Hindawi Publishing Corporation
ISRN Genomics
Volume 2013, Article ID 191206, 8 pages
http://dx.doi.org/10.1155/2013/191206

Review Article

A Review of Soft Computing Techniques for Gene Prediction

Neelam Goel, Shailendra Singh, and Trilok Chand Aseri

Department of Computer Science and Engineering, PEC University of Technology, Sector-12, Chandigarh 160012, UT, India

Correspondence should be addressed to Neelam Goel; neelam.goyal85@gmail.com

Received 26 December 2012; Accepted 6 February 2013

Academic Editors: S. Cavallaro, A. Piepoli, and A. Stubbs

In the past decade, various genomes have been sequenced in both plants and animals. The falling cost of genome sequencing manifests a great impact on the research community with respect to annotation of genomes. Genome annotation helps in understanding the biological functions of the sequences of these genomes. Gene prediction is one of the most important aspects of genome annotation and it is an open research problem in bioinformatics. A large number of techniques for gene prediction have been developed over the past few years. In this paper a theoretical review of soft computing techniques for gene prediction is presented. The problem of gene prediction, along with the issues involved in it, is first described. A brief description of soft computing techniques, before discussing their application to gene prediction, is then provided. In addition, a list of different soft computing techniques for gene prediction is compiled. Finally some limitations of the current research and future research directions are presented.

1. Introduction

In the past several years, there has been a virtual explosion of genomic sequence data with numerous of genomes in various stages of sequencing and annotation. As the human genome project came to an end in 2003, all the human chromosomes have been sequenced [1]. In fact, with the number of genomes sequenced numbering over one hundred, it is clear that quick, accurate annotation of these genomes is essential to learning more about biology and the evolutionary relationships between these genomes [2]. However, the pace of genome annotation is not matching the pace of genome sequencing. The experimental annotation of genomes is slow and time consuming. Therefore there is a real need to develop automatic techniques for genome annotation. The first step towards successful genome annotation is gene prediction. Gene prediction is mainly concerned with the identification of protein-coding genes in DNA but may also include the identification of other functional elements of genomic DNA such as RNA genes and regulatory regions. A large number of techniques have been developed for the prediction of protein-coding genes. However the prediction accuracy of these techniques is still far from satisfactory. There are two main problems with the existing protein-coding gene prediction techniques. First, most of the techniques are developed for specific genomes. Second, the gene level accuracy of these techniques is very low. It is obvious that further improvement to protein-coding gene prediction is much needed. An extensive list of existing gene prediction softwares can be found in [3].

Most of the previous reviews on this problem have focused on traditional techniques of gene prediction like hidden Markov model, decision trees, and dynamic programming-based approaches [4–6]. In addition to traditional gene prediction techniques like those based on hidden Markov model and dynamic programming, approaches based on soft computing techniques have gained popularity in recent times. Soft computing techniques can work well for gene prediction because they can handle uncertainty and approximate nature of data. A review of traditional and computational intelligence technique is presented in [1]. The problem of predicting RNA-coding genes is a promising area of research these days. In a recent review, various techniques to predict one of the classes of noncoding RNA (ncRNA) are presented [7]. The techniques reviewed are mainly based on rules and support vector machine. However, none of the aforementioned reviews focused on soft computing techniques for gene prediction in the last few years. So the main focal point of

this paper is to review soft computing techniques for both protein-coding and ncRNA gene prediction.

The rest of the paper is organized as follows: in the next section the problem of gene prediction is addressed along with the issues involved in it. Thereafter, different soft computing techniques like neural networks, genetic algorithms, and hybrid approaches are discussed along with their application to gene prediction. Subsequently, a theoretical analysis of these techniques is presented. Finally some conclusions and future research directions are given.

2. Background

In this section the problem of gene prediction is discussed followed by the issues that still need improvement.

Gene prediction is the problem of identifying the portions of DNA sequence that are biologically functional. This especially includes protein-coding regions but may also include other functional elements such as ncRNA genes. The main aim behind the problem of gene prediction is to correctly label each element of DNA sequence as belonging to protein-coding region, RNA coding region, and noncoding or intergenic regions. Intergenic regions are the regions of DNA in between genes. The problem of gene prediction can then be formally stated as follows [1].

Input. A sequence of DNA

$$X = (x_1, \ldots, x_n) \in \Sigma^*, \quad \text{where } \Sigma = \{\text{A, T, C, G}\}. \qquad (1)$$

Output. Correct labeling of each element in X as belonging to protein-coding region, RNA coding region, noncoding region, or intergenic region.

