrast, only ~12% of reads in input Caco-2 samples could be uniquely mapped to the genome, which was due to more severe adapter contamination. The percentage of pure adapter reads was much higher in input samples (~58%) than in CLIP samples (~25%) (Additional File 1 (see Supplementary Material available online at http://dx.doi.org/10.1155/2015/196082)). There were ~17–22 million reads for CLIP DLD1, Lovo, and mouse samples, ~200 million reads for DLD1 and Lovo RNAseq samples, and ~60 million reads for mouse colon RNAseq samples. About 20% of reads could be uniquely mapped to the genome for CLIP samples, while ~60% of reads could be uniquely aligned to

Table 1: Mapping summary of CLIP, INPUT, and RNAseq reads.

Sample		Total reads	Aligned reads (%)	Unique aligned reads (%)
Human	Caco2_CLIP_1	34,498,894	12,095,664 (35.1%)	10,977,657 (31.8%)
	Caco2_CLIP_2	33,617,355	12,650,034 (37.6%)	11,514,201 (34.3%)
	Caco2_CLIP_3	48,866,964	17,965,486 (36.8%)	15,502,638 (31.7%)
	Caco2_INPUT_1	26,095,707	4,634,066 (17.8%)	3,257,784 (12.5%)
	Caco2_INPUT_2	40,683,388	7,954,926 (19.6%)	5,717,731 (14.1%)
	Caco2_INPUT_3	33,214,425	6,393619 (19.2%)	3,729901 (11.2%)
	DLD1_CLIP	36,860,853	18,303,689 (49.7%)	10,826,660 (29.4%)
	DLD1_RNAseq	196,664,529	120,668,419 (61.4%)	111,056,779 (56.5%)
	Lovo_CLIP	35,860,144	16,426,136 (45.8%)	8,435,054 (23.5%)
	Lovo_RNAseq	193,142,724	130,200,660 (67.4%)	120,721,857 (62.5%)
Mouse	Colon_CLIP_1	62,821,728	18,841,261 (30.0%)	13,884,667 (22.1%)
	Colon_CLIP_2	63,087,357	22,749,788 (36.1%)	15,226,733 (24.1%)
	Colon_RNAseq_1	56,706,471	44,663,179 (78.8%)	34,097,906 (60.1%)
	Colon_RNAseq_2	62,437,020	51,525,926 (82.5%)	40,559,464 (65.0%)

the genome for RNAseq samples. The mapping results were summarized in Table 1. We also used BWA to map CLIP reads to the genome with default parameters and obtained lower percentage of aligned reads than Novoalign (data not shown here).

2.3. CLIP Peaks Calling and Normalization. CLIP peaks were called by HOMER (http://homer.salk.edu/homer/index.html) [43]. The global threshold for the number of reads that determine a valid peak was selected at a false discovery rate of 0.001 based on a Poisson distribution [43]. Peak sizes were chosen based on the length distribution of mappable reads.

It is known that CLIP-Seq read counts depend on the expression abundance of the corresponding transcript. To reduce the distortion introduced by sequencing bias or abundant RNA, normalization is recommended to make binding sites across the full transcriptome comparable [41]. Here we compared five different strategies to normalize and rank peaks: (1) no normalization, which simply ranks the peaks by the reads coverage (Raw); (2) normalizing to the average CLIP data, which ranks the peaks by the relative enrichment of CLIP counts to the average CLIP counts within the transcript (AVE-CLIP). This strategy is generally recommended to study RBPs binding pre-mRNA because it is difficult to measure the RNA abundance by the traditional RNAseq techniques; (3) normalizing to the average input RNA data, which ranks the peaks by the relative enrichment of CLIP counts to the average input counts within the transcript (AVE-INPUT); (4) normalizing to the input RNA, which ranks the peaks by the relative enrichment of CLIP counts to input counts within the same sites (INPUT); (5) normalizing to RNAseq (RPKM), which ranks the peaks by the relative enrichment of CLIP counts to the transcript abundance, obtained from RNAseq. Here RPKM (reads per kilobase of exon model per million mapped reads) was calculated to estimate the transcript abundance, where read counts were normalized by the transcript length as well as the total number of mappable reads. Using RNAseq as control sample is recommended and has proved to be useful in the analysis of RBPs targeting messenger RNAs (mRNAs) [29, 41].

2.4. Quality of Binding Sites. LIN28 is a conserved RNA-binding protein. It plays an important role in differentiation, reprogramming, and oncogenesis [44–47]. Mammals have two paralogs, Lin28a and Lin28b. In the previous studies, the motif GGAG has been reported as the binding site of Lin28a [48], which was also confirmed by crystal structure of mouse Lin28a in complex with pre-let7 families [49]. Although two recent CLIP-Seq experiments on Lin28a revealed different

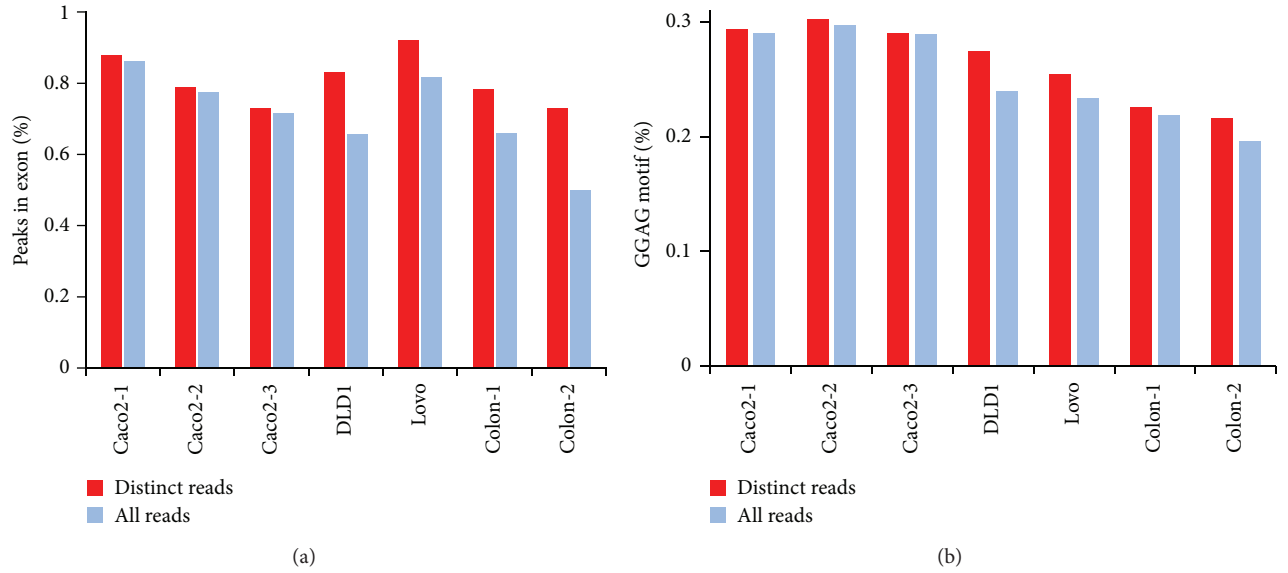


Figure 2: The percentage of CLIP peaks located in exonic regions and the GGAG motif occurrence when all reads or distinct reads were used.

binding motifs, they both contained "GGAG" [27, 33]. Wilbert et al. found that Lin28 bound to GGAGA sequences [33], while Cho et al. reported that AAGGAG was the most frequently observed hexamers [27]. In addition, both of these two studies found that Lin28a-binding sites were enriched in exons but depleted of intronic regions, indicating that Lin28a largely interacts with messenger RNA. Lin28a and Lin28b have different physiological expression patterns but similar behavior *in vitro* [49]. Lin28b CLIP peaks were found mainly within mRNAs, with 70%~90% located in exonic regions [31]. The motif GGAG was detected in the binding sites of let-7 (Additional File 2). *De novo* motif analysis of robust CIMSs (cross-link induced mutation sites) from Caco-2 cells yielded the motif similar to GGAG [31]. Collectively, Lin28b, similar to Lin28a, binds messenger RNAs at the GGAG motif.

We used two criteria to assess the quality of peaks, the percentage of peaks located in exonic regions and the percentage of peaks containing the GGAG motif. The higher the percentage of exonic peaks and GGAG motif occurrence, the better the peak quality. Human exonic regions were obtained from Ensembl version 65. Mouse exonic regions were obtained from Ensembl version 61. Peaks that overlap with the annotated exonic reads at least 1 bp were counted as exonic peaks using BEDTools.

3. Results

3.1. Removing PCR Amplification Bias. PCR amplification artifacts distort the quantitative analysis of sequencing data. This problem is exacerbated in CLIP-Seq experiments whose library complexity is limited owing to numerous enzymatic steps required in the protocol and the small amount of starting material. One way to reduce amplification bias is duplication removal. To assess the effect of duplication removal, we compared the quality of peaks identified using either all unique mappable reads or distinct reads (reads mapped to same locations were collapsed) with the same threshold (FDR < 0.001). Compared to the method using all reads, using distinct reads yielded fewer peaks but a higher percentage of peaks in exonic regions for all CLIP samples (Figure 2). A slight increase in the percentage of exonic peaks was observed for Caco-2 CLIP samples using distinct reads versus all reads, while a larger increase was shown for DLD1, Lovo, and mouse colon samples, especially for DLD1 and Colon-2 CLIP samples (Additional File 3). For DLD1 CLIP samples, 83.2% of 17225 peaks were located in exonic regions using distinct reads. In contrast, only 65.6% of 65548 peaks were situated in exonic regions using all reads. Even if 16671 peaks were identified with a more stringent threshold, close to the number of peaks with distinct reads, only 79.8% (13311) were located in exonic regions. For mouse colon-2 CLIP samples, 72.9% of 6781 peaks were located in exonic regions using distinct reads, while only 49.8% of 13,302 peaks were found in exonic regions using all mappable reads. Furthermore, using distinct reads obtained higher percentage of peaks with GGAG motif compared to using all reads, especially for DLD1 and Lovo CLIP samples (Figure 2 and Additional File 3). The results suggest that using distinct reads reduces PCR amplification bias, leading to the improved quality of peaks.

3.2. Peak Normalization and Ranking. After peaks are identified, normalization is recommended to quantify protein-RNA interactions, making peaks across the transcriptome comparable. Here we assessed the performance of different peak normalization and ranking strategies. Important differences were observed between different methods. For Caco-2 CLIP samples with input RNA as control, we compared four different ranking and normalization strategies, including Raw, AVE-CLIP, AVE-INPUT, and INPUT (Materials and

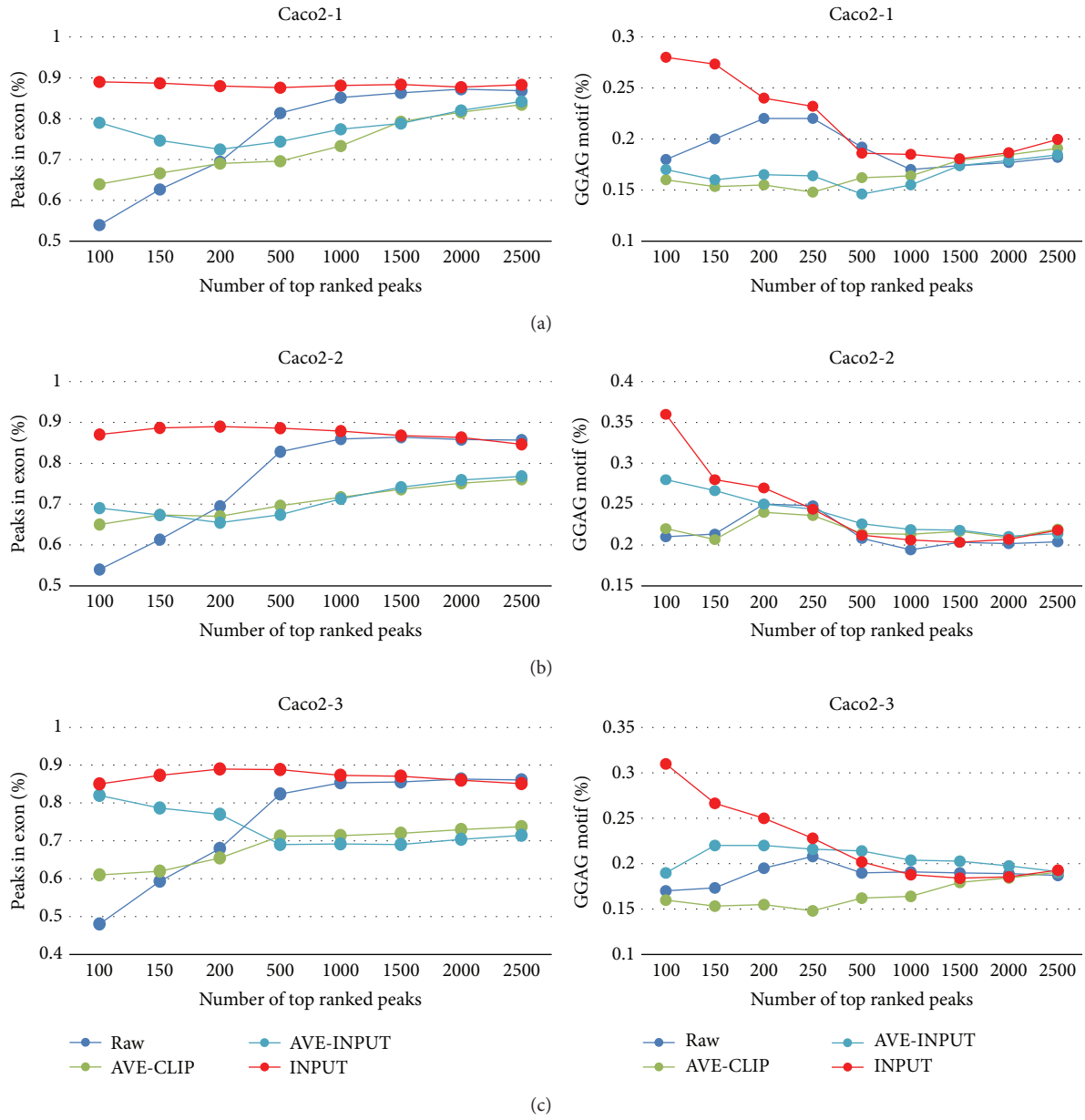


Figure 3: The percentage of peaks in exonic regions and the GGAG motif occurrence for Caco2 CLIP samples when four different normalization and ranking strategies were used, including Raw, AVE-CLIP, AVE-INPUT, and INPUT.

