Analytical Biochemistry 438 (2013) 14–21

Contents lists available at SciVerse ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

Review

A comparative analysis of soft computing techniques for gene prediction

Neelam Goel *, Shailendra Singh, Trilok Chand Aseri

Department of Computer Science and Engineering, PEC University of Technology, Sector-12, Chandigarh 160 012, UT, India

ARTICLE INFO

Article history:
Received 27 July 2012
Received in revised form 5 March 2013
Accepted 14 March 2013
Available online 22 March 2013

Keywords:
Gene prediction
Neural networks
Genetic algorithm
Fuzzy logic
Splice site
Protein coding regions

ABSTRACT

The rapid growth of genomic sequence data for both human and nonhuman species has made analyzing these sequences, especially predicting genes in them, very important and is currently the focus of many research efforts. Beside its scientific interest in the molecular biology and genomics community, gene prediction is of considerable importance in human health and medicine. A variety of gene prediction techniques have been developed for eukaryotes over the past few years. This article reviews and analyzes the application of certain soft computing techniques in gene prediction. First, the problem of gene prediction and its challenges are described. These are followed by different soft computing techniques along with their application to gene prediction. In addition, a comparative analysis of different soft computing techniques for gene prediction is given. Finally some limitations of the current research activities and future research directions are provided.

During the past decades, various genomes have been sequenced in both plants and animals. With the development of genomic sequence data, genome annotation has become important in bioinformatics. Genome annotation helps in understanding the biological functions of the sequences of these genomes. Gene prediction is one of the most essential aspects of genome annotation. Since the time the Human Genome Project started, the database of DNA sequences has been increasing exponentially [1,2]. The sequence data is growing rapidly, and our ability to predict genes in them has lagged behind significantly [3]. The process of predicting genes using experimentation methods is very slow and time-consuming. Thus, the development of reliable automated techniques for predicting genes in uncharacterized genomic sequences became significant [4]. The problem of gene prediction consists mainly of identifying protein coding regions in genomic DNA, but it may also include the identification of other functional regions such as RNA coding and regulatory regions. Various gene prediction techniques have been developed during the past several years. The current techniques of gene prediction are considerably more accurate, reliable, and useful than those available in the past. However, the performance of these gene prediction methods is still below the expected level. The main problems of the existing gene prediction techniques are genome dependency and gene-level accuracy. Most of the techniques are developed for specific genomes, and the gene-level accuracy of these techniques is very low. It is obvious that further improvement to gene prediction is much needed. An extensive list of existing gene prediction programs can be found in Ref. [5].

Most of the previous reviews written on this problem have focused on traditional gene prediction techniques such as the hidden Markov model (HMM),[1] dynamic programming (DP), and decision trees [6–8]. In addition to these traditional gene prediction techniques, approaches based on soft computing have gained popularity during recent times. A review on traditional and computational intelligence techniques is presented in Ref. [9]. Soft computing techniques can work well for gene prediction due to their ability to handle uncertainty and noise in the sequence data. However, none of the reviews has focused on soft computing techniques for gene prediction during the past few years. The main focus of this article is to provide a comparative analysis of soft computing techniques for gene prediction.

The article outline is as follows. The next section describes the basic terminology related to the problem of gene prediction along with its challenges and the types of information used by gene prediction techniques. Different soft computing techniques used in the field of gene prediction are then discussed, followed by a comparative analysis of these techniques. Finally, some conclusions and future research directions are presented.

* Corresponding author. Address: #1060/B-2, Vishvakarma Colony, Pinjore-134102, District Panchkula, Haryana, India.
E-mail address: neelam.goyal85@gmail.com (N. Goel).

[1] *Abbreviations used:* HMM, hidden Markov model; DP, dynamic programming; A, adenine; T, thymine; G, guanine; C, cytosine; ORF, open reading frame; EST, expressed sequence tag; ANN, artificial neural network; HOMM, higher order Markov model; IMCM, inhomogeneous Markov chain model; GIN, gene identification using neural nets and homology information; RBFN, radial basis function network; ANFIS, adaptive network-based fuzzy inference system; GRNN, generalized regression neural network; ncRNA, noncoding RNA.

0003-2697/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ab.2013.03.015

Background

This section provides the basic terminology related to gene prediction as well as some of the challenges involved in it. The problem of gene prediction is then described, followed by types of information used by gene prediction techniques.

Basic terminology

Proteins are considered as the building blocks of life. A protein is a chain of simpler molecules called *amino acids* linked by polypeptide bonds. The information necessary to build protein in an organism is encoded in DNA. For this reason, DNA is referred to as the "blueprint of life" [10]. All organisms self-replicate due to the presence of genetic material in DNA [11]. DNA exists in nearly every cell of an organism and is contained in a larger structure known as a chromosome. The number of chromosomes in each cell depends on the species and can vary to a large degree. For example, a human has 46 chromosomes, whereas *Drosophila melanogaster* has only 8. The entire DNA content of a cell is known as the *genome*.