By identifying the location of different regions in the DNA sequence, genes can be predicted easily. The basic terminology required to understand the problem of gene prediction is illustrated in [8]. All living organisms are made up of cells and these cell falls into two categories: eukaryotes and prokaryotes. Computational methods of gene prediction are developed for both prokaryotes and eukaryotes. In prokaryotes, genes are made up of long coding segments, that is, open reading frames (ORFs). On the other hand, genes in eukaryotes consist of coding segments interrupted by long noncoding segments. These coding segments are termed as exons and noncoding segments as introns. In case of human eukaryotes only 3% of DNA sequence is coding [9].

Gene prediction is relatively simple in prokaryotes due to higher gene density and the absence of introns [10]. The main difficulty in prokaryote gene prediction is the presence of overlapping regions [11]. The process is more complex for eukaryotes, because of large genome size and short exons are bordered by large introns. Furthermore, eukaryote coding segments are subject to alternative splicing, that is, a process of joining exons in different ways during RNA splicing [12]. Indeed, it is estimated that more than 95% of human genes show evidence of at least one alternative splice site [9]. In this paper the soft computing techniques of gene prediction for both eukaryote and prokaryote are being discussed.

There are two important aspects to any gene prediction program: one is the type of information used by the program and the other is the algorithm that is employed to combine that information into a consistent prediction. Three types of information are generally used in predicting gene structure: functional sites in the sequence, content statistics, and similarity to known genes [6]. Among the types of functional sites are splice sites, start and stop codons, branch points, promoters and terminators of transcription, polyadenylation sites, and various transcription binding sites. These sites are generally known as signals and methods used to detect them are signal sensors. Genomic DNA signals can be contrasted with extended and variable length regions such as exons and introns, which are recognized by different methods that are called content sensors [13]. These content sensors are generally categorized as intrinsic content sensors. Intrinsic content sensors use statistical properties (compositional bias, codon usage, etc.) of the coding segments to distinguish them from noncoding segments. The existence of a sufficient similarity with a biologically characterized sequence can also be used as a mean for gene prediction. These similarity-based methods have often been called extrinsic content sensors [14]. The methods that use signal or both signal and intrinsic content sensors are known as ab initio methods of gene prediction. In the last few years, gene prediction methods based on the combination of ab initio and similarity information have been developed. The prediction accuracy of these combined methods is better than the methods that are purely based on ab initio approaches [15]. Although protein-coding gene prediction methods have achieved a significant level of accuracy but there are issues that still need improvement and these issues are as follows:

(i) prediction of short exons,

(ii) prediction of complete gene structure,

(iii) prediction of partial and overlapping genes,

(iv) alternative splicing,

(v) database sequences are not completely correct,

(vi) prediction of genes in newly sequenced genomes,

(vii) noncanonical splice sites.

So far in this section the background necessary for gene prediction is discussed. The next section describes soft computing techniques for gene prediction.

3. Soft Computing Techniques for Gene Prediction

Soft Computing is the modern approach for constructing a computationally intelligent system. The ultimate goal of soft computing is to emulate human mind as closely as possible [16]. Soft computing is the blend of methodologies designed to solve the real-world problems, which are not solved or too difficult to solve mathematically. Nowadays, soft computing techniques are identified as attractive alternatives to the standard, well-established "hard computing" methods. Traditional hard computing methods are often inconvenient for real-world problems. They always need a precisely stated systematic model and often a lot of computational time [17]. Unlike hard computing methods, soft computing methods

cope up with those problems that deal with imprecision, uncertainty, learning, and approximation to achieve tractability, robustness, and low-cost solutions [18]. The unique property of soft computing is that it is deeply involved in learning from experimental data that makes it suitable for gene prediction. While predicting genes, specific patterns in DNA sequence are recognized and soft computing techniques have been extensively used in pattern recognition problems [19]. Soft computing consists of several techniques, the most important being neural networks, genetic algorithms, and fuzzy logic. The significance of soft computing techniques lies in the fact that they are complementary, not competitive. In many cases a problem can be solved by using neural network, fuzzy logic, and genetic algorithm in combination rather than exclusively. This section describes the application of these soft computing techniques in the area of gene prediction.

3.1. Neural Networks. A neural network is an artificial representation of human brain. The main aim behind the development of neural network is to acquire human ability to adapt to changing circumstances and the environment. Artificial neural network (ANN) is an interconnected group of artificial neurons [18]. The main characteristic of ANN is their ability to learn. Neural network system helps in situations where one cannot formulate an algorithmic solution or can get lots of examples of the behavior required. These properties of neural networks make it suitable for predicting both types of genes, that is, protein coding and RNA coding. Neural networks can be divided into different architectures on the basis of learning algorithms. The use of various neural network architectures [20] including supervised and unsupervised learning algorithms for gene prediction is described here.