Methods). We estimated their performance by the percentage of peaks located in exonic regions and the GGAG motif occurrence among the top ranked peaks. The higher the rank is, the more reliable the peaks should be. That is, the accuracy is supposed to decrease as the number of top ranked peaks increases. However, the method without normalization (Raw) got lower percentage of exonic peaks and GGAG motif occurrence among more highly ranked peaks (Figure 3). For example, there were only 54% of peaks located in exonic regions among the top 100 peaks for the CLIP Caco-2 replicate 1 sample, in contrast to 70% in the top 200 peaks and 87.5% in the top 3000 peaks. Similarly, only 18% of peaks contained the GGAG motif in the top 100 peaks, compared to 22% in the top 200 peaks (Figure 3). These results suggest that there are lots of nonspecific and background binding in the highly ranked peaks. AVE-CLIP, although recommended by previous studies, did not show good performance and it even had the lowest percentage of the GGAG motif occurrence compared to the other three methods (Figure 3), indicating that averaging the CLIP-data on the individual transcript is not an appropriate way to reduce the bias introduced by the transcript abundance but instead weakens the signal. Input sample as control helped remove false positives. Normalizing to AVE-INPUT obtained higher percentage of exonic peaks and GGAG motif occurrence in the highly ranked peaks than the "Raw" method (Figure 3). However, it performed worse

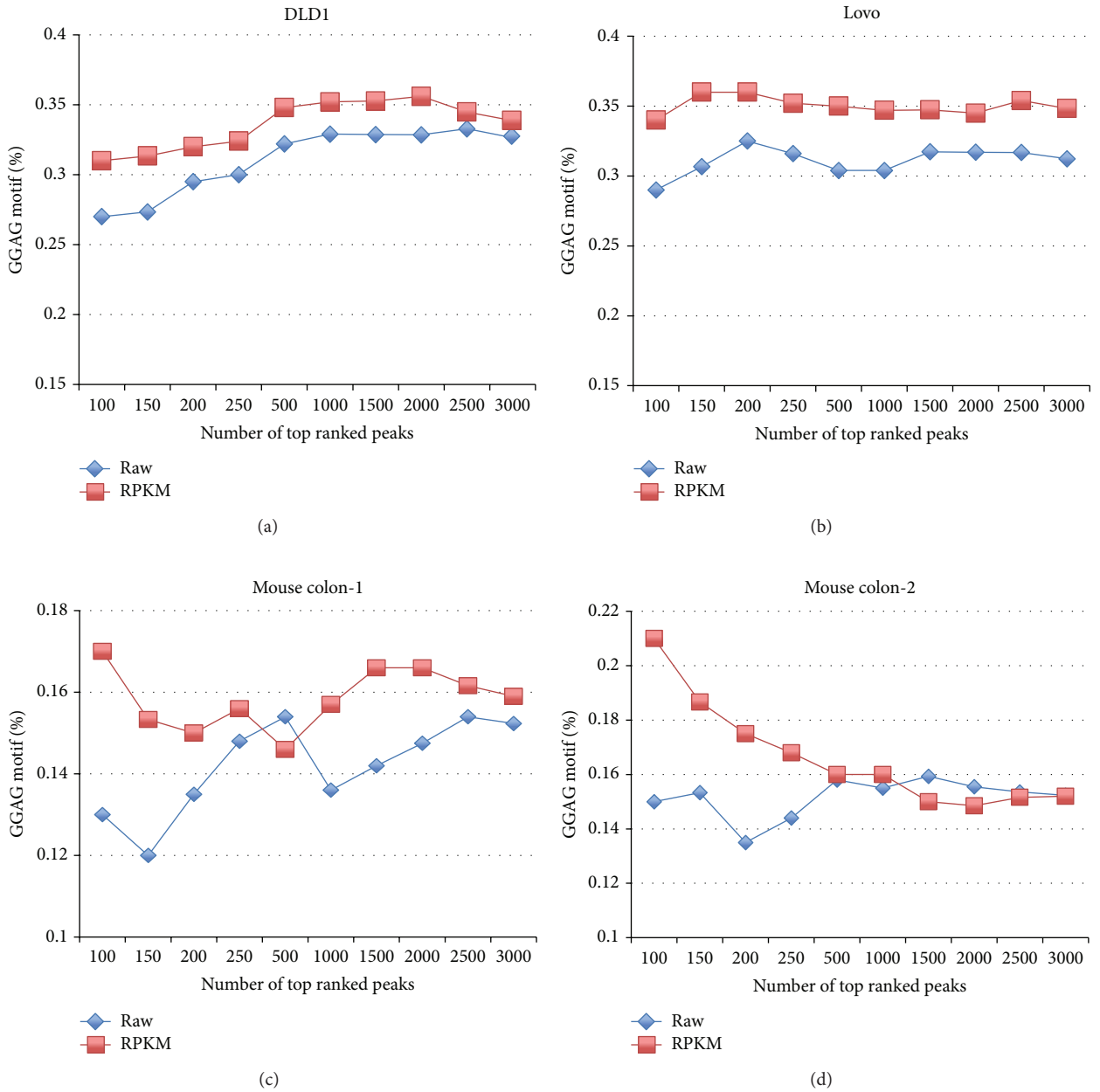


Figure 4: The percentage of peaks containing the GGAG motif for DLD1, Lovo, and mouse colon CLIP samples when peaks were ranked by reads coverage (Raw) or by relative enrichment to RPKM.

when the number of top ranked peaks increased, suggesting that it distorts the binding affinity when the signal turns weaker. Normalizing to the INPUT performed best, which yielded the highest percentage of exonic peaks and the GGAG motif compared to the other three methods (Figure 3). The protein level changes of Lin28b targets following Lin28b knockdown were also correlated with the relative enrichment of CLIP reads to INPUT ($R = 0.79$) [31], which further demonstrate the usefulness of normalizing CLIP reads to INPUT.

For DLD1, Lovo, and mouse colon CLIP samples with the corresponding RNAseq as control, we compared ranking by the relative enrichment of CLIP reads to RPKM with the simple ranking by the reads coverage (Raw). Consistent with previous studies, ranking peaks by the relative enrichment of CLIP reads to RPKM performed better than the simple ranking method, which yielded higher percentage of the GGAG motif occurrence in DLD1, Lovo, and mouse colon samples (Figure 4). In addition to the motif occurrence, we also identified peaks common in both DLD1/Lovo and mouse samples, which are evolutionary conserved and can be considered as reliable Lin28b binding targets. The rank of binding targets in human and mouse would be correlated if the ranking could represent binding affinity. That is, the higher the correlation, the better the representation of binding affinity of the ranking strategy. We compared four

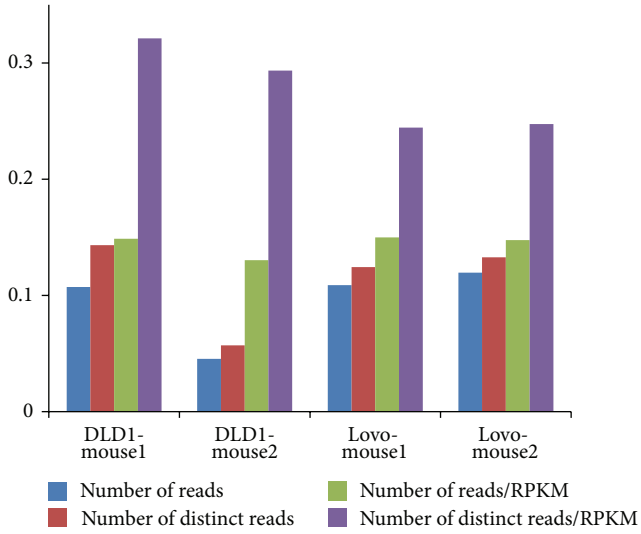


Figure 5: The correlation coefficient of peaks detected both in DLD1/Lovo and in mouse using different strategies.

different strategies, which included ranking by the coverage of all reads, distinct reads, the relative enrichment of all reads to RPKM, and the relative enrichment of distinct reads to RPKM (Figure 5). Consistent with our previous results, using distinct reads obtained higher correlation than using all reads. Furthermore, ranking peaks by the relative enrichment of distinct reads to RPKM obtained the highest correlation, ranging from 0.25 to 0.32. These results suggest that normalizing CLIP data to mRNAseq can improve the specificity when RBP targeting messenger RNA.

3.3. Motif Discovery. Motif discovery within CLIP peaks or surrounding regions reveals the unanticipated sequence signals associated with the RBP of interest. In a typical application of *de novo* motif analysis, motifs are generally discovered by differential enrichment analysis between sequences of peak regions and background sequences. Therefore, selection of an appropriate set of background sequences is very important. We enumerated all tetramers, ranked them by occurrence frequency, and filtered out insignificant ones. We used two different sets of background sequences to calculate the significance of enrichment. One was automatically generated by Homer where sequences were randomly selected from the genome with matched GC content [43], while the other was extracted from peaks identified from the INPUT samples. CTGG, GCTG, CCTG, TGGA, and TCCT were found to be the most frequently occurring tetramers in all Caco-2 CLIP samples; however, these motifs did not show any significant enrichment when compared to the background INPUT sequences (P value = 1) (Additional File 4). Previous studies have reported the excess of CTG and CT/TG on coding sequence structures [50], suggesting that the excessive recurrences of these motifs are mostly due to the coding features but not associated with Lin28b. Those false positives were successfully removed by differential motif analysis between target sequences and the background INPUT sequences. After filtering (FDR < 0.001), the motif GGAG ranked the first in all three Caco-2 replicate samples (Table 2 and Additional File 4), which provided validation of the success of using INPUT sequences as background. The method using randomly generated sequences from the reference genome as background reduced some degree of bias introduced by coding features, for example, filtering out the motif CCTG; however, it also removed the true motif GGAG. Since CLIP target regions are RNA, which generally has different sequence features from genomic regions, generating random sequences from the genome cannot define the background model correctly.

4. Discussion

Using Lin28b CLIP-Seq studies with input RNA or mRNAseq as control samples, we presented a systematic evaluation of different strategies to implement data preprocessing, peak normalization and ranking, and motif discovery for the analysis of CLIP-Seq data. We found that counting only distinct reads, normalizing to input RNA or mRNAseq, and defining the background model from control samples can improve the quality of binding sites. These findings will enable the design of refined experimental and computational protocols of CLIP-Seq studies.

To date, generating a high resolution and precise map of protein-RNA interactions still remains a big challenge, which requires novel experimental, computational, and statistical solutions. The crosslinking efficiency varies between proteins and the optimal protocol should be determined experimentally for individual proteins. Improving antibody specificity and optimizing the conditions of partial digestion with a relatively unspecific nuclease will help increase the library complexity and reduce the number of false positives. Recently, analysis of cross-linked induced mutation sites (CIMS) provides a nucleotide-resolution map of protein-RNA interactions [31, 51, 52]. Combining CLIP peaks with CIMS will improve the quantification of RBP binding. However, it is still far from fully understanding fundamental properties of cross-linking and its local sequence preference. Unlike Nova-CLIP and Ago-CLIP [51], which revealed deletions in ~8%–20% of CLIP reads, Lin28b-CLIP had very low percentage of reads containing deletions [31]. The percentage of insertion and substitution reads were similar between CLIP and input samples and cannot be used as CIMS signatures [31]. Therefore, the percentage of CIMS sites and the usefulness of each kind of CIMS signatures (insertion, deletion, and substitution) should be evaluated before CIMS sites are used. Novel bioinformatics methods to account for the local sequence features will help identify binding sites and reveal the binding motif accurately. Additionally, computational methods are needed to summarize scores of peaks in the given RNA and measure the effect of protein-RNA interactions. Finally, binding does not always suggest function. With more binding preferences of RBPs studied [53] and various kinds of omics data available [54], integrating CLIP-Seq with multiple omics data is necessary to reliably infer the functional effect of RBP binding events

Table 2: Top 10 tetramers in Caco-2 CLIP peaks (FDR < 0.001).

(a)

	Caco2-1			
	Background from INPUT		Raw	Background from genome
4mers	Percent (%)	*P* value	Rank	*P* value
GGAG[a]	19.6[a]	1.36*e* − 07[a]	10	1.00
AGGA[b]	18.4[b]	8.61*e* − 13[b]	17	1.00
CAGG[c]	18.3[c]	4.52*e* − 20[c]	19	1.00
TGGC[d]	18.3[d]	4.54*e* − 04[d]	20	0.18
GTGG[e]	18.3[e]	4.96*e* − 12[e]	21	0.65
TGGG[f]	17.8[f]	3.65*e* − 12[f]	23	1.00
AAGA	17.0	4.72*e* − 09	28	0.05
GCAG[g]	17.0[g]	1.20*e* − 05[g]	29	1.00
GGTG	16.8	2.56*e* − 12	30	1.00
GGCT	16.8	2.11*e* − 07	32	1.00

(b)

	Caco2-2			
	Background from INPUT		Raw	Background from genome
4mers	Percent (%)	*P* value	Rank	*P* value
GGAG[a]	20.1[a]	9.07*e* − 10[a]	9	1.00
TGGC[d]	19.1[d]	2.97*e* − 06[d]	13	0.05
GTGG[e]	18.8[e]	1.07*e* − 13[e]	16	0.41
TGGG[f]	18.4[f]	6.45*e* − 11[f]	19	1.00
GGTG	17.9	9.79*e* − 15	23	1.00
AGGA[b]	17.7[b]	8.42*e* − 09[b]	24	1.00
GCAG[g]	17.6[g]	3.70*e* − 11[g]	26	1.00
CAGG[c]	17.3[c]	1.00*e* − 14[c]	27	1.00
CAGA	17.0	1.27*e* − 07	30	1.00
AAGA	16.1	1.78*e* − 05	39	1.00

(c)

	Caco2-3			
	Background from INPUT		Raw	Background from genome
4mers	Percent (%)	*P* value	Rank	*P* value
GGAG[a]	21.2[a]	1.05*e* − 32[a]	7	1.00
GAAG	19.5	1.98*e* − 11	12	7.21*e* − 11
TGGC[d]	19.4[d]	3.19*e* − 08[d]	13	3.22*e* − 04
CAGG[c]	18.3[c]	9.98*e* − 28[c]	18	1.00
AGGA[b]	17.9[b]	1.26*e* − 23[b]	21	1.00
TGGG[f]	17.9[f]	1.39*e* − 19[f]	22	1.00
GTGG[e]	17.8[e]	2.68*e* − 21[e]	24	1.00
GCAG[g]	17.7[g]	6.28*e* − 13[g]	26	1.00
AGAA	17.4	4.83*e* − 07	27	1.00
GGAA	17.1	9.00*e* − 05	28	1.00

*Tetramers common in all three samples have the same letters.

and to obtain a comprehensive view of posttranscriptional regulatory networks.

5. Conclusions

In this study we presented the first systematic comparison of different strategies to implement major CLIP-Seq data analysis steps. Our findings can serve as the practical guideline for CLIP experiments design and the comprehensive analysis of CLIP-Seq data.

Conflict of Interests

The authors declare no competing financial interests.

Authors' Contribution

Yu Shyr and Anil K. Rustgi led the project and oversaw the analysis. Blair B. Madison designed and performed the experiments. Qi Liu and Xue Zhong conducted data analysis. Qi Liu wrote the paper. All authors have read and approved the final paper.

Acknowledgments

This work was supported in part by National Cancer Institute Grants U01 CA163056, P50 CA095103, P50 CA098131, and P30 CA068485 (to Yu Shyr) and in part by NIH R01DK056645 (to Blair B. Madison and Anil K. Rustgi), NIH K01DK093885 (to Blair B. Madison), The Hansen Foundation (to Anil K. Rustgi), National Colon Cancer Research Alliance (to Anil K. Rustgi), and NIH/NIDDK P30DK050306 Center for Molecular Studies in Digestive and Liver Diseases (Molecular Biology/Gene Expression Facility).

References

[1] T. Glisovic, J. L. Bachorik, J. Yong, and G. Dreyfuss, "RNA-binding proteins and post-transcriptional gene regulation," *FEBS Letters*, vol. 582, no. 14, pp. 1977–1986, 2008.

[2] K. B. Jensen and R. B. Darnell, "CLIP: crosslinking and immunoprecipitation of in vivo RNA targets of RNA-binding proteins," *Methods in Molecular Biology*, vol. 488, pp. 85–98, 2008.