One of the earliest attempts to use neural network for gene prediction is made in 1991, GRAIL (gene recognition and analysis internet link). GRAIL is available in two versions: GRAIL-I [21] and GRAIL-II [22]. In GRAIL-I a multilayer feed-forward neural network is used, which receives input from seven statistical measures taken on a 99-base window. In this system sensor algorithms are used to derive the seven inputs. The main problem with this technique is that it predicts large exons while misses a large number of short exons.

A similar system [23] to GRAIL came in 1993. A program named GeneParser is described in [24], which is used to predict protein genes in genomic DNA sequences [25]. This program makes use of content statistics as well as site statistics to predict exons, introns, and their boundaries. Here both single and multilayer neural networks are used to combine information from these statistics and database search information. Here a recursive optimization procedure based on dynamic programming is used to find the most probable combinations of exon and introns. The dynamic programming algorithm used in this system has enforced some syntactical constraints on gene structures. The main intention behind the development of GeneParser is error tolerance. The performance of the program is better than similar programs developed in meanwhile.

A new version of GRAIL is developed in 1994. In this system, modifications are made in GRAIL-I to improve its performance. Although GRAIL-II can be used for protein-coding region recognition, PolyA site and transcription promoter recognition, gene model construction, translation to protein, and DNA/protein database searching capabilities, in this paper only the exon recognition and gene construction capabilities are discussed. In GRAIL-II variable-length windows are used to find the locations of introns and exons. This technique solves the problem of missing shorter exons by allowing it to examine all possible exons rather than just those in the sliding window. The gene-modeling module of this system uses a dynamic programming algorithm to form a single gene model from exon candidates by applying some constraints.

A further attempt to improve the performance of GRAIL-I was made in 1995. A system named CODEX was developed. The system [26] was developed with the intention to predict exons precisely in plant sequences. This system takes same inputs as taken by GRAIL-I. Unlike GRAIL-I it uses a series of neural networks for the classification task. The CODEX system predicts the location of coding regions by examining the output of the combination of five neural networks. The performance of the technique is much better in predicting the base positions at which an exon starts or ends. The technique helps in reducing the number of false predictions but it classifies some sequences as do not know.

GRAIL has evolved by the time it started in 1991. An improved version of GRAIL came in 1996, which uses a multiagent neural network system to recognize coding regions [27]. The system uses indel detection and correction algorithm to improve the prediction results.

Many gene prediction techniques utilize homology information, which helps in improving the prediction results. In the latest version of Grail, homology information has been incorporated. The new system resulted in improved performance and named as GrailExp [28]. A computational gene prediction system GIN (gene identification using neural nets and homology information) was developed in 1998 to avoid false positive predictions. The technique combines homology information from protein and expressed sequence tag databases into back-propagation neural network [29]. The program can recognize multiple genes within genomic DNA. GIN performs better than other methods (e.g., GeneID+ [30] or GeneParser3 [24]) that make use of homology information to predict genes. The performance of the system is better than GENSCAN [31] in gene level accuracy. This technique does not work well in the absence of homology information.

A system for predicting protein-coding regions based on single-nucleotide frequencies at three codon positions in ORFs and the redundancy of the entropy was developed in 2003. The system [32] is specifically designed to predict protein-coding regions in yeast genomes. Here a multilayer feed-forward artificial neural network (MLFANN) method is developed which takes as an input a 12-dimensional vector obtained from DNA sequence. The input ORFs from six different classes are trained and evaluated separately. The system predicts ORFs with 96% accuracy. The method is based on the assumption that coding sequence in 1st-class ORFs has similar statistical properties to those coding for 2nd–6th-class ORFs.

Most of the gene prediction programs developed during 90s are based on a single model to represent protein-coding regions in a genome and unable to predict genes that have atypical sequence composition. A system mainly developed for prokaryotes, based on self-organizing map [33], was developed in 2004 to identify multiple gene models within a genome. The system RescueNet [34] is developed specifically to deal with intragenomic compositional variations. The system makes use of relative synonymous codon usage (RSCU) as a measure of protein-coding potential. RescueNet predicts some genes that are not predicted by other methods and promising results can be obtained if the method is used in conjunction with other gene prediction programs. The program is able to find the general location of frameshifts within a large sequence. The main problem with this method is that it is not able to predict most start and stop sites exactly.