[3] J. Ule, K. Jensen, A. Mele, and R. B. Darnell, "CLIP: a method for identifying protein-RNA interaction sites in living cells," *Methods*, vol. 37, no. 4, pp. 376–386, 2005.

[4] J. Ule, K. B. Jensen, M. Ruggiu, A. Mele, A. Ule, and R. B. Darnell, "CLIP identifies Nova-regulated RNA networks in the brain," *Science*, vol. 302, no. 5648, pp. 1212–1215, 2003.

[5] S. Guil and J. F. Cáceres, "The multifunctional RNA-binding protein hnRNP A1 is required for processing of miR-18a," *Nature Structural and Molecular Biology*, vol. 14, no. 7, pp. 591–596, 2007.

[6] J. R. Sanford, P. Coutinho, J. A. Hackett, X. Wang, W. Ranahan, and J. F. Caceres, "Identification of nuclear and cytoplasmic mRNA targets for the shuttling protein SF2/ASF," *PLoS ONE*, vol. 3, no. 10, Article ID e3369, 2008.

[7] P. Becht, J. König, and M. Feldbrügge, "The RNA-binding protein Rrm4 id essential for polarity in *Ustilago maydis* and shuttles along microtubules," *Journal of Cell Science*, vol. 119, no. 23, pp. 4964–4973, 2006.

[8] R. S. Daughters, D. L. Tuttle, W. Gao et al., "RNA gain-of-function in spinocerebellar ataxia type 8," *PLoS Genetics*, vol. 5, no. 8, Article ID e1000600, 2009.

[9] E. J. Wurtmann and S. L. Wolin, "A role for a bacterial ortholog of the Ro autoantigen in starvation-induced rRNA degradation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 9, pp. 4022–4027, 2010.

[10] M. Xu, S. Medvedev, J. Yang, and N. B. Hecht, "MIWI-independent small RNAs (MSY-RNAs) bind to the RNA-binding protein, MSY2, in male germ cells," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 30, pp. 12371–12376, 2009.

[11] R. B. Darnell, "HITS-CLIP: panoramic views of protein-RNA regulation in living cells," *Wiley Interdisciplinary Reviews: RNA*, vol. 1, no. 2, pp. 266–286, 2010.

[12] D. D. Licatalosi, A. Mele, J. J. Fak et al., "HITS-CLIP yields genome-wide insights into brain alternative RNA processing," *Nature*, vol. 456, no. 7221, pp. 464–469, 2008.

[13] Y. Xue, Y. Zhou, T. Wu et al., "Genome-wide analysis of PTB-RNA interactions reveals a strategy used by the general splicing repressor to modulate exon inclusion or skipping," *Molecular Cell*, vol. 36, no. 6, pp. 996–1006, 2009.

[14] S. W. Chi, J. B. Zang, A. Mele, and R. B. Darnell, "Argonaute HITS-CLIP decodes microRNA-mRNA interaction maps," *Nature*, vol. 460, no. 7254, pp. 479–486, 2009.

[15] G. W. Yeo, N. G. Coufal, T. Y. Liang, G. E. Peng, X. D. Fu, and F. H. Gage, "An RNA code for the FOX2 splicing regulator revealed by mapping RNA-protein interactions in stem cells," *Nature Structural and Molecular Biology*, vol. 16, no. 2, pp. 130–137, 2009.

[16] J. Spitzer, M. Hafner, M. Landthaler et al., "PAR-CLIP (Photoactivatable Ribonucleoside-Enhanced Crosslinking and Immunoprecipitation): a step-by-step protocol to the transcriptome-wide identification of binding sites of RNA-binding proteins," *Methods in Enzymology*, vol. 539, pp. 113–161, 2014.

[17] M. Hafner, S. Lianoglou, T. Tuschl, and D. Betel, "Genome-wide identification of miRNA targets by PAR-CLIP," *Methods*, vol. 58, no. 2, pp. 94–105, 2012.

[18] M. Ascano, M. Hafner, P. Cekan, S. Gerstberger, and T. Tuschl, "Identification of RNA-protein interaction networks using PAR-CLIP," *Wiley Interdisciplinary Reviews: RNA*, vol. 3, no. 2, pp. 159–177, 2012.

[19] M. Hafner, M. Landthaler, L. Burger et al., "PAR-CliP—a method to identify transcriptome-wide the binding sites of RNA binding proteins," *Journal of Visualized Experiments*, no. 41, Article ID e2034, 2010.

[20] M. Hafner, M. Landthaler, L. Burger et al., "Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP," *Cell*, vol. 141, no. 1, pp. 129–141, 2010.

[21] C. Yao, L. Weng, and Y. Shi, "Global protein-RNA interaction mapping at single nucleotide resolution by iCLIP-Seq," *Methods in Molecular Biology*, vol. 1126, pp. 399–410, 2014.

[22] I. Huppertz, J. Attig, A. D'Ambrogio et al., "iCLIP: protein-RNA interactions at nucleotide resolution," *Methods*, vol. 65, no. 3, pp. 274–287, 2014.

[23] Y. Sugimoto, J. König, S. Hussain et al., "Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein-RNA interactions," *Genome Biology*, vol. 13, no. 8, article R67, 2012.

[24] J. Konig, K. Zarnack, G. Rot et al., "iCLIP—transcriptome-wide mapping of protein-RNA interactions with individual nucleotide resolution," *Journal of Visualized Experiments*, no. 50, Article ID e2638, 2011.

[25] Z. Wang, M. Kayikci, M. Briese et al., "iCLIP predicts the dual splicing effects of TIA-RNA interactions," *PLoS Biology*, vol. 8, no. 10, 2010.

[26] J. König, K. Zarnack, G. Rot et al., "iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution," *Nature Structural & Molecular Biology*, vol. 17, no. 7, pp. 909–915, 2010.

[27] J. Cho, H. Chang, S. C. Kwon et al., "LIN28A is a suppressor of ER-associated translation in embryonic stem cells," *Cell*, vol. 151, no. 4, pp. 765–777, 2012.

[28] Y. Katz, E. T. Wang, E. M. Airoldi, and C. B. Burge, "Analysis and design of RNA sequencing experiments for identifying isoform regulation," *Nature Methods*, vol. 7, no. 12, pp. 1009–1015, 2010.

[29] S. Kishore, L. Jaskiewicz, L. Burger, J. Hausser, M. Khorshid, and M. Zavolan, "A quantitative analysis of CLIP methods for identifying binding sites of RNA-binding proteins," *Nature Methods*, vol. 8, no. 7, pp. 559–564, 2011.

[30] A. K. Leung, A. G. Young, A. Bhutkar et al., "Genome-wide identification of Ago2 binding sites from mouse embryonic stem cells with and without mature microRNAs," *Nature Structural & Molecular Biology*, vol. 18, no. 2, pp. 237–244, 2011.

[31] B. B. Madison, Q. Liu, X. Zhong et al., "LIN28B promotes growth and tumorigenesis of the intestinal epithelium via Let-7," *Genes and Development*, vol. 27, no. 20, pp. 2233–2245, 2013.

[32] M. Polymenidou, C. Lagier-Tourenne, K. R. Hutt et al., "Long pre-mRNA depletion and RNA missplicing contribute to neuronal vulnerability from loss of TDP-43," *Nature Neuroscience*, vol. 14, no. 4, pp. 459–468, 2011.

[33] M. L. Wilbert, S. C. Huelga, K. Kapeli et al., "LIN28 binds messenger RNAs at GGAGA motifs and regulates splicing factor abundance," *Molecular Cell*, vol. 48, no. 2, pp. 195–206, 2012.

[34] C. Zhang, M. A. Frias, A. Mele et al., "Integrative modeling defines the nova splicing-regulatory network and its combinatorial controls," *Science*, vol. 329, no. 5990, pp. 439–443, 2010.

[35] V. Murigneux, J. Saulière, H. Roest Crollius, and H. Le Hir, "Transcriptome-wide identification of RNA binding sites by CLIP-seq," *Methods*, vol. 63, no. 1, pp. 32–40, 2013.

[36] T. Wang, B. Chen, M. Kim, Y. Xie, and G. Xiao, "A model-based approach to identify binding sites in CLIP-seq data," *PLoS ONE*, vol. 9, no. 4, Article ID e93248, 2014.

[37] B. Chen, J. Yun, M. S. Kim, J. T. Mendell, and Y. Xie, "PIPE-CLIP: a comprehensive online tool for CLIP-seq data analysis," *Genome Biology*, vol. 15, no. 1, article R18, 2014.

[38] S. Althammer, J. González-vallinas, C. Ballaré, M. Beato, and E. Eyras, "Pyicos: a versatile toolkit for the analysis of high-throughput sequencing data," *Bioinformatics*, vol. 27, no. 24, Article ID btr570, pp. 3333–3340, 2011.

[39] P. J. Uren, E. Bahrami-Samani, S. C. Burns et al., "Site identification in high-throughput RNA-protein interaction data," *Bioinformatics*, vol. 28, no. 23, pp. 3013–3020, 2012.

[40] A. Kucukural, H. Özadam, G. Singh, M. J. Moore, and C. Cenik, "ASPeak: an abundance sensitive peak detection algorithm for RIP-Seq," *Bioinformatics*, vol. 29, no. 19, pp. 2485–2486, 2013.

[41] J. König, K. Zarnack, N. M. Luscombe, and J. Ule, "Protein-RNA interactions: new genomic technologies and perspectives," *Nature Reviews Genetics*, vol. 13, no. 2, pp. 77–83, 2012.

[42] C. Trapnell, A. Roberts, L. Goff et al., "Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks," *Nature Protocols*, vol. 7, no. 3, pp. 562–578, 2012.

[43] S. Heinz, C. Benner, N. Spann et al., "Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities," *Molecular Cell*, vol. 38, no. 4, pp. 576–589, 2010.

[44] E. G. Moss, R. C. Lee, and V. Ambros, "The cold shock domain protein LIN-28 controls developmental timing in C. elegans and is regulated by the lin-4 RNA," *Cell*, vol. 88, no. 5, pp. 637–646, 1997.

[45] S. Peng, L.-L. Chen, X.-X. Lei et al., "Genome-wide studies reveal that Lin28 enhances the translation of genes important for growth and survival of human embryonic stem cells," *Stem Cells*, vol. 29, no. 3, pp. 496–504, 2011.

[46] J. E. Thornton and R. I. Gregory, "How does Lin28 let-7 control development and disease?" *Trends in Cell Biology*, vol. 22, no. 9, pp. 474–482, 2012.

[47] C. E. King, L. Wang, R. Winograd et al., "LIN28B fosters colon cancer migration, invasion and transformation through let-7-dependent and-independent mechanisms," *Oncogene*, vol. 30, no. 40, pp. 4185–4193, 2011.

[48] I. Heo, C. Joo, Y.-K. Kim et al., "TUT4 in concert with Lin28 suppresses microRNA biogenesis through pre-microRNA uridylation," *Cell*, vol. 138, no. 4, pp. 696–708, 2009.

[49] Y. Nam, C. Chen, R. I. Gregory, J. J. Chou, and P. Sliz, "Molecular basis for interaction of *let-7* microRNAs with Lin28," *Cell*, vol. 147, no. 5, pp. 1080–1091, 2011.

[50] S. Ohno, "Universal rule for coding sequence construction: TA/CG deficiency-TG/CT excess," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 85, no. 24, pp. 9630–9634, 1988.

[51] C. Zhang and R. B. Darnell, "Mapping in vivo protein-RNA interactions at single-nucleotide resolution from HITS-CLIP data," *Nature Biotechnology*, vol. 29, no. 7, pp. 607–614, 2011.

[52] Y. Sugimoto, J. König, S. Hussain et al., "Analysis of CLIP and iCLIP methods for nucleotide-resolution studies of protein-RNA interactions," *Genome Biology*, vol. 13, no. 8, article R67, 2012.

[53] D. Ray, H. Kazan, K. B. Cook et al., "A compendium of RNA-binding motifs for decoding gene regulation," *Nature*, vol. 499, no. 7457, pp. 172–177, 2013.

[54] B. Kechavarzi and S. C. Janga, "Dissecting the expression landscape of RNA-binding proteins in human cancers," *Genome Biology*, vol. 15, no. 1, article R14, 2014.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 491502, 9 pages
http://dx.doi.org/10.1155/2015/491502

Research Article

Classification of Cancer Primary Sites Using Machine Learning and Somatic Mutations

Yukun Chen,[1] Jingchun Sun,[2] Liang-Chin Huang,[2] Hua Xu,[2] and Zhongming Zhao[1,3,4]

[1]*Department of Biomedical Informatics, Vanderbilt University School of Medicine, Nashville, TN 37203, USA*
[2]*School of Biomedical Informatics, The University of Texas Health Science Center at Houston, Houston, TX 77030, USA*
[3]*Department of Psychiatry, Vanderbilt University School of Medicine, Nashville, TN 37212, USA*
[4]*Department of Cancer Biology, Vanderbilt University School of Medicine, Nashville, TN 37232, USA*

Correspondence should be addressed to Hua Xu; hua.xu@uth.tmc.edu and Zhongming Zhao; zhongming.zhao@vanderbilt.edu

Received 21 October 2014; Revised 5 February 2015; Accepted 19 February 2015

Academic Editor: Federico Ambrogi

An accurate classification of human cancer, including its primary site, is important for better understanding of cancer and effective therapeutic strategies development. The available big data of somatic mutations provides us a great opportunity to investigate cancer classification using machine learning. Here, we explored the patterns of 1,760,846 somatic mutations identified from 230,255 cancer patients along with gene function information using support vector machine. Specifically, we performed a multiclass classification experiment over the 17 tumor sites using the gene symbol, somatic mutation, chromosome, and gene functional pathway as predictors for 6,751 subjects. The performance of the baseline using only gene features is 0.57 in accuracy. It was improved to 0.62 when adding the information of mutation and chromosome. Among the predictable primary tumor sites, the prediction of five primary sites (large intestine, liver, skin, pancreas, and lung) could achieve the performance with more than 0.70 in *F*-measure. The model of the large intestine ranked the first with 0.87 in *F*-measure. The results demonstrate that the somatic mutation information is useful for prediction of primary tumor sites with machine learning modeling. To our knowledge, this study is the first investigation of the primary sites classification using machine learning and somatic mutation data.

1. Introduction

Cancer is a complex disease, which is driven by the combination of genetic, environmental, and lifestyle factors. Among these factors, the combination of multiple genes driving cancer development varies considerably among cancer types and patients [1]. During the past decade, investigation of mutations at both large-scale and specific loci has been made in order to increase our knowledge of the molecular heterogeneity in this complex disease. Notably, several large-scale, network-based cancer genome projects have generated multidimensional and genome-wide data. These projects include The Cancer Genome Atlas (TCGA) [2], Wellcome Trust Sanger Institute's Cancer Genome Project [3], and the International Cancer Genome Consortium (ICGC) [4]. These projects have dramatically advanced cancer research, especially in its genetics and genomics [5]. A cancer somatic mutation landscape, primarily focusing on nucleotide change patterns (e.g., C->T) and mutation signatures in the cancer genomes, has been released to the community [6]. Among these achievements, some have been translated into molecular diagnosis, better prognosis, and new targeted therapies. For example, the germline mutations in *BRCA1* and *BRCA2* confer high risks to breast and ovarian cancers [7]. Their genotyping is used to determine susceptibility to breast and ovarian cancer [8–10]. To monitor the treatment, the increased expression level of circulating tumor marker, human epidermal growth factor receptor 2 (HER2), is used to determine the treatment of a monoclonal antibody trastuzumab in breast cancer [11–13]. However, cancer is strongly heterogeneous, and the cancer classification is a critical first step in the further investigation of the pathology of cancer and the development of effective treatments.

For cancer classification, the fundamental method is mainly based on the cell of origin or their histological types [14]. During the last two decades, molecular profiling has been unveiled for classification of cancer types and subtypes, as well as assessment of heterogeneity of cancer samples [15]. For example, in breast cancer, recent studies that are mainly based on microarray-based gene expression data and unbiased hierarchical clustering have identified several molecular subtypes: basal-like, ErbB2$^+$, normal breast-like, luminal subtype A, and luminal subtype B [16, 17]. Further gene expression profiling was found to be effective on identifying even more specific subtypes in triple negative breast cancer type [18]. As massive amount of genomic, transcriptomic, and proteomic data in cancer cells and patients becomes available, an integrated model of cancer classification was recently proposed to capture the known attributes of cancer by integrating morphology, cancer stem cells, proteomics, and genomics [19]. However, as other data integration schemes, it presents a big challenge to develop an effective and comprehensive method for cancer classification.

Recently, next-generation sequencing approaches have been applied to cancer studies, including whole genome sequencing, whole exome sequencing, targeted gene sequencing, whole transcriptome sequencing, genome-wide microRNA sequencing, and epigenomics, providing the highest resolution (base-pair resolution) of genetic and genomic information in cancer. These datasets provide us an unprecedented opportunity on systematic and integrated investigation of molecular mechanisms of cancer. For example, Vogelstein et al. systematically analyzed the mutation landscapes in 96 cancer types reported from 127 publications, providing deep insights into the cancer genomic architecture [20]. Among these datasets, somatic mutation data in cancer genomes has been accumulated dramatically, which makes it possible to discover novel cancer genes and mutations [21–23], draw mutational landscapes among multiple cancers [6, 24], and explore the molecular mechanisms of tumorigenesis [25]. In this study, we hypothesized that features from the massive amount of somatic mutations could act as effective contributors for cancer site classification. Moreover, another goal of the study is to search for the associations between cancer sites and mutation features in a larger scale using machine learning.

In this study, we proposed a novel cancer site classification framework by investigating somatic mutations through machine learning approaches. The somatic mutation information includes (1) patient information, (2) mutation-associated genes, and (3) mutation-associated chromosomes. We extracted these types of information from the database COSMIC (Catalogue of Somatic Mutations In Cancer) [26]. We further integrated the mutation-associated gene function using gene pathways from the database KEGG (Kyoto Encyclopedia of Genes and Genomes) [27]. Our evaluation showed that the combination of the somatic mutation, mutation-associated gene, and mutation-associated chromosome features achieved the best performance of cancer site classification.

2. Methods and Materials

2.1. Overview of Study Design. The main purpose of this study is to test if the somatic mutation features and mutation-related information are useful or have the power to predict the primary cancer site since more than a million somatic mutations in cancer genomes have been reported, collected, and systematically analyzed. To address this important question, we took advantage of the data in COSMIC, which is the most comprehensive, annotation-based database for the somatic mutations from numerous patients with cancer type information. Figure 1 illustrates the study design.

2.2. Data Sources. The COSMIC database is established to collect, store, and display somatic mutations and related information extracted from the primary literature on human cancers as well as those identified from cancer genome projects [26]. The COSMIC data provides a consistent view of histology and tissue ontology with the mutation information. We downloaded the data from COSMIC website on April 18, 2014. The downloaded data contained 990,529 samples, 25,660 genes, 1,292,597 coding mutations, 1,528,225 noncoding variations, and 11,330 references.

To normalize the gene names to the gene official symbols, we took a two-step strategy. First, we utilized the mutation positions from COSMIC data to map the gene regions using the UCSC Genome Browser based on the GRCh37 genome annotation [28]. Thus, we obtained three sets of gene names: (1) gene names without position information in COSMIC; (2) gene names with position information in COSMIC but could not be matched to the UCSC Genome Browser; and (3) gene names with the matched information (gene names and locations) in the UCSC Genome Browser. Finally, we utilized the Entrez Gene Table to match these gene names to their corresponding official gene symbols [29].

To clean the data, we removed the records that do not have the information about gene name, sample ID, primary site, or mutation description. Additionally, we removed the mutations that were involved in fusion genes because they do not have a single-mutation position. Eventually, the filtered dataset contained 230,255 patients, 22,111 unique genes, and 1,760,846 mutations.

KEGG pathway database manually collects and annotates the molecular interactions and regulations among genes and then draws pathway maps [27]. We downloaded the data on May 21, 2014, from website (http://www.kegg.jp/kegg/). We extracted the genes from their involved pathways. In total, there are 285 human pathways and 6,503 genes involved in 22,573 pathway-gene relationships. Then, we matched the mutation-associated genes into the pathways.

2.3. Datasets and Features. In this study, we mainly explored the somatic mutations and their relative information for cancer primary site classification. From the filtered data obtained above, we extracted 7,251 patients who had at least ten mutations. Patients with a very small number of mutations would be more likely outliers in the dataset and fail to provide sufficient information for a model to distinguish the final label with other patients. These limitations increase

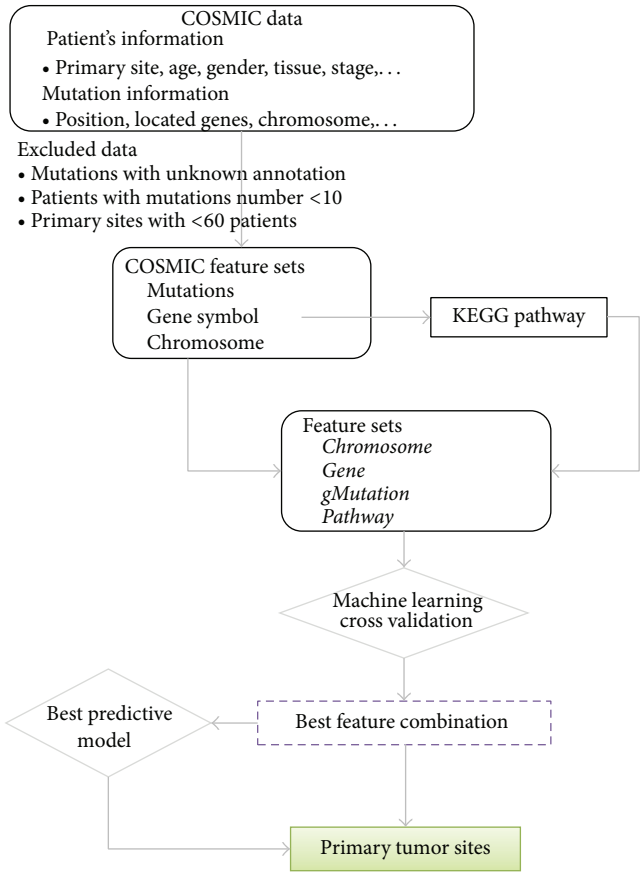


Figure 1: Study design using somatic mutations to classify primary tumor sites by machine learning model. In order to precisely represent the mutations, we generated a feature *gMutation* by binding mutations with their corresponding gene symbols.

Table 1: Distribution of primary tumor sites.

Primary tumor site	Number of patients	Percentage (%)
Lung	970	14.43
Breast	967	14.39
Large intestine	654	9.73
Haematopoietic and lymphoid tissue	644	9.58
Kidney	491	7.31
Ovary	490	7.29
Liver	400	5.95
Central nervous system	377	5.61
Prostate	374	5.56
Endometrium	261	3.88
Pancreas	252	3.75
Autonomic ganglia	222	3.30
Skin	184	2.74
Oesophagus	174	2.59
Urinary tract	110	1.64
Upper aerodigestive tract	91	1.35
Stomach	60	0.89

the difficulty in training a good predictive model. On the other hand, patients with a larger number of mutations more likely have common features and thus induce better training to find a more reliable pattern in the model. We chose ten as the threshold because the filtered patients set of over seven thousand is large enough for machine learning experiments and the number of features generated for each patient based on the threshold of ten does not discourage the modeling process.

We further filtered out several minority classes of primary tumor sites. Each of them has less than 60 patients in the dataset, such as "Bone," "Meninges," and "Eye." Thus, the final set of 6,751 patients was chosen to be used in this study. These patients were diagnosed to be one type of cancer among the 17 primary tumor sites. Table 1 shows the distribution of the patients with the primary tumor sites.

From the COSMIC data, we collected mutations and their corresponding mutated genes and chromosomes to represent the genetic characteristics of each patient. As a result, our process led to twelve unique types into four categories (e.g., substitution, insertion, deletion, and complex) and eight more specific descriptions (e.g., substitution-nonsense, substitution-missense, substitution-coding silent, substitution-intronic, insertion-in frame, insertion-frameshift, deletion-frameshift, and deletion-frameshift) according to the mutation description in the COSMIC and our filtering procedure. Table 2 includes their detailed descriptions. In our dataset, these mutations could be mapped to 21,286 unique genes in all patients.

Instead of directly using individual mutation description, we bound them with their corresponding gene symbols to precisely represent the mutations. It resulted in 79,865 unique combos of gene symbols and mutation descriptions in the dataset, such as "CHDC2_Insertion-Frameshift," "SPEN_Complex," and "SP1_Substitution-Missense." In this paper, we use "*gMutation*" to represent the feature set of mutations associated with genes. In our study design, we considered *gene* symbol and *gMutation* as two different features. *Gene* symbol feature represents a larger range of biological activity at the gene level while the *gMutation* feature represents a more precise feature at the mutation level located in a specific gene region. Despite the fact that both features are not independent, they could represent cancer patients at two different levels. Thus, we utilized them together in the prediction modeling.

Since the human somatic mutation landscape is related to chromosome [30], we further considered the *Chromosome* as the third feature in our study. The human genome includes 22 autosomes (1–22), two sex chromosomes (X, Y), and one mitochondrial genome (MT). Thus, there are a total of 25 features included in the *Chromosome* feature set.

Besides the mutation-related information, we further integrated the KEGG dataset to provide the functional knowledge of the genes involved in the patients' mutations. There are 285 unique pathways for the 21,286 genes.

Table 2: Mutation description.

Mutation description	Definition
Substitution	A mutation involving the substitution of a single nucleotide
Substitution-nonsense	A substitution mutation resulting in a termination codon, foreshortening the translated peptide
Substitution-missense	A substitution mutation resulting in an alternate codon, altering the amino acid at this position only
Substitution-coding silent	A synonymous substitution mutation which encodes the same amino acid as the wild type codon
Substitution-intronic	A substitution mutation outside the coding domains; no interpretation is made as to its effect on splice sites or nearby regulatory regions
Insertion	An insertion of novel sequence into the gene
Insertion-in frame	An insertion of nucleotides which does not affect the gene's translation frame, leaving the downstream peptide sequence intact
Insertion-frameshift	An insertion of novel sequence which alters the translation frame, changing the downstream peptide sequence (often resulting in premature termination)
Deletion	A deletion of a portion of the gene's sequence
Deletion-in frame	A deletion of nucleotides which does not affect the gene's translation frame, leaving the downstream peptide sequence intact
Deletion-frameshift	A deletion of nucleotides which alters the translation frame, changing the downstream peptide sequence (often resulting in premature termination)
Complex	A compound mutation which may involve multiple insertions, deletions, and substitutions

Therefore, in this study, we defined four features: *Gene*, *gMutation*, *Chromosome*, and *Pathway*. Furthermore, we attempted to find the optimal combination of these four feature sets for the best prediction performance using the *Gene* feature as the baseline.

2.4. Machine Learning Experiments. In the data we collected, each sample contains an array of features that are present in one patient. We present all the collected features in all patients as a feature vector in the machine learning fashion. All features in the vector were represented by binary values; namely, "1" represents present while "0" represents not present. Then, we constructed a data matrix, in which each row includes all the features for one patient while each column includes one type of feature for all patients.

With respect to the classification method, we implemented a one-versus-all multiclass classification schema to identify the primary tumor site based on patients' mutation-associated features and the gene pathway feature. For each primary tumor site, we trained a binary classifier that could distinguish the class belonging to the site versus the one that does not. Each classifier was a support vector machine (SVM) with linear kernel implemented by LIBLINEAR [31]. Given the 17 trained binary classifiers, we predicted the primary tumor site for an undiagnosed patient to be a class from the corresponding classifier with the highest confidence value, which is the distance to the hyperplane from the trained SVM. For the experimental parameter set in LIBLINEAR, we used "L1-regularized L2-loss support vector classification" as the solver for the multiclass classification task. L1-regularization was selected because the gene mutation based feature set is large (>100,000 features among <7,000 samples) and sparse (very few nonzero entries in the data matrix).

We performed the multiclass classification experiments on the *Gene* feature (baseline) and six different combinations of four feature sets in the fashion of 10-fold cross validation (see Section 2.5). To avoid being overoptimistic on the modeling, we did not optimize the parameter set of the linear SVM model. That is, the parameter set (L1-regularization, L2-loss function, and cost = 10, among others) was fixed through the entire 10-fold cross validation experiments. We chose the combination of feature sets with the best performance in accuracy for the generation of our best predictive model. Then, we applied the best model to predict the primary tumor site over 17 cancer candidates. The performance over each primary tumor site was evaluated by precision, recall, and F-measure.

2.5. Evaluation. We conducted experiments by 10-fold cross validation. All patient samples were split into ten folds with stratification so that the class distribution in each is much similar to the one from the original dataset. We alternately treated one fold as the test set and the other as the training set. Then we did the predictive model training and testing 10 times. Eventually, each patient would have a diagnosis of the primary tumor site by the predictive model. We computed the accuracy as global metric to evaluate different feature combinations. We also evaluated the performance of prediction on each primary tumor site by precision, recall, and F-measure:

$$\begin{aligned}
\text{Accuracy} &= \frac{\sum_{y_i} \text{TP}(y_i)}{\sum_{y_i} \text{Pred}(y_i)}, \\
\text{Precision}(y_i) &= \frac{\text{TP}(y_i)}{\text{Pred}(y_i)}, \\
\text{Recall}(y_i) &= \frac{\text{TP}(y_i)}{\text{True}(y_i)}, \\
F\text{-measure}(y_i) &= \frac{2 * \text{Precision}(y_i) * \text{Recall}(y_i)}{\text{Precision}(y_i) + \text{Recall}(y_i)},
\end{aligned} \tag{1}$$

Table 3: Micro- and macroaveraged accuracies of seven combinations of gene symbols with three other features.

Feature combination	Number of features	miAccuracy	maAccuracy (mean)	maAccuracy (SD)
Gene (baseline)	21,286	0.57	0.57	0.019
Gene + *gMutation*	101,151	0.58	0.58	0.019
Gene + *Pathway*	21,571	0.58	0.58	0.010
Gene + *Chromosome*	21,311	0.60	0.60	0.022
Gene + *gMutation* + *Pathway*	101,436	0.60	0.60	0.013
Gene + *gMutation* + *Chromosome*	101,176	0.62	0.62	0.021
Gene + *gMutation* + *Chromosome* + *Pathway*	101,461	0.60	0.60	0.015

Note: miAccuracy represents the microaverage accuracy; maAccuracy represents the macroaverage accuracy, which is reported in mean and standard deviation (SD) over 10 accuracies from 10-fold cross validation.

where y_i is one of the given primary tumor sites or classes; $\mathrm{TP}(y_i)$ is the number of true positives of the given class y_i predicted by the model; $\mathrm{Pred}(y_i)$ is the number of the predictions of the given class; $\mathrm{True}(y_i)$ is the number of true positives of the given class in the dataset.