Recently many gene prediction programs are developed which predict genes in specific genomes. Such a system was developed in 2011, in which content-based gene prediction method is used in conjunction with back-propagation neural network for predicting genes. A method to predict Lac gene structures in Streptococcus pyogenes M Group A Streptococcus strains is described in [35]. The frequency of occurrence of all possible 64 codons, 4 nucleotides (A, T, G, C), and chemically similar nucleotides (A, T and G, C) altogether form 70 parameters calculated from Lac genes which are used to train the neural network. In this work a tool named SpyMGASLacGenePred is developed to identify ORFs from DNA sequences. From these ORFs Lac genes are predicted based on mean gene content. For predicted Lac genes the tool displays the position, length, frame, G + C content, and translated sequence. The performance of the tool is above an acceptable threshold level. The method is specifically developed to predict Lac genes. The main problem with this method is that it uses the same sequences to train and test the network, which leads to 100% sensitivity. The value of the sensitivity might drop from 100 if different sequences are used for training and testing of the network.

Another method that predicts essential genes (EG) in microbial genome was developed in 2011. Essential genes are the minimal set of genes, an organism needs for its survival [36]. The proposed method [37] relies solely on sequence-derived input features for making prediction. In this work three supervised classification methods, support vector machine (SVM), artificial neural network (ANN), and decision tree (DT), are used for classification task. In this method 52 genomic features corresponding to each gene are calculated by using, gene paralogs, amino acid composition, codon frequency, protein physiochemical features, and subcellular localization features. The dataset used in this work suffered from a class imbalance problem, which is reduced by employing homology clustering and random sampling. To test the generalizability of the classifiers across genome and taxonomic boundaries, two novel testing schemes leave-one-genome-out (LOGO) and leave-one-taxon group-out (LOTO) are used. The experimental results show that SVM and ANN perform better than DT with area under the receiver operating characteristics (AU-ROC) scores. The fundamental advantage of this method is the use of multigenome input to learn the classifier models and apply them to predict on new genomes.

Among all gene prediction programs very few of them have addressed the problem of predicting ncRNA genes. Noncoding RNA genes make transcripts that function as RNA [38]. The main difficulty in identifying these genes is diversity of ncRNA genes as well as lack of consensus patterns of such genes. Most of the gene prediction programs used neural network for predicting protein-coding genes but very little work has been done for noncoding RNA gene prediction. A machine learning technique for the prediction of known RNA genes in prokaryotic and archaeal genomes was developed in 2001. This approach is based on the fact that characteristic signals exist in the sequences of functional RNA (fRNA) that are distinguishable from noncoding regions of the genome [39]. In this work three parameters are used: compositional, structural motif, and calculated free energy of folding for RNA. Two of them are used to train and test the neural networks used in this technique. High prediction accuracy is achieved by using these networks, which shows that neural networks are able to learn to distinguish between RNA regions and noncoding regions. In addition to jackknife test, cross-prediction tests are also performed to increase the number of RNA genes. This method is only applicable for prokaryotic genomes.

Another approach to identify functional RNA (fRNA) genes using evolved neural networks is discussed in [40]. An fRNA gene discovery tool was developed in 2005 that uses an evolved neural network for pattern recognition. Evolutionary computation is used as an optimization method during training. The tool is mainly developed for eukaryotes *C. elegans* [41, 42] and *H. sapiens* [41, 43]. The results show that, ANN trained using evolutionary computation is capable of predicting fRNA coding regions with high prediction accuracy.

3.2. Genetic Algorithms. Genetic algorithms (GAs) are heuristic search algorithms based on the process of natural evolution [44, 45]. GAs often encode a candidate solution as a fixed-length bit string called chromosome. GA is mainly used to find an optimal solution to an optimization problem. The first attempt of using genetic algorithm as a main tool for gene prediction was made in 2011. Many sources of evidence are used in this algorithm that identify coding regions and must be combined to get enough information to predict an exon or intron [46]. In this work a weight matrix method (WMM) and some constraints in the gene structure are used to limit the search space. Here fitness function is calculated using site and content statistics based on in-frame hexamer frequency and positional weight matrix. As the dataset used here is of imbalance nature, therefore accuracy is not the correct parameter to evaluate the performance of the system. A k-fold cross-validation test is employed here to evaluate the performance. The experimental results show that the system achieves moderately good results at nucleotide level. By adding a little bit more flexibility to the system, it will be able to deal with many gene prediction issues: alternative splicing, noncanonical functional sites, ignored stop codons, and pseudogenes. The performance of the system is not up to the

Table 1: Summary of soft computing techniques for protein-coding gene prediction.

Soft computing technique used	Organism (datasets used)	Program (URLs wherever available)	Prediction type
Back-propagation NN[1]	Human, mouse, arabidopsis, drosophila, rice (*GenBank* [52])	GRAIL-I [21] http://compbio.ornl.gov/grailexp	Exons
Back-propagation NN	Human, vertebrates (*GenBank*)	GeneParser [24] http://beagle.colorado.edu/~eesnyder/GeneParser.html	Exons, introns
Back-propagation NN	Human, mouse (*GenBank*)	GRAIL-II [22]	Exons
Back-propagation NN	Human, mouse, plant (*GenBank*)	CODEX [26]	Exons
Back-propagation NN	Vertebrates (*GenBank*)	GIN [29] http://www.bork.emblheidelberg.de/fmilpetz/GIN/	Exons
Back-propagation NN	*S. cerevisiae* genome (*MIPS* [53])	MLFANN (yeast genome) [32]	Open reading frames
Back-propagation ANN	Streptococcus pyogenes M group A Streptococcus strains (*GenBank*)	SpyMGASLacGenePred [35]	Open reading frames
Self-organizing map NN	*E. coli, B. subtilis, H. influenza*, Buchnera, *B. burgdorferi, M. jannaschii, M. genitalium, H. pylori, A. aeolicus*, Synechocystis, *Y. pestis, D. radiodurans, R. solanacearum, S. coelicolor, C. jejuni* (*GenBank*)	RescueNet [34] http://bioinf.nuigalway.ie/RescueNet/	Gene-coding region (prokaryotes)
Multilayer perceptron NN	Microbial Genome (*DEG* [54] *NCBI* [55])	EG-MLP (microbial genome) [37]	Genes
GA[2]	Human genome (*GenBank*)	Evolutionary algorithm [46]	Exons, introns
NN + GA	*E. coli* (*PromEC [56], Wisconsin-Madison* [57])	MultiNNProm [48]	Promoters
NN + GA	Arabidopsis, *E. coli*, human, mouse, rat (*GenBank and HMR195* [58])	RBFN-combining [49]	Exons

[1]NN (Neural Network), [2]GA (Genetic Algorithms).

mark, but it proves the validity of genetic algorithm as a tool in gene prediction. The evolved neural network mentioned in the previous section also makes use of genetic algorithm for the optimization of the neural network.

3.3. Hybrid Systems. Hybrid system integrates two or more technologies to solve a problem for example neural network combined with GA or neural network combined with fuzzy logic. Fuzzy logic is based on multivalued logic that allows multiple values to be defined between conventional values like 0 and 1. It provides a technique to deal with imprecision and uncertainty [16]. The main idea behind fuzzy logic is to approximate human decision making by using natural language terms instead of quantitative terms [47]. Some common examples of hybrid systems are neurofuzzy and neurogenetic. In neuro-fuzzy systems fuzzy input is provided to the neural network. In neuro-genetic systems neural network calls a genetic algorithm to optimize its structural parameters [44].

Each gene in a DNA sequence is preceded by promoter sequence. Successful identification of the location of promoter regions in DNA sequence leads to the prediction of the corresponding genes [48]. A neural network-based multiclassifier system for the prediction of *E. coli* promoter sequence was developed in 2005. In *E. coli* sequences promoters are located immediately before *E. coli* genes. The promoter sequences are encoded using four different encoding methods, which are used to train four different neural networks. The use of different encoding methods helps the multiclassifier network to specialize in different types of promoters present in the sequences. In this technique an aggregation function is used, to combine the individual results of the neural networks. To determine the weights of this aggregation

Table 2: Performance of soft computing techniques for protein-coding gene prediction.

Program (soft computing technique used)	Exon level		Nucleotide level	
	Sensitivity (ESn)	Specificity (ESp)	Sensitivity (Sn)	Specificity (Sp)
GRAIL-I (NN[1])	53%	90%	—	—
GeneParser (NN)	—	—	83%	83%
GRAIL-II (NN)	89%	91%	91%	90%
CODEX (NN)	72%	89%	—	—
GIN (NN)	78%	80%	92%	99%
MLFANN (NN)	—	—	96.65%	96.18%
SpyMGASLacGenePred (NN)	—	—	100%	76.90%
RescueNet (NN)	—	—	89.39%	89.04%
EG-MLP (NN)	—	—	79%	78%
Evolutionary algorithm (GA[2])	—	—	43%	66%
MultiNNProm (NN + GA)	—	—	98%	97%
RBFN-combining (NN + GA)	77%	79%	89%	90%

[1]NN (Neural network), [2]GA (Genetic algorithms).

function genetic algorithm is used. Genetic algorithm gives optimal set of weights by using the classification accuracy of the combined classifier as a fitness value. The main advantage obtained by combining multiple classifiers is that the other classifiers will recognize the genes not recognized by one classifier. The main difficulty with the proposed approach is in obtaining optimal configurations for the neural networks. The results prove that the performance of the multiclassifier system is better than the individual performances of the neural networks.