We also used microaverage and macroaverage methods to report the accuracy. In the microaverage accuracy (miAccuracy), $\mathrm{TP}(y_i)$ in the numerator is the summation of true positives of a given class over tenfold and the denominator is equivalent to the number of total patients. In the macroaverage accuracy (maAccuracy), we generated the accuracy for each and reported the mean and standard deviation (SD) over 10-fold results.

3. Results

Following the study design in Figure 1, we first rigorously filtered the data at the mutation, patient, and tumor site levels to reduce the data noises and improve the predictive performance. Thus, among the 990,529 samples in the downloaded data, we only recruited 6,751 patients in our study. To test if the higher-level functional knowledge is useful to improve the performance, we integrated the gene pathway information into the feature set. Then, we identified the best feature combination by cross validation. Finally, based on the best feature combination, we developed one best predictive model set and applied it to predict the primary tumor sites.

3.1. Identification of the Best Feature Combination. We have trained seven predictive models using different combinations of feature sets. The specific features for each combination, sizes of features, and the accuracies as their global scores are shown in Table 3. We considered the predictive model using *Gene* feature only as our baseline, which achieved 0.57 in accuracy. With one additional feature set (*gMutation*, *Chromosome*, or *Pathway*), the model achieved slightly better (0.58, 0.58, and 0.60, resp.). If we combined three types of feature sets, the model reached the best performance (0.62) when features *Gene*, *gMutation*, and *Chromosome* were combined. However, when we added *Pathway* as the fourth feature set, the accuracy dropped back to 0.60. We also tested other combinations, but none of them had better achievement than the best model (data not shown).

3.2. Prediction of Primary Tumor Site. Using the best model set, we predicted the whole dataset using 10-fold cross validation and evaluated the performance on every primary tumor site by precision, recall, and F-measure. Table 4 shows the performance of the best predictive model set using the combination of three features (*Gene* + *gMutation* + *Chromosome*) over each tumor site.

The average precision and recall were 0.70 and 0.49, respectively. This predictive model could achieve the precision of 0.75 or higher in 8 out of 17 primary tumor sites, recall of 0.60 or higher for 8 out of 17, and F-measure of 0.60 or higher for 9 out of 17.

4. Discussion

In this study, we performed a systematic exploration of the somatic mutations and their related features for cancer classification using a machine learning approach and the most comprehensive somatic mutation dataset so far. The study filtered the somatic mutation data from COSMIC, identified the best feature combination, and predicted the primary tumor sites using the machine learning methods.

Machine learning approaches have been applied to cancer prognosis and prediction [32]. In our study, the performance of primary tumor site prediction is strongly correlated with its sample size (correlation coefficient = 0.58). Therefore, increasing the sample size could be a major way to improve the performance. However, for some specific sites, this is not always true. For example, the primary tumor site "skin" only contains 2.74% samples in the dataset and ranked the 13th over the 17 primary sites studied based on the sample percentage in this study, but its model ranked 3rd in F-measure (0.73). The primary site "Lung" has the largest percentage of samples, but it was ranked 5th in F-measures. To discover the underlying reason for this observation, we further computed the coverage rate of the genes that occurred in the true positives identified by the predictive model for each primary tumor site. The coverage rate of a gene X in a primary tumor site is the ratio between the counts of the true positives where the gene X occurred and the total number of true positives. The top four primary tumor sites in prediction (large intestine, liver, skin, and pancreas) share the pattern of "Top Heavy" in coverage rate distribution (with max coverage rate over 50%), while "Lung" has distribution over genes

Table 4: Precision, recall, and F-measure for the best predictive model using "*Gene*," "*gMutation*," and "*Chromosome*" on each primary tumor site.

Primary tumor site	Precision	Recall	F-measure
Large intestine	0.88	0.85	0.87
Liver	0.88	0.72	0.79
Skin	0.91	0.61	0.73
Pancreas	0.75	0.67	0.71
Lung	0.66	0.75	0.70
Endometrium	0.91	0.52	0.67
Kidney	0.72	0.62	0.66
Haematopoietic and lymphoid tissue	0.50	0.75	0.60
Breast	0.50	0.75	0.60
Central nervous system	0.63	0.51	0.56
Ovary	0.40	0.49	0.44
Prostate	0.46	0.35	0.40
Autonomic ganglia	0.45	0.28	0.34
Oesophagus	0.81	0.20	0.31
Urinary tract	0.83	0.09	0.16
Upper aerodigestive tract	1.00	0.05	0.10
Stomach	0.60	0.05	0.09

closer to uniform (max coverage rate of 16%). Therefore, without as many as relatively strong associated genes, it is harder to predict "Lung" than these top four primary sites, although "Lung" has the most number of training samples.

For the bottom four primary sites with the smallest sample size, the performance by the model tended to be poorest. Specifically, "Oesophagus," "Urinary tract," "Upper aerodigestive tract," and "Stomach" had smallest numbers of samples, and they were also ranked at the bottom according to F-measure values. For those primary tumor sites with a large number of samples but without excellent prediction performance (e.g., "Lung," "Breast," "Haematopoietic and lymphoid tissue"), they had a much better recall (all 0.75) than others, but poor precision (0.66, 0.50, and 0.50, resp.).

One important output of this study is the best feature combination (*Gene* + *gMutation* + *Chromosome*) compared to other combinations. Though the three features were directly related to mutation feature, they reflected three features at differently genetic architecture at three levels, namely, DNA-sequence, DNA function, and DNA organization. This observation indicated that, with more detailed information on mutation, the best combination could contribute the cancer class classification. The result illustrated that the somatic mutation could be used to predict primary tumor sites in the individual way or the integrative way.

To test if the high-level function-associated features could improve the performance of cancer site classification, we explored the KEGG pathway that mutation-associated genes are involved in. However, in our study, there is no improvement of performance by integrating the *Pathway* feature into other features. One possible reason is that a gene can be involved in multiple pathways; this is especially true for cancer genes, which have important function and regulation in biological system and often involve in multiple signaling pathways. If only one pathway had a high association with one cancer type, the additional pathways could lead to the noise for the prediction of such cancer type. Moreover, the *Pathway* feature increased the dimensions of our feature space rather than refined our predictors. Finally, pathway size varies greatly, but this characteristic was not taken into account in the pathway analysis in this study. We would use a better way to represent the KEGG pathway feature set. Instead of using binary (zero or one) representation, for example, we could use quantitative value between zero and one to represent the involvement of the mutated genes in the pathway so that the pathways with higher number of mutated genes involved would have higher weights as predictors.

Our prediction model utilized the *Gene* feature as the baseline. Table 5 summarizes the genes that have been used in the model. The number of genes used in the modeling varied greatly, which might be one reason that the performance for multiple primary sites is much different.

Among the 17 primary tumor sites, five primary tumor sites achieved better performance, according to their F-measure values (>0.70). They are "large intestine," "liver," "skin," "pancreas," and "lung." To illustrate the common and specific genes in these five tumor sites, we selected the top 50 genes according to the counts of genes that occurred in the true positive patients identified by the model for each primary tumor site. Figure 2 shows the overlap among the five sets of the genes in five primary sites. The number of common genes among the five primary sites is different, which might reflect their histological relationship among them. For example, the "large intestine" has 25 common genes to "skin," 20 common genes to "pancreas," 14 common genes to "lung," and 7 common genes to "liver." Notably, there are only 2 (*TTN* and *LRP1B*) common genes among the five sets of the genes. Searching the COSMIC (version 69) dataset, the gene *TTN* has 3,403 mutations in the unique 1,881 samples. However, only 17 mutations have been reported in more than three samples. This gene is the longest human gene, and its cancer risk remains unclear [33, 34]. The gene *LRP1B*, which encodes one of the low density lipoproteins (LDL), is reported as a novel candidate tumor suppressor gene [35]. It has 1,302 mutations in the unique 939 samples. Only two mutations have been reported in more than three samples. Besides the common genes, each primary site has its own mutation-associated genes. It will be useful further to check them for further understanding of their genetic architectures.

In this exploratory study, we demonstrated that the somatic mutation information could be used for cancer classification. As the first attempt for prediction of cancer sites, we have seen many opportunities to improve the performance based on the genetic and genomic information in future work. First, refinement of the features might improve the performance of machine learning experiments in several ways. (1) The first is identification and analysis of the most frequently mutated genes across multiple primary sites. (2) The second is reducing redundancy of feature sets by automatic dimension reduction techniques. We can use two

Table 5: Summary of genes and samples used in the primary tumor site prediction.

Primary tumor site	Number of genes	Number of true positives
Large intestine	18,066	555
Liver	19,778	287
Skin	10,898	113
Pancreas	3,364	170
Lung	18,423	724
Endometrium	18,234	137
Kidney	10,601	302
Haematopoietic and lymphoid tissue	14,545	723
Breast	6,327	486
Central nervous system	2,773	192
Ovary	8,169	238
Prostate	5,875	132
Autonomic ganglia	1,425	62
Oesophagus	6,200	34
Urinary tract	3,288	10
Upper aerodigestive tract	1,013	5
Stomach	86	3

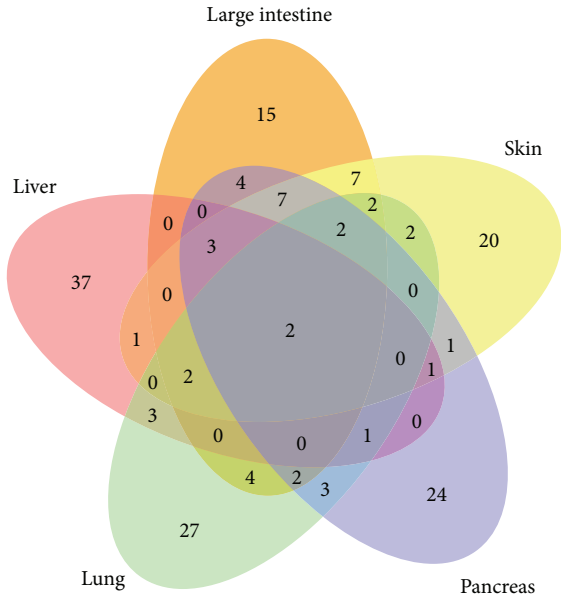


Figure 2: Comparison among five sets of the top 50 genes used in the machine learning modeling for five primary tumor sites (large intestine, liver, lung, pancreas, and skin).

types of methods. One is the algorithms without the label information, such as, principle component analysis [36], latent Dirichlet allocation [37], and sufficient dimension reduction [38]. Another is the algorithm using the label information including HITON [39] and random forest. (3) The third is normalization of *Gene* feature by the gene length. For example, the gene *TTN* is the longest known coding gene and thus might accrue mutations by chance that happened in many tumor sites. (4) The fourth is integrating more molecular features (e.g., methylation, gene expression, and gene direct interacting relationship). And (5) considering specific functional mutations or mutational features may improve prediction power. For example, mutations of specific nucleotide change (e.g., C->T in melanoma), or mutations causing critical amino acid changes or protein structure alterations, will likely be more informative. Second, adding information from normal subjects in our modeling might boost the performance. Adding the normal subjects as a control group to the modeling process is more applicable on a clinical perspective. Finally, applying multiple machine learning algorithms to our task might be robust for evaluation of prediction performance. Many other machine learning methods for cancer classification have been reported. Dettling [40] combined bagging and boosting to precisely classify cancerous malignancies at an early stage using microarray data. Liu et al. [41] also utilized microarray data and machine learning for cancer classification research. Their proposed classifier Recursive Feature Addition with a gene selection method Lagging Prediction Peephole Optimization outperformed popular learning machines such as SVM, Naïve Bayes classifier, and random forest. Hofree et al. [42] introduced a network-based stratification (NBS) algorithm to stratify cancer into informative subtypes by grouping patients together with mutations in similar network regions. They have demonstrated that the identified subtypes are predictive of many clinical outcomes such as patient survival, response to therapy, or tumor histology. We are interested in using the networks with subtype labels generated by NBS to identify the primary tumor site. We plan to design and evaluate better machine learning methods and explore deep learning techniques [43] for better cancer classification with the resource of big data.

5. Conclusion

In conclusion, our application of the machine learning technique to somatic mutations could predict some primary tumor sites, such as the large intestine, liver, skin, pancreas, and lung. Since treatment of cancer does rely on not only the known cancer site, but also the underlying molecular profiles (e.g., cancer driver mutations) and cancer cells migrate to multiple sites at metastasis stage, the prediction of cancer sites based on mutation profiles may be helpful for the enhancement of molecular therapeutics development. This study represents the first large-scale prediction of primary tumor site using comprehensive, publicly available somatic mutations through a machine learning approach.

Conflict of Interests

All the authors declare no conflict of interests.

Authors' Contribution

Yukun Chen and Jingchun Sun contributed equally.

Acknowledgments

This project is partially supported by National Institutes of Health Grants (R01LM011177, P30CA68485, P50CA095103, and P50CA098131), Vanderbilt-Ingram Cancer Center's Breast Cancer SPORE pilot grant (to Zhongming Zhao), Ingram Professorship Funds (to Zhongming Zhao), and Cancer Prevention & Research Institute of Texas (CPRIT R1307) Rising Star Award (to Hua Xu).

References

[1] L. A. Hindorff, E. M. Gillanders, and T. A. Manolio, "Genetic architecture of cancer and other complex diseases: lessons learned and future directions," *Carcinogenesis*, vol. 32, no. 7, pp. 945–954, 2011.

[2] Cancer Genome Atlas Research Network, "Comprehensive genomic characterization defines human glioblastoma genes and core pathways," *Nature*, vol. 455, no. 7216, pp. 1061–1068, 2008.

[3] E. D. Pleasance, R. Keira Cheetham, P. J. Stephens et al., "A comprehensive catalogue of somatic mutations from a human cancer genome," *Nature*, vol. 463, pp. 191–196, 2010.

[4] The International Cancer Genome Consortium, "International network of cancer genome projects," *Nature*, vol. 464, pp. 993–998, 2010.

[5] L. Chin, W. C. Hahn, G. Getz, and M. Meyerson, "Making sense of cancer genomic data," *Genes and Development*, vol. 25, no. 6, pp. 534–555, 2011.

[6] M. S. Lawrence, P. Stojanov, P. Polak et al., "Mutational heterogeneity in cancer and the search for new cancer-associated genes," *Nature*, vol. 499, pp. 214–218, 2013.

[7] R. L. Milne and A. C. Antoniou, "Genetic modifiers of cancer risk for BRCA1 and BRCA2 mutation carriers," *Annals of Oncology*, vol. 22, supplement 1, pp. i11–i17, 2011.

[8] K. E. Malone, J. R. Daling, D. R. Doody et al., "Prevalence and predictors of *BRCA1* and *BRCA2* mutations in a population-based study of breast cancer in White and Black American women ages 35 to 64 years," *Cancer Research*, vol. 66, no. 16, pp. 8297–8308, 2006.

[9] V. M. Basham, J. M. Lipscombe, J. M. Ward et al., "BRCA1 and BRCA2 mutations in a population-based study of male breast cancer," *Breast Cancer Research*, vol. 4, article R2, 2002.

[10] A. H. Trainer, C. R. Lewis, K. Tucker, B. Meiser, M. Friedlander, and R. L. Ward, "The role of *BRCA* mutation testing in determining breast cancer therapy," *Nature Reviews Clinical Oncology*, vol. 7, no. 12, pp. 708–717, 2010.