A common approach in gene prediction is to combine the results of several existing gene prediction programs to predict genes with better accuracy. These systems are called combiners and their performance is better than individual gene prediction programs. A novel method for predicting genes by combining the prediction results of three gene-finding program was developed in 2007. The main motivation behind this work is to improve the prediction accuracy at exon level [49]. In this method high-prediction gene-finding tools: GENSCAN, HMMgene, and Glimmer are combined using artificial neural network. Genetic algorithm here is used to calculate the equitable weighted parameters. Integrative evaluation of the technique is done using radial basis function network (RBFN). The experimental results show that the proposed method is effective in combining gene-finding programs and achieves higher accuracy at exon level than the single gene prediction tool.

An effective approach based on fuzzy neural network with structure learning (FNNSL) was developed in 2010 for ncRNA gene prediction. In this method four features are used for making predictions: the mean pairwise identity score (MPI), the structure conservation index (SCI), mean of normalized measures for thermodynamic stability, and the number of sequences in the alignment. The structure-learning algorithm is used here to enhance computational efficiency and to avoid overlearning [50]. The proposed system takes advantage of both the learning capability of the neural networks and the approximate reasoning capability of fuzzy logic. The experimental results validate the effectiveness of this hybrid approach with improved accuracy.

4. Analysis of Protein-Coding Gene Prediction Techniques

A theoretical analysis of protein-coding gene prediction techniques is presented in Table 1. The techniques are analyzed on the basis of prediction type, organism, and dataset used. It is very difficult to evaluate the performance of gene prediction techniques on the basis of a single parameter. Moreover, the performance comparison of these techniques is not possible because each technique is designed for a specific genome. Here the performance of these techniques is analyzed on the basis of two widely used parameters: sensitivity and specificity. The accuracy of prediction can be measured at three different levels: nucleotide level, exon level, and gene level. Very few techniques predicted complete gene structure. In this paper nucleotide- and exon-level accuracies are considered to measure the performance of gene prediction techniques. Nucleotide-level accuracy gives a measure of prediction in terms of content ability and exon level accuracy gives a measure of prediction in terms of signal ability [51]. Sensitivity and specificity at nucleotide level are defined as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{Specificity} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

where TP is the true positives, FP is the false positives, and FN is the false negatives.

Due to the inaccessibility of these techniques for performance evaluation, the sensitivity and specificity is calculated on the basis of the values given in their respective publications. These results are given in Table 2. The results obtained show that most of the techniques have higher sensitivity and specificity at nucleotide level than at exon level.

5. Conclusion

In this paper, the applications of soft computing techniques in the field of gene prediction are discussed. Soft computing techniques, especially neural networks, appear to be a powerful tool in gene prediction. It seems to be an ideal technique for combining multiple sources of information. But the success of neural networks as a gene prediction technique mainly depends on the type of information that is used as an input. Genetic algorithms and hybrid techniques give promising results but they are applied in a very limited fashion. Even though the current soft computing techniques are very helpful in identifying protein-coding and ncRNA genes, the output results are still far from being perfect as most of the work is done for specific genomes. In future techniques like fuzzy logic, genetic algorithms, neuro-fuzzy, and neuro-genetic need to be explored. Neural networks can be combined with traditional gene prediction techniques like hidden Markov model to achieve better results. As ncRNA gene prediction is a promising area of research, it can be further explored using these techniques.

References

[1] S. Bandyopadhyay, U. Maulik, and D. Roy, "Gene Identification: classical and computational intelligence approaches," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 38, pp. 55–68, 2008.

[2] M. McElwain, "A critical review of gene prediction software," *BioClinica*, vol. 218, pp. 1–10, 2007.

[3] W. Li, "A list of gene prediction softwares," [online] available: http://www.geneprediction.org/software.html.

[4] J. M. Claverie, "Computational methods for the identification of genes in vertebrate genomic sequences," *Human Molecular Genetics*, vol. 6, no. 10, pp. 1735–1744, 1997.

[5] A. krogh, "Gene finding: putting the parts together," in *Guide to Human Genome Computing*, M. Bishop, Ed., pp. 261–274, Academic Press, 2nd edition, 1998.

[6] G. D. Stormo, "Gene-finding approaches for eukaryotes," *Genome Research*, vol. 10, no. 4, pp. 394–397, 2000.