[11] K. P. Garnock-Jones, G. M. Keating, and L. J. Scott, "Trastuzumab: a review of its use as adjuvant treatment in human epidermal growth factor receptor 2 (HER2)-positive early breast cancer," *Drugs*, vol. 70, pp. 215–239, 2010.

[12] S. Y. Kong, D. H. Lee, E. S. Lee, S. Park, K. S. Lee, and J. Ro, "Serum HER2 as a response indicator to various chemotherapeutic agents in tissue HER2 positive metastatic breast cancer," *Cancer Research and Treatment*, vol. 38, no. 1, pp. 35–39, 2006.

[13] B. S. Sorensen, L. S. Mortensen, J. Andersen, and E. Nexo, "Circulating HER2 DNA after trastuzumab treatment predicts survival and response in breast cancer," *Anticancer Research*, vol. 30, no. 6, pp. 2463–2468, 2010.

[14] H. M. Kvasnicka, "WHO classification of myeloproliferative neoplasms (MPN): a critical update," *Current Hematologic Malignancy Reports*, vol. 8, no. 4, pp. 333–341, 2013.

[15] E. Bair and R. Tibshirani, "Semi-supervised methods to predict patient survival from gene expression data," *PLoS Biology*, vol. 2, no. 4, article e108, 2004.

[16] C. M. Perou, T. Sørile, M. B. Eisen et al., "Molecular portraits of human breast tumours," *Nature*, vol. 406, no. 6797, pp. 747–752, 2000.

[17] T. Sørlie, C. M. Perou, R. Tibshirani et al., "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 10869–10874, 2001.

[18] B. D. Lehmann, J. A. Bauer, X. Chen et al., "Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies," *Journal of Clinical Investigation*, vol. 121, no. 7, pp. 2750–2767, 2011.

[19] H. A. Idikio, "Human cancer classification: a systems biology-based model integrating morphology, cancer stem cells, proteomics, and genomics," *Journal of Cancer*, vol. 2, no. 1, pp. 107–115, 2011.

[20] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz Jr., and K. W. Kinzler, "Cancer genome landscapes," *Science*, vol. 340, no. 6127, pp. 1546–1558, 2013.

[21] M. S. Lawrence, P. Stojanov, C. H. Mermel et al., "Discovery and saturation analysis of cancer genes across 21 tumour types," *Nature*, vol. 505, no. 7484, pp. 495–501, 2014.

[22] P. Jia, W. Pao, and Z. Zhao, "Patterns and processes of somatic mutations in nine major cancers," *BMC Medical Genomics*, vol. 7, article 11, 2014.

[23] P. Jia, Q. Wang, Q. Chen, K. E. Hutchinson, W. Pao, and Z. Zhao, "MSEA: detection and quantification of mutation hotspots through mutation set enrichment analysis," *Genome Biology*, vol. 15, article 489, 2014.

[24] C. Kandoth, M. D. McLellan, F. Vandin et al., "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, 2013.

[25] F. Cheng, P. Jia, Q. Wang, C.-C. Lin, W.-H. Li, and Z. Zhao, "Studying tumorigenesis through network evolution and somatic mutational perturbations in the cancer interactome," *Molecular Biology and Evolution*, vol. 31, no. 8, pp. 2156–2169, 2014.

[26] S. A. Forbes, N. Bindal, S. Bamford et al., "COSMIC: mining complete cancer genomes in the catalogue of somatic mutations in cancer," *Nucleic Acids Research*, vol. 39, no. 1, pp. D945–D950, 2011.

[27] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[28] W. J. Kent, C. W. Sugnet, T. S. Furey et al., "The human genome browser at UCSC," *Genome Research*, vol. 12, no. 6, pp. 996–1006, 2002.

[29] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: gene-centered information at NCBI," *Nucleic Acids Research*, vol. 39, no. 1, pp. D52–D57, 2011.

[30] G. Fudenberg, G. Getz, M. Meyerson, and L. A. Mirny, "High order chromatin architecture shapes the landscape of chromosomal alterations in cancer," *Nature Biotechnology*, vol. 29, no. 12, pp. 1109–1113, 2011.

[31] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[32] J. A. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis," *Cancer Informatics*, vol. 2, pp. 59–77, 2006.

[33] P. Jia and Z. Zhao, "VarWalker: personalized mutation network analysis of putative cancer genes from next-generation sequencing data," *PLoS Computational Biology*, vol. 10, no. 2, Article ID e1003460, 2014.

[34] J. Xia, P. Jia, K. E. Hutchinson et al., "A meta-analysis of somatic mutations from next generation sequencing of 241 melanomas: a road map for the study of genes with potential clinical relevance," *Molecular Cancer Therapeutics*, vol. 13, no. 7, pp. 1918–1928, 2014.

[35] C.-X. Liu, S. Musco, N. M. Lisitsina, S. Y. Yaklichkin, and N. A. Lisitsyn, "Genomic organization of a new candidate tumor suppressor gene, LRP1B," *Genomics*, vol. 69, no. 2, pp. 271–274, 2000.

[36] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, 1986.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[38] A. Globerson and N. Tishby, "Sufficient dimensionality reduction," *Journal of Machine Learning Research*, vol. 3, pp. 1307–1331, 2003.

[39] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 171–234, 2010.

[40] M. Dettling, "BagBoosting for tumor classification with gene expression data," *Bioinformatics*, vol. 20, no. 18, pp. 3583–3593, 2004.

[41] Q. Liu, A. H. Sung, Z. Chen et al., "Gene selection and classification for cancer microarray data based on machine learning and similarity measures," *BMC Genomics*, vol. 12, supplement 5, article S1, 2011.

[42] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based stratification of tumor mutations," *Nature Methods*, vol. 10, no. 11, pp. 1108–1118, 2013.

[43] 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.

Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 921517, 9 pages
http://dx.doi.org/10.1155/2015/921517

Research Article

Coexpression Network Analysis of miRNA-142 Overexpression in Neuronal Cells

Ishwor Thapa,[1] Howard S. Fox,[2] and Dhundy Bastola[1]

[1]*College of Information Science and Technology, University of Nebraska Omaha, Omaha, NE 68182, USA*
[2]*Department of Pharmacology and Experimental Neuroscience, University of Nebraska Medical Center, Omaha, NE 68198, USA*

Correspondence should be addressed to Dhundy Bastola; dkbastola@unomaha.edu

Received 20 November 2014; Revised 26 January 2015; Accepted 28 January 2015

Academic Editor: Tatsuya Akutsu

MicroRNAs are small noncoding RNA molecules, which are differentially expressed in diverse biological processes and are also involved in the regulation of multiple genes. A number of sites in the 3′ untranslated regions (UTRs) of different mRNAs allow complimentary binding for a microRNA, leading to their posttranscriptional regulation. The miRNA-142 is one of the microRNAs overexpressed in neurons that is found to regulate *SIRT1* and *MAOA* genes. Differential analysis of gene expression data, which is focused on identifying up- or downregulated genes, ignores many relationships between genes affected by miRNA-142 overexpression in a cell. Thus, we applied a correlation network model to identify the coexpressed genes and to study the impact of miRNA-142 overexpression on this network. Combining multiple sources of knowledge is useful to infer meaningful relationships in systems biology. We applied coexpression model on the data obtained from wild type and miR-142 overexpression neuronal cells and integrated miRNA seed sequence mapping information to identify genes greatly affected by this overexpression. Larger differences in the enriched networks revealed that the nervous system development related genes such as *TEAD2, PLEKHA6*, and *POGLUT1* were greatly impacted due to miRNA-142 overexpression.

1. Introduction

MicroRNAs are small noncoding RNA molecules known to regulate gene expression at the transcriptional and posttranscriptional levels in a cell. The abnormal levels of microRNAs (miRNAs) may alter target gene expression or protein expression leading to pathogenesis of certain diseases. A large number of diseases including various forms of cancers have been implicated by aberrant expression of different miRNAs [1, 2]. Several studies have also explored miRNA and disease associations [3, 4]. In recent years, many *in vitro* gene expression profiling studies on overexpression or inhibition of miRNAs have been completed to discover novel miRNAs and their targets leading to innovative findings of miRNA associated diseases.

Differential gene expression studies focus on identifying genes that are up- and downregulated in a given experimental condition. However, it fails to capture the dependencies between genes and their relationships to other coexpressed genes. For this reason, network has become an increasingly popular model that analyzes relationships between genes/proteins within a biological system. Correlation networks based on coexpression have been extensively used [5]. Previously, a number of network properties have been associated with various biological processes, some of which include gene regulation, protein complex, and core diseasome within a given network [6–10]. Additionally, multiple networks have been compared using graph alignment algorithms to examine substructures that were conserved or destroyed between networks [11, 12].

A survey of the literature shows that miRNA-142 is upregulated in neurons and its overexpression through stable gene transfer downregulates the expression of NAD-dependent deacetylase Sirtuin 1 (*SIRT1*) gene and consequently the expression of the monoamine oxidase (*MAOA*) gene [13]. Such indirect regulations are inherent properties of a biological system and are difficult to capture in a gene expression study using differential analysis. Since miRNAs

regulate target gene expression levels in a cell through complementary binding of its sequence to the 3′ untranslated region (UTR) of the mRNA it targets, one may use miRNA seed sequence mapping information to explain the down-regulation of targeted genes. However, this gene regulation is not consistent in all of the miRNA seed sequence targets and conversely, genes that are regulated do not always have the seed sequence mapped. Therefore, in the present study, we propose a coexpression based method which incorporates multiple knowledge sources to find relevant target genes.

The correlation networks are built with the gene expression profiling data from (1) control (called the miR-null network) and (2) neuronal cells overexpressing miRNA-142 (called the miR-142 network) and are enriched with different knowledge sources derived from miRNA-mRNA seed mapping and differential gene expression. The comparison between the two enriched networks resulted in a number of genes, which were greatly affected by the overexpression of miRNA-142. Analyzing only the differential expression and miRNA-mRNA seed mapping led to noisy results. However, when coexpression networks were enriched with the different knowledge sources mentioned above, the networks highlighted key genes impacted by the biological transformation. The results showed some genes common between the coexpression network enriched with seed sequence mapping and the differentially expressed (DE) genes. These observations along with the approach taken in the present study are expected to be valuable in obtaining a comprehensive list of genes, which might be directly or indirectly regulated by the overexpression of miRNA in the cells.

2. Materials and Methods

A pipeline using the existing and new tools was developed in the process of this study. The Affymetrix power tool (APT) software package was used to retrieve exon level expression data and the 3′ UTR mapping was performed using Perl programming. Other analyses were accomplished using free and open source (FOSS) tools like R, Cytoscape, Limma, and GOFunction [14–18].

2.1. Dataset. Affymetrix gene expression data of RNA extracted from three independent overexpressed miRNA-142 clones and three stable clones with no miRNA overexpression were processed with the APT package for background correction, normalization, and summarizing probe sets. This data is available in NCBI GEO identified as GSE50133 series. The default parameters were used with *apt-probeset-summarize* and quantile normalization was applied in *rma-sketch* to obtain gene expression values for all samples. This gene level expression data contained three columns of miR-null replicates and three columns of miR-142 replicates. In order to find complementary mapping of miRNA seed sequence to 3′ UTR of mRNA, miRNA-142 seed sequence information was downloaded from TargetScan website and all the human mRNA sequences were obtained from Ensembl FTP site [19, 20].

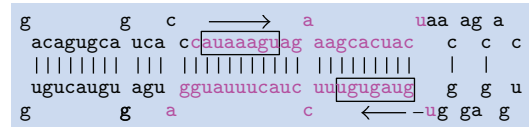


Figure 1: Stem-loop structure of miRNA-142 from miRBase website [23]. The highlighted boxes represent seed sequences for miR-142-5p and miR-142-3p.

2.2. Network Construction. Pearson correlation coefficients were calculated for all pairs of genes in miR-142 overexpressed dataset and miR-null dataset. To address multiple testing problem while computing the correlation coefficients for all pairs of genes, various P value adjustment methods such as "Holm," "Hochberg," and "Bonferroni" were applied. All the adjusted methods produced the same result and the minimum correlation coefficient was found to be 0.9998. Using the sample size estimation software (http://www.cct.cuhk.edu.hk/stat/other/correlation.htm) based on [21], it was derived that, with significance level at 0.01 and power of 80%, the number of samples could be of size three, if the minimum correlation was set to 0.9998. Thus, with confidence, the correlation coefficient values were represented as edges and genes as nodes to construct two independent networks. Only positive correlations were considered to model the coexpression as a positive effect. This resulted in two small, undirected networks, named here as miR-142 and miR-null. The Cytoscape tool was used to visualize the networks and enrich them with various information such as differential expression and miRNA seed mapping.

2.3. miRNA-mRNA Seed Matching. The miRNA-142 is one of the several miRNAs, which has both 5′ and 3′ arms of the stem-loop that can independently participate in regulation [22]. Figure 1 shows the miRNA-142 stem-loop structure obtained from the miRBase [23]. For simplicity, 5′ arm of miRNA-142 is written as miR-142-5p and 3′ arm as miR-142-3p.

The miRNA-mRNA complementary binding, which may result in the mRNA degradation or translation repression, is not always exact. Studies have shown that the binding may include an imperfect site [24]. Therefore, the exact seed sequences of miR-142-5p (AUAAAGU), miR-142-3p (GUAGUGU), and their 1-base mutated seed sequences were complementarily mapped to 3′ UTR of all the transcripts obtained from human genome. The human mRNA sequences were downloaded from the Ensembl FTP site [19, 20]. The list of both the exact and inexact seed sequences is presented in Table 1.

2.4. Differential Analysis. Due to small number of samples (three) in each group, we used the "lmFit" and "topTable" procedures from "Limma" package [17]. The genes which were differentially expressed with adjusted P values ≤ 0.1 were used in this study. To address multiple testing problem, P values were adjusted using "Benjamini and Hochberg" method. Additionally, we have also used the Mann-Whitney

Table 1: Seed sequence of miR-142 and all possible one-base mutation. Bold base indicates mutated site.

Original/mutated	Seed sequence	
	miR-142-5p	miR-142-3p
Original	AUAAAGU	GUAGUGU
hsa-miR-142-1a	**U**UAAAGU	**A**UAGUGU
hsa-miR-142-1b	**G**UAAAGU	**U**UAGUGU
hsa-miR-142-1c	**C**UAAAGU	**C**UAGUGU
hsa-miR-142-2a	A**A**AAAGU	G**A**AGUGU
hsa-miR-142-2b	A**G**AAAGU	G**G**AGUGU
hsa-miR-142-2c	A**C**AAAGU	G**C**AGUGU
hsa-miR-142-3a	AU**U**AAGU	GU**G**GUGU
hsa-miR-142-3b	AU**G**AAGU	GU**U**GUGU
hsa-miR-142-3c	AU**C**AAGU	GU**C**GUGU
hsa-miR-142-4a	AUA**U**AGU	GUA**A**UGU
hsa-miR-142-4b	AUA**G**AGU	GUA**U**UGU
hsa-miR-142-4c	AUA**C**AGU	GUA**C**UGU
hsa-miR-142-5a	AUAA**U**GU	GUAG**A**GU
hsa-miR-142-5b	AUAA**G**GU	GUAG**G**GU
hsa-miR-142-5c	AUAA**C**GU	GUAG**C**GU
hsa-miR-142-6a	AUAAAA**U**	GUAGU**A**U
hsa-miR-142-6b	AUAAA**U**U	GUAGU**U**U
hsa-miR-142-6c	AUAAA**C**U	GUAGU**C**U
hsa-miR-142-7a	AUAAAG**A**	GUAGUG**A**
hsa-miR-142-7b	AUAAAG**G**	GUAGUG**G**
hsa-miR-142-7c	AUAAAG**C**	GUAGUG**C**

(also called the Wilcoxon Rank Sum) test to perform the t-test.