[7] R. Sarker, S. Bandyopadhyay, and U. Maulik, "An overview of computational approaches for prediction of miRNA genes and their targets," *Current Bioinformatics*, vol. 6, no. 1, pp. 129–143, 2011.

[8] J. C. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*, PWS, Boston, Mass, USA, 1996.

[9] R. D. Sleator, "An overview of the current status of eukaryote gene prediction strategies," *Gene*, vol. 461, no. 1-2, pp. 1–4, 2010.

[10] Z. Wang, Y. Chen, and Y. Li, "A brief review of computational gene prediction methods," *Genomics, Proteomics & Bioinformatics*, vol. 2, pp. 216–221, 2004.

[11] K. Davies, *Eukaryotic Gene Prediction*, 2009.

[12] M. J. Schellenberg, D. B. Ritchie, and A. M. MacMillan, "Pre-mRNA splicing: a complex picture in higher definition," *Trends in Biochemical Sciences*, vol. 33, no. 6, pp. 243–246, 2008.

[13] D. Haussler, "Computational genefinding," *Trends in Biochemical Sciences*, vol. 23, pp. 12–15, 1998.

[14] C. Mathe, M. F. Sagot, T. Schiex, and P. Rouze, "Current methods of gene prediction, their strengths and weaknesses," *Nucleic Acids Research*, vol. 30, pp. 4103–4117, 2002.

[15] M. Yandell and D. Ence, "A beginner's guide to eukaryotic genome annotation," *Nature Reviews*, vol. 13, pp. 329–342, 2012.

[16] J. S. R. Jang, C. T. Sun, and E. Mizulani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, 1996.

[17] A. B. Kurhe, S. S. Satonkar, P. B. Khanale, and S. Ashok, "Soft computing and its applications," *BIOINFO Soft Computing*, vol. 1, pp. 5–7, 2011.

[18] S. Rajasekaran and G. A. V. Pai, *Neural Network, Fuzzy Logic and Genetic Algorithms- Synthesis and Applications*, Prentice-Hall, 2005.

[19] S. S. Ray, S. Bandyopadhyay, P. Mitra, and S. K. Pal, "Bioinformatics in neurocomputing framework," *IEEE Proceedings Circuits, Devices & Systems*, vol. 152, pp. 556–564, 2005.

[20] C. H. Wu, "Artificial neural networks for molecular sequence analysis," *Computers and Chemistry*, vol. 21, no. 4, pp. 237–256, 1997.

[21] E. C. Uberbacher and R. J. Mural, "Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 88, no. 24, pp. 11261–11265, 1991.

[22] Y. Xu, J. R. Einstein, R. J. Mural, M. Shah, and E. C. Uberbacher, "An improved system for exon recognition and gene modeling in human DNA sequences," in *Proceedings of the 16th Annual International Conference Intelligent Systems for Molecular Biology*, pp. 376–383, 1994.

[23] E. E. Snyder and G. D. Stormo, "Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks," *Nucleic Acids Research*, vol. 21, no. 3, pp. 607–613, 1993.

[24] E. E. Snyder and G. D. Stormo, "Identification of protein coding regions in genomic DNA," *Journal of Molecular Biology*, vol. 248, no. 1, pp. 1–18, 1995.

[25] M. Burset and R. Guigó, "Evaluation of gene structure prediction programs," *Genomics*, vol. 34, no. 3, pp. 353–367, 1996.

[26] L. Roberts, N. Steele, C. Reeves, and G. J. King, "Training neural networks to identify coding regions in genomic DNA," in *Proceedings of the 4th International Conference on Artificial Neural Networks*, pp. 399–403, June 1995.

[27] X. U. Ying, R. J. Mural, J. R. Einstein, M. B. Shah, and E. C. Uberbacher, "GRAIL: a multi-agent neural network system for gene identification," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1544–1551, 1996.

[28] Y. Xu and E. C. Uberbacher, "Gene prediction by pattern recognition and homology search," in *Proceedings of the 20th Annual International Conference on Intelligent Systems for Molecular Biology*, pp. 241–251, AAAI Press, 1996.

[29] Y. Cai and P. Bork, "Homology-based gene prediction using neural nets," *Analytical Biochemistry*, vol. 265, no. 2, pp. 269–274, 1998.

[30] R. Guigo, S. Knudsen, N. Drake, and T. Smith, "Prediction of gene structure," *Journal of Molecular Biology*, vol. 226, no. 1, pp. 141–157, 1992.

[31] C. Burge and S. Karlin, "Prediction of complete gene structures in human genomic DNA," *Journal of Molecular Biology*, vol. 268, no. 1, pp. 78–94, 1997.