2.5. Functional Analysis. The functional analysis program was written in R using GOFunction package [18]. The GOFunction tool is an enrichment analysis tool for Gene Ontology (GO). One advantage of using this tool is that it combines redundant GO terms based on the GO structure and gives statistically interpretable enriched GO terms. By default, the P value adjustment method selected was the "BY" option representing the "Benjamini, Hochberg, and Yekutieli" method. Table 5 shows GO terms enriched in the differentially expressed gene list obtained from the GOFunction.

3. Results

The goal of this study was to understand the changes in a cell due to the overexpression of miRNA-142 and its downstream effect on cellular functions.

In our results, we first show why the miRNA seed mapping and differential expression analysis could lead to noisy results (with many false positives and true negatives). Next, we show the impact of miR-142 overexpression on the coexpression networks of genes. These networks are further enriched with miRNA-mRNA seed sequence mapping, which illustrates that the networks are biologically relevant. We apply differential analysis result to our network and find an overlap between the differentially expressed genes and the nodes (genes) in the network. Functional analysis of the differentially expressed genes reveals enrichment in nervous system/neuron related genes.

Table 2: Overlap between differentially expressed (DE) genes and miRNA seed targets. The first three rows represent exact seed sequence mapping and the last three represent inexact seed sequence mapping. "P" represents adjusted P values for the differential analysis. There are genes, which are not mapped to any miR-142 seed sequence but are not shown in this table.

Common genes	DE ($P \leq 0.1$)	
	Down	Up
miR-142-5p	12	12
miR-142-3p	3	1
miR-142-5p/3p (both)	2	1
Inexact miR-142-5p	9	2
Inexact miR-142-3p	5	1
Inexact both	12	3
Total number of genes	60	25

3.1. Overlap between DE Genes and Seed Sequence Mapping. We independently identified genes which are differentially expressed and also mapped seed sequence of miR-142-5p and miR-142-3p to all the 3-prime untranslated regions (3p UTRs) of human mRNAs. We discuss this process in Section 2. Here, we show Table 2 that contains number of genes that are up- and downregulated and also have miRNA seed sequence mapping to their 3p UTRs. We observed that both the upregulated and the downregulated genes have large fraction (74%) of genes with matching seed sequence.

Since the miRNA-mRNA complementary binding may include an imperfect site, we also compared the inexact seed sequence mapping results with the differentially up- and downregulated gene lists (discussed more in Section 2). With inexact mapping, we also observed similar results as seen for exact seed sequence mapping.

When we analyzed the differential expression using the "Limma" package (see Figure 2), we found 85 genes were differentially expressed with P values ≤ 0.1 (P values corrected by Benjamini and Hochberg method). We checked these 85 genes with the DE list obtained from the "Wilcoxon Rank Sum (WRS)" test and found all of them included. Out of 85 DE genes obtained using "Limma," 63 (20 upregulated and 43 downregulated) genes had miRNA-142 seed sequence mapping. We observed that both up- and downregulated genes have seed sequence mapping and, conversely, not all the differentially expressed genes have seed sequence mapping. The 17 downregulated genes and 5 upregulated genes (P value ≤ 0.1) did not have any miR-142 seed mapping. Hence, direct relationship between gene regulation and a microRNA cannot be inferred only by looking into seed sequence mapping. Moreover, the result in Table 2 suggests that the miRNA-mRNA seed sequence mapping may directly or indirectly impact gene regulation. The Chi-square test was inconclusive to suggest any relation between up-/downregulation and

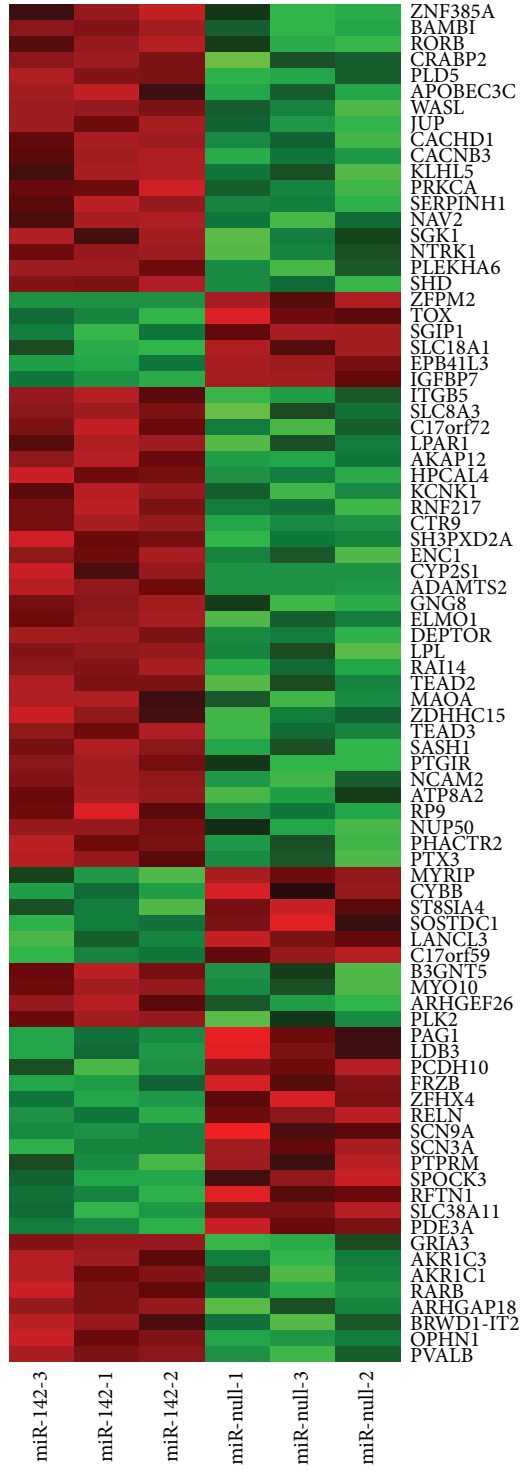


Figure 2: Heatmap showing normalized gene expression values for 85 DE genes obtained from "Limma" package.

preference to either of the arms of miRNA-142 ($\chi^2 = 2.49$, df = 2, and P value = 0.2878).

3.2. Distinct Coexpressed Networks in miRNA-Null and miRNA-142 Overexpression. Coexpressed networks were generated by assigning an edge for Pearson's correlation coefficient between every pair of genes. For addressing false positives and true negatives in large number of multiple testing, we applied different P value adjustment methods and obtained the same results. There were a fewer number of edges, which passed through the P value adjustment methods. Hence, the coexpressed networks were small-sized

Table 3: Size of miR-142 and miR-null networks.

	Number of nodes	Number of edges	Edge density
miR-null	57	217	0.136
miR-142	52	158	0.119

Table 4: Count of nodes/genes, which have seed sequence mapping in human genome and are present in networks.

	miR-null	miR-142	Genome
miR-142-5p	9	6	2914
miR-142-3p	1	2	720
miR-142-5p/3p (both)	—	—	387
Inexact miR-142-5p	7	7	3644
Inexact miR-142-3p	7	8	2520
Inexact miR-142-5p/3p (both)	9	4	4820

networks. No overlap was found between miR-null and miR-142 networks, meaning all the nodes and edges were different. This suggests that there is a big impact in the coexpression of genes due to the miRNA-142 overexpression. Table 3 shows the size of these networks. All 57 genes in Figure 3 and all 52 genes in Figure 4 are candidate genes of interest because they appear and disappear in these networks due to the miRNA-142 overexpression.

Next, we demonstrate that these networks are also biologically relevant by enriching it with miRNA-mRNA seed sequence mapping. We observe that more than 50% of the nodes (genes) in these networks have miRNA-142 seed sequence mapped to its 3′ UTR.

Table 4 shows the exact number of nodes in these networks with the seed sequence mapping and also in all the genes from human genome. If we take into account the larger probability of inexact seed mapping (21 times more than that of exact mapping because of the 21 different mutated seed sequences shown in Table 1), a large proportion of nodes in the networks are expected to have inexact mapping. However, we observed 18 and 42 nodes with exact and inexact mapping, respectively, which is less than the anticipated ratio. Moreover, if we compare the number of nodes having exact mapping of miR-142-5p and that with again exact mapping of miR-142-3p in each network, we observed higher number of nodes with the exact mapping of miR-142-5p. This suggests that the 5′ arm of miRNA-142 has greater impact than the 3′ arm during the miRNA-142 overexpression. The Chi-square test to check if the seed mapping (exact/inexact) in coexpression networks is independent of matching by different miRNA arms suggested associations between the number of exact/inexact seed mapping targets in coexpression networks and the matching by different arms ($\chi^2 = 13.6125$, df = 2, and P value = 0.0011).

3.3. Overlap between DE Genes, Seed Mapping, and the Enriched Network. The enriched network in Figure 4 shows the overlap between the miR-142 coexpressed genes, miRNA-mRNA seed mapping genes, and the differentially expressed genes. There are two genes (*TEAD2* and *PLEKHA6*) which appear in miR-142 coexpression network, have seed mapping, and are downregulated in the miR-142 overexpression. We examine the roles of these two genes in Section 4.

While comparing the nodes in the enriched networks, their seed mapping, and other miRNA target prediction tools such as TargetScan [19], DIANA-microT [25], and miRanda [26], we observed that a large proportion of the seed mapping nodes in the networks overlap with the targets predicted from those publicly available tools (see Figure 5). The Chi-square test was inconclusive to suggest any relation between occurrence in coexpressed networks and preference to either of the seed mapping/publicly available methods ($\chi^2 = 4.38$, df = 2, and P value = 0.1119).

3.4. Functional Analysis Reveals DE Genes Enriched in Nervous System. The individual networks (miR-null and miR-142) only contained around 50 nodes (genes). With this list of genes, no significant functional enrichment was observed. Next, we considered differentially expressed genes with adjusted P value ≤ 0.1. This list consisted of 85 genes and the functional enrichment for this list is shown in Table 5. Among several GO terms enriched, the table shows that all these terms are related to neuron and nervous system.

4. Discussion and Conclusion

Network analysis in gene expression profiling study is one of the rising trends in bioinformatics. Although the correlation does not imply causation, in a microRNA overexpression experiment, we have shown that a network model can be applied to extract meaningful biological relationships. The combined use of microRNA-mRNA 3′ UTR seed mapping and differential gene expression can further strengthen the efficacy of network analysis.

By comparing the enriched networks in different experimental conditions, we showed that several genes were greatly affected by the respective treatment, namely, miRNA-142 overexpression. The results obtained from the "Limma" package show that the genes *TEAD2* and *PLEKHA6*, which appear in miR-142 overexpression network and have seed sequence mapping, are downregulated. Similar results obtained from Wilcoxon Rank Sum test show *POGLUT1* in addition to *TEAD2* and *PLEKHA6*. Figure 6 shows the downregulation effect on these genes due to the overexpression of miR-142. The functional enrichment of differentially expressed genes using "Limma" package shows the enrichment of neuron and nervous system related GO terms. Next, we discuss the relevance of these genes in neuronal cells.

4.1. TEAD2 Gene and Nervous System Development. The *TEAD2* gene encodes for *Tead2* transcription factor, which is one of the first transcription factors expressed at the beginning of mammalian development [27]. In 2007, [28]

Table 5: Results from GO functional analysis of differentially expressed genes.

GO ID	Name	Total	In DE list	*P* value	Adjusted *P* value
GO:0042995	Cell projection	1451	22	$2.62495317149281e-07$	0.00313028071281764
GO:0097458	Neuron part	903	16	$2.19851448246544e-06$	0.0131087433407394
GO:0043005	Neuron projection	726	14	$3.97485763192762e-06$	0.0144904323637896
GO:0030425	Dendrite	360	10	$4.86048503367531e-06$	0.0144904323637896

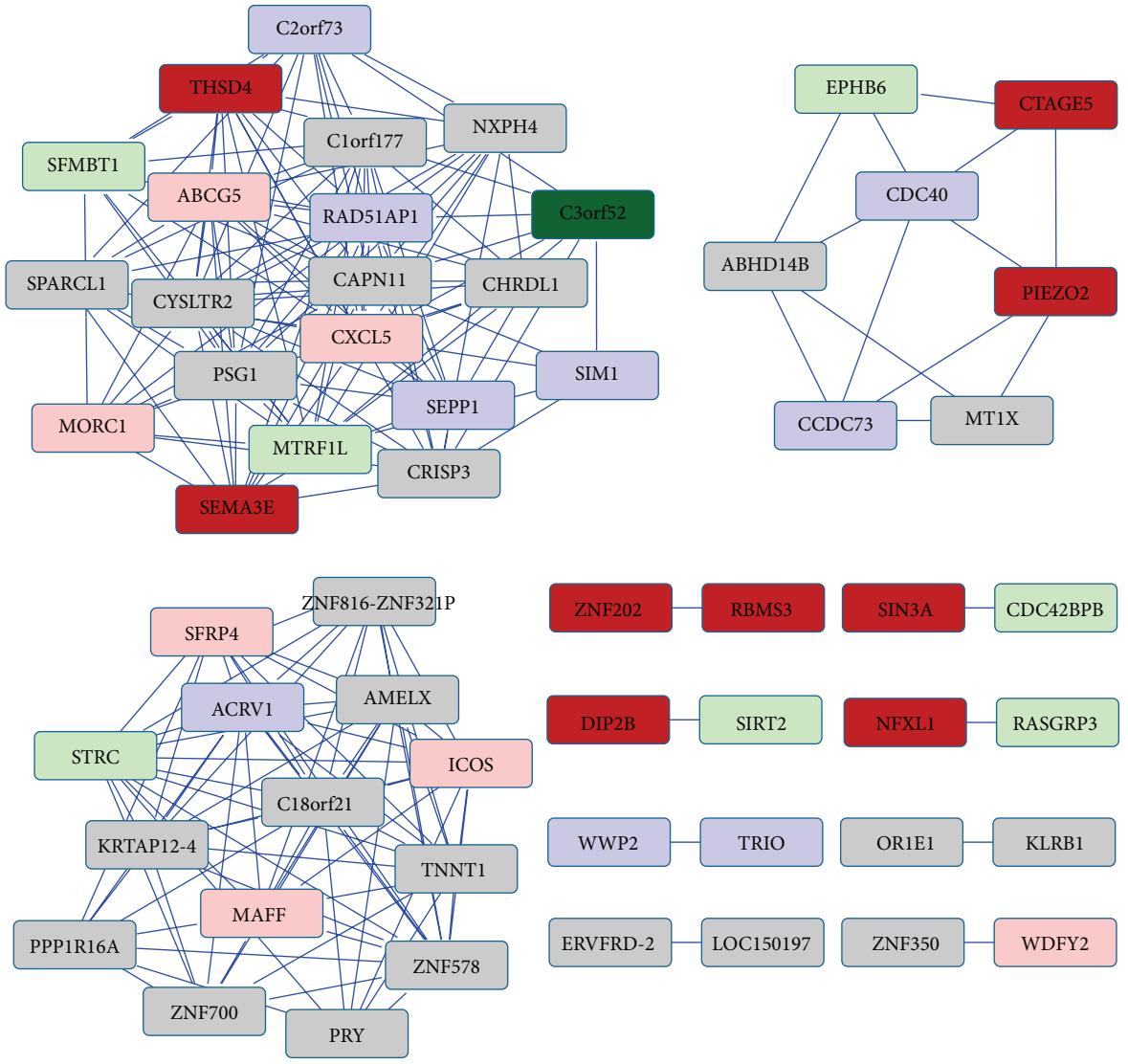


Figure 3: The miR-null coexpression network enriched with miRNA-mRNA seed sequence mapping. Node represents gene name and an edge between two nodes (genes) represents coexpression of the genes. Color of the node represents the miRNA seed sequence mapping onto the 3p UTR of a gene (mRNA). The "dark red" node represents exact miR-142-5p seed mapping, while the "pink" node represents inexact miR-142-5p seed mapping. Similarly, "dark green" node represents exact mapping of miR-142-3p seed and the "light green" node represents inexact mapping. The "blue" node represents genes with inexact mapping for both miR-142-3p and miR-142-5p seed sequences. No node was found to have exact mapping of both miR-142-5p and miR-142-3p.