[32] C. Li, P. He, and J. Wang, "Artificial neural network method for predicting protein-coding genes in the yeast genome," *Internet Electronic Journal of Molecular Design*, vol. 2, pp. 527–538, 2003.

[33] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[34] S. Mahony, J. O. McInerney, T. J. Smith, and A. Golden, "Gene prediction using the Self-Organizing Map: automatic generation of multiple gene models," *BMC Bioinformatics*, vol. 5, article 23, pp. 1–9, 2004.

[35] S. Rebello, U. Maheshwari, Safreena, and R. V. Dsouza, "Back propagation neural network method for predicting lac gene structure in streptococcus pyogenes M group A streptococcus strains," *International Journal for Biotechnology and Molecular Biology Research*, vol. 2, pp. 61–72, 2011.

[36] E. V. Koonin, "How many genes can make a cell: the minimal-gene-set concept," *Annual Review of Genomics and Human Genetics*, vol. 1, pp. 99–116, 2000.

[37] K. Palaniappan and S. Mukherjee, "Predicting "essential" genes across microbial genomes: a machine learning approach," in *Proceedings of the IEEE International Conference on Machine Learning and Applications*, pp. 189–194, 2011.

[38] J. S. Mattick, "Non-coding RNAs: the architects of eukaryotic complexity," *EMBO Reports*, vol. 2, no. 11, pp. 986–991, 2001.

[39] R. J. Carter, I. Dubchak, and S. Holbrook, "A computational approach to identify genes for functional RNAs in genomic sequences," *Nucleic Acids Research*, vol. 29, pp. 3928–3938, 2001.

[40] M. Cheung and G. B. Fogel, "Identification of functional RNA genes using evolved neural networks," in *Proceedings of the IEEE symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 1–7, November 2005.

[41] R. C. Lee and V. Ambros, "An extensive class of small RNAs in Caenorhabditis elegans," *Science*, vol. 294, no. 5543, pp. 862–864, 2001.

[42] N. C. Lau, L. P. Lim, E. G. Weinstein, and D. P. Bartel, "An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans," *Science*, vol. 294, no. 5543, pp. 858–862, 2001.

[43] Y. Zeng, E. J. Wagner, and B. R. Cullen, "Both natural and designed micro RNAs can inhibit the expression of cognate mRNAs when expressed in human cells," *Molecular Cell*, vol. 9, no. 6, pp. 1327–1333, 2002.

[44] R. C. Chakraborty, "Soft computing-introduction," 2010, http://www.myreaders.info/html/soft_computing.html.

[45] M. Mitchell, *An Introduction to Genetic Algorithm*, MIT Press, 1998.

[46] J. Perez-Rodriguez and N. Garcia-Pedrajas, "An evolutionary algorithm for gene structure prediction," *Industrial Engineering and Other Applications of Applied Intelligent Systems II*, vol. 6704, pp. 386–395, 2011.

[47] J. Bih, "Paradigm shift—an introduction to fuzzy logic," *IEEE Potentials*, vol. 25, no. 1, pp. 6–21, 2006.

[48] R. Ranawana and V. Palade, "A neural network based multi-classifier system for gene identification in DNA sequences," *Neural Computing and Applications*, vol. 14, no. 2, pp. 122–131, 2005.

[49] Y. Zhou, Y. Liang, C. Hu, L. Wang, and X. Shi, "An artificial neural network method for combining gene prediction based on equitable weights," *Neurocomputing*, vol. 71, no. 4–6, pp. 538–543, 2008.

[50] D. Song and Z. Deng, "A novel ncRNA gene prediction approach based on fuzzy neural networks with structure learning," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–5, June 2010.

[51] A. Nagar, S. Purushothaman, and H. Tawfik, "Evaluation and fuzzy classification of gene finding programs on human genome sequences," in *Proceedings of the 2nd International Confernce on Fuzzy Systems and Knowledge Discovery (FSKD '05)*, vol. 3614, pp. 821–829, August 2005.

[52] http://www.ncbi.nlm.nih.gov/genbank.

[53] "The munich information center for protein sequences (MIPS)," 2001, http://mips.gsf.de.

[54] "DEG 5. 0., a database of essential genes in both prokaryotes and eukaryotes," 2008, http://tubic.tju.edu.cn/deg.

[55] "NCBI Microbial genomes," 2009, http://www.ncbi.nlm.nih.gov/genomes/Iproks.cgi?view=1.

[56] http://bioinfo.md.huji.ac.il/marg/promec.

[57] http://www.genome.wisc.edu/sequencing/K12.htm#seq.

[58] http://www.cs.ubc.ca/labs/beta/genefinding/.