showed that this gene is required during neural development specifically for the neural tube closure. Recently, in 2014, [29] showed that *Tead2* together with *Yap* and *Taz* controls the expression of genes critical for epithelial-mesenchymal transition (EMT). EMT has long been known as an essential process for neural tube formation. In our study, we observed that this gene is differently coexpressed in miR-142 overexpressed network, downregulated in miR-142 overexpression, and has exact seed mapping for miRNA-142-5p. These observations and its role in nervous system

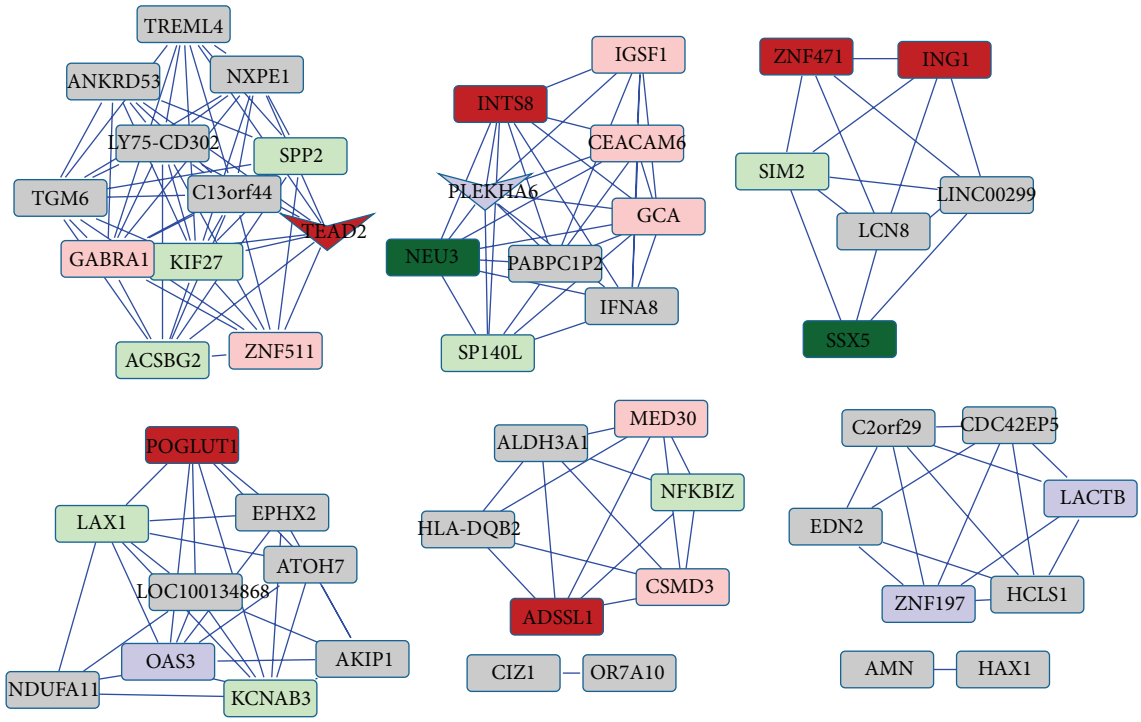


Figure 4: The miR-142 coexpression network enriched with miRNA-mRNA seed sequence mapping and differential expression. Node represents gene name and an edge between two nodes (genes) represents coexpression of the genes. Color of the node represents the miRNA seed sequence mapping onto the 3p UTR of a gene (mRNA). The "dark red" node represents exact miR-142-5p seed mapping, while the "pink" node represents inexact miR-142-5p seed mapping. Similarly, "dark green" node represents exact mapping of miR-142-3p seed and the "light green" node represents inexact mapping. The "blue" node represents genes with inexact mapping for both miR-142-3p and miR-142-5p seed sequences. No node was found with exact mapping of both miR-142-5p and miR-142-3p. The shape of a node represents the differential expression. Down-pointing arrow shaped node represents downregulation in expression.

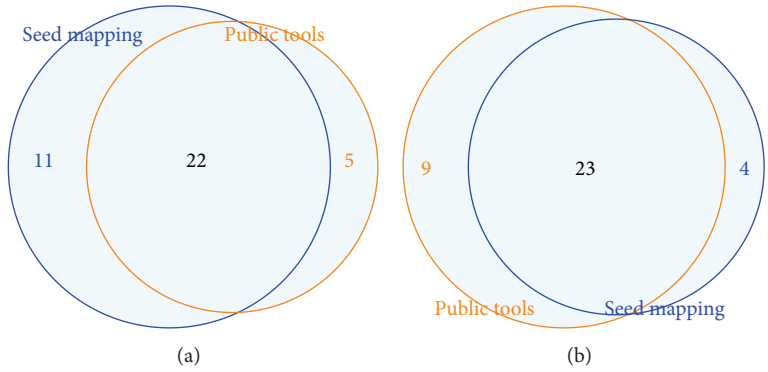


Figure 5: Venn diagrams showing overlap between nodes in coexpression networks ((a) miR-null network, (b) miR-142 network), seed mapping, and targets predicted by public tools like TargetScan, DIANA-microT, and miRanda. A R package [33] was used to create these figures.

development further demonstrate that it is a crucial target in miR-142 overexpression.

4.2. PLEKHA6 Gene and Schizophrenia. *PLEKHA6* gene encodes for *pleckstrin homology (PH) domain containing family A member 6* protein. Recently, in 2014, [30] found that *PLEKHA6* is involved in intracellular signaling and associated its polymorphisms with schizophrenia. The authors suggested that this gene might be involved in the pathophysiology of schizophrenia and the therapy response towards antipsychotics. In our study, we found that *PLEKHA6* was present in our miR-142 network and was downregulated in miR-142 overexpression sample. It also contained inexact mapping of both miR-142-5p and miR-142-3p seed sequences in the 3p UTR. Studies have shown that this seed sequence mapping is not perfect. Allowing a mismatch in miRNA seed

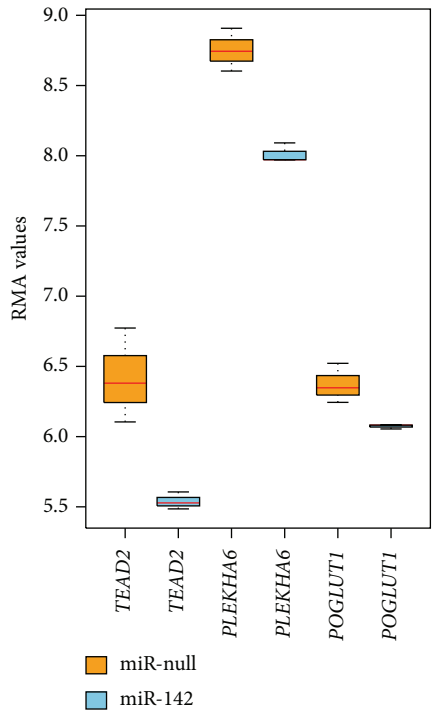


Figure 6: Expression values of *TEAD2*, *PLEKHA6*, and *POGLUT1* genes in miR-null and miR-142 samples. The first box for each gene represents gene expression in miR-null sample and the second box represents the same for miR-142 overexpressed sample. *TEAD2* and *PLEKHA6* were found differentially expressed using both the "Limma" package and "WRS" test (P value ≤ 0.1), while *POGLUT1* was found differentially expressed in the "WRS" t-test (P value ≤ 0.1).

sequence can increase the regulation of diverse targets leading to the coexpression of these target genes. The correlation network captures this information and can serve as a better model to extract such information from these studies.

4.3. POGLUT1 Gene and Notch Signaling Pathway. *POGLUT1* (also known as *RUMI* or *hCLP46*) is a gene that encodes for protein O-glucosyltransferase 1. It is shown that this gene is required for Notch signaling [31]. One of the major functions of Notch signaling is the neuronal function and its development. In [32], the authors have showed that *hCLP46* is the homolog of *Rumi* and its knockdown may result in Notch signaling impairment. Like *TEAD2*, we also found that *POGLUT1* was differently coexpressed in miR-142 network, was downregulated by miR-142 overexpression, and contained miR-142-5p exact seed mapping in its 3p UTR. These observations suggest that *POGLUT1* is another significant target in miR-142 overexpression.

In this study, we have identified candidate genes such as *TEAD2*, *PLEKHA6*, and *POGLUT1* that were highlighted in our enriched network and were greatly impacted by miRNA-142 overexpression. Most importantly, these genes were known to have very crucial neuronal functions.

Disclaimer

The work content is solely the responsibility of the authors and does not necessarily represent the official views of NIH.

Conflict of Interests

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

Acknowledgment

This work was supported by NIH Grants P30 MH062261 and R01 DA030962.

References

[1] R. Garzon, G. A. Calin, and C. M. Croce, "MicroRNAs in cancer," *Annual Review of Medicine*, vol. 60, pp. 167–179, 2009.

[2] S. Fichtlscherer, S. de Rosa, H. Fox et al., "Circulating microRNAs in patients with coronary artery disease," *Circulation Research*, vol. 107, no. 5, pp. 677–684, 2010.

[3] M. Lu, Q. Zhang, M. Deng et al., "An analysis of human microRNA and disease associations," *PLoS ONE*, vol. 3, no. 10, Article ID e3420, 2008.

[4] Q. Jiang, Y. Wang, Y. Hao et al., "miR2Disease: a manually curated database for microRNA deregulation in human disease," *Nucleic Acids Research*, vol. 37, supplement 1, pp. D98–D104, 2009.

[5] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.

[6] Z. Bar-Joseph, G. K. Gerber, T. I. Lee et al., "Computational discovery of gene modules and regulatory networks," *Nature Biotechnology*, vol. 21, no. 11, pp. 1337–1342, 2003.

[7] 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.

[8] N. Pržulj, D. A. Wigle, and I. Jurisica, "Functional topology in a network of protein interactions," *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2004.

[9] S. Chavali, F. Barrenas, K. Kanduri, and M. Benson, "Network properties of human disease genes with pleiotropic effects," *BMC Systems Biology*, vol. 4, no. 1, article 78, 2010.

[10] V. Janjić and N. Pržulj, "The core diseasome," *Molecular BioSystems*, vol. 8, no. 10, pp. 2614–2625, 2012.

[11] J. Berg and M. Lässig, "Local graph alignment and motif search in biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 41, pp. 14689–14694, 2004.

[12] N. Pržulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[13] A. D. Chaudhuri, S. V. Yelamanchili, and H. S. Fox, "MicroRNA-142 reduces monoamine oxidase a expression and activity in neuronal cells by downregulating SIRT1," *PLoS ONE*, vol. 8, no. 11, Article ID e79579, 2013.

[14] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2005.

[15] P. Shannon, A. Markiel, O. Ozier et al., "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[16] R. Christmas, I. Avila-Campillo, H. Bolouri et al., "Cytoscape: a software environment for integrated models of biomolecular interaction networks," in *AACR Education Book*, pp. 12–16, 2005.

[17] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, Eds., pp. 397–420, Springer, New York, NY, USA, 2005.

[18] J. Wang, *GO-Function: Deriving Biologcially Relevant Functions from Statistically Significant Functions*, R Package Version 1.6.0, 2011.

[19] B. P. Lewis, C. B. Burge, and D. P. Bartel, "Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets," *Cell*, vol. 120, no. 1, pp. 15–20, 2005.

[20] T. Hubbard, D. Barker, E. Birney et al., "The Ensembl genome database project," *Nucleic Acids Research*, vol. 30, no. 1, pp. 38–41, 2002.

[21] J. M. Lachin, "Introduction to sample size determination and power analysis for clinical trials," *Controlled Clinical Trials*, vol. 2, no. 2, pp. 93–113, 1981.

[22] M. Skårn, T. Barøy, E. W. Stratford, and O. Myklebost, "Epigenetic regulation and functional characterization of microRNA-142 in mesenchymal cells," *PLoS ONE*, vol. 8, no. 11, Article ID e79231, 2013.

[23] S. Griffiths-Jones, R. J. Grocock, S. van Dongen, A. Bateman, and A. J. Enright, "miRBase: microRNA sequences, targets and gene nomenclature," *Nucleic Acids Research*, vol. 34, supplement 1, pp. D140–D144, 2006.

[24] B. John, A. J. Enright, A. Aravin, T. Tuschl, C. Sander, and D. S. Marks, "Human microRNA targets," *PLoS Biology*, vol. 2, no. 11, article 363, 2004.

[25] M. Kiriakidou, P. T. Nelson, A. Kouranov et al., "A combined computational-experimental approach predicts human microRNA targets," *Genes and Development*, vol. 18, no. 10, pp. 1165–1178, 2004.

[26] B. John, A. J. Enright, A. Aravin, T. Tuschl, C. Sander, and D. S. Marks, "Human microrna targets," *PLoS Biology*, vol. 2, no. 11, article e363, 2004.

[27] K. J. Kaneko, E. B. Cullinan, K. E. Latham, and M. L. De-Pamphilis, "Transcription factor mTEAD-2 is selectively expressed at the beginning of zygotic gene expression in the mouse," *Development*, vol. 124, no. 10, pp. 1963–1973, 1997.

[28] K. J. Kaneko, M. J. Kohn, C. Liu, and M. L. DePamphilis, "Transcription factor TEAD2 is involved in neural tube closure," *Genesis*, vol. 45, no. 9, pp. 577–587, 2007.

[29] M. Diepenbruck, L. Waldmeier, R. Ivanek et al., "Tead2 expression levels control the subcellular distribution of yap and taz, zyxin expression and epithelial-mesenchymal transition," *Journal of Cell Science*, vol. 127, no. 7, pp. 1523–1536, 2014.

[30] I. Spellmann, D. Rujescu, R. Musil et al., "Pleckstrin homology domain containing 6 protein (plekha6) polymorphisms are associated with psychopathology and response to treatment in schizophrenic patients," *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 51, pp. 190–195, 2014.

[31] M. Acar, H. Jafar-Nejad, H. Takeuchi et al., "Rumi is a cap10 domain glycosyltransferase that modifies notch and is required for notch signaling," *Cell*, vol. 132, no. 2, pp. 247–258, 2008.

[32] W. Ma, J. Du, Q. Chu et al., "HCLP46 regulates U937 cell proliferation via notch signaling pathway," *Biochemical and Biophysical Research Communications*, vol. 408, no. 1, pp. 84–88, 2011.

[33] H. Chen, *VennDiagram: Generate High-Resolution Venn and Euler Plots*, R Package Version 1.6.9, 2014, http://cran.r-project.org/package=VennDiagram.