DNA can be viewed as a sequence of organic molecules called nucleotides. A nucleotide is made up of a base and a sugar linked to it. There are four different nucleotides in DNA, each differing in its base. These four bases are adenine (A), thymine (T), guanine (G) and cytosine (C). In DNA sequence, A pairs with T and G pairs with C due to the presence of hydrogen bonds. The nucleotide bases are classified into two types: purines and pyrimidines. Adenine and guanine are called purines, and cytosine and thymine are called pyrimidines. The DNA molecule is made up of two complementary strands of nucleotides wound in a double helix. One of the strands in double helix structure is called the sense strand, and other is called the antisense strand. The antisense strand is the one that is generally transcribed. In a double helix, the direction of the nucleotides on one strand is opposite to their direction on the complementary strand; therefore, they are said to be anti-parallel to one another. The ends of DNA strand are referred to as 5′ and 3′ ends.

A gene is a segment of DNA that codes for either protein or noncoding RNA. The part of DNA that does not contain any gene is known as noncoding or intergenic regions. Intergenic regions are the regions of DNA between genes. Introns are the noncoding segments of DNA found within genes. After the DNA is transcribed into RNA, the introns are spliced out from it to form mature RNA. The rest of the coding segments of RNA are known as exons. The exon/intron boundary is usually referred to as splice sites. The mRNA is further converted into protein. The process by which information is extracted from DNA to make protein is called central dogma of molecular biology.

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 noncoding RNA genes. This article deals only with protein-coding gene prediction techniques. 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, or noncoding or intergenic region. The problem of gene prediction can be formally stated as follows [9]:

Input: A sequence of DNA

$$X = (x_1, \cdots, x_n) \in \sum^{*}, \text{ where } \sum = \{A, T, C, G\}$$

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

All living organisms in this world fall into one of two categories: eukaryotes or prokaryotes. 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 are termed as introns. In eukaryotes, only 3% of human DNA sequence is coding [3]. Gene prediction in prokaryotes is less difficult due to higher gene density and the absence of introns in their protein coding regions [1]. The main difficulty in prokaryotic gene prediction is due to the presence of overlapping regions [12]. The process is more complex for eukaryotes due to large genome size, and exons are interrupted by introns. Furthermore, in eukaryotes, coding sequences are subject to alternative splicing, that is, a process of reconnecting exons in multiple ways during RNA splicing [13]. Indeed, it is estimated that more than 95% of human genes show evidence of at least one alternative splice site [3].

Numerous gene prediction methods have been developed for both eukaryotes and prokaryotes. In this article, the soft computing techniques of gene prediction are discussed for eukaryotes only. Although gene prediction techniques in eukaryotes have achieved a significant level of accuracy, there are many challenges that still need to be resolved. The major challenges in eukaryotic gene prediction are as follows:

- Prediction of short exons, especially those bordered by large introns
- The exact boundaries of exons and their assemblies into complete genes
- The exact number of genes in human genome being unknown [14]
- Alternative splicing
- Reliance on known sequences
- Presence of overlapping genes
- Large proportion of human genome being composed of noncoding RNA
- The possibility of sequences stored in databases containing error
- Prediction of partial and multiple genes
- Noncanonical splice sites (splice sites other than those based on GT–AG dinucleotides)
- Prediction of genes in newly sequenced genome.

Types of information used

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 technique used to combine that information into a reliable prediction [8]. This information is generally divided into content sensors and signal sensors [4]. This subsection is devoted to the types of information used in predicting the gene structure:

- *Content sensors:* Content sensors are measures that try to classify a DNA region into coding and noncoding. The existence of a sufficient similarity with a biologically characterized sequence can also be used as a means for predicting coding and noncoding regions. Content sensors are further classified as extrinsic or intrinsic. The *extrinsic* content sensors take a genomic DNA sequence and calculate its similarity to a protein or DNA sequence present in the database in order to determine whether the region is coding or noncoding. Similarities with three different types of sequences can be used to find information about coding regions: protein sequence, cDNA or expressed sequence tag (EST) sequence, and DNA sequence. With *intrinsic* content sensors, coding regions have statistical properties (i.e., asymmetries and

periodicities) that help intrinsic content sensors to distinguish them from noncoding regions. Some of the common measures used include codon (a triplet of DNA bases) usage, GC content, nucleotide composition, hexamer frequency, and base occurrence periodicity.

- *Signal sensors:* The basic and natural approach of finding the presence of functional sites is signal sensors. 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. Local sites such as these are called signals, and methods used to detect them are known as signal or site detection methods.

The methods that use signal sensors or both signal and intrinsic content sensors are known as ab initio methods of gene prediction. During the past 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 that of methods based purely on ab initio approaches [15].

This section has presented the background knowledge for gene prediction. The soft computing techniques for gene prediction are described in the subsequent section.

Soft computing techniques for gene prediction

Soft computing is the modern approach to constructing a computationally intelligent system. Soft computing is the blend of methodologies that provides flexible information processing capabilities for handling real-world problems [16]. 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 need a lot of computational time [17]. Unlike hard computing methods, soft computing methods cope with those problems that deal with imprecision, uncertainty, learning, and approximation to achieve tractability, robustness, low-cost solutions and human-like decision making [18].

Certain properties of soft computing techniques make them suitable for sequencing tasks. These techniques can be easily adapted to changing circumstances. They are able to handle very large data sets with missing and noisy data and can be used to extract hidden relationships from these data. One unique property of soft computing is that it is deeply involved in learning from experimental data, making it suitable for gene prediction. While predicting genes, specific patterns in DNA sequence are recognized and soft computing techniques have been used extensively in pattern recognition problems [19]. Soft computing consists of several techniques, with the most important being neural networks, genetic algorithms, and fuzzy logic. The importance of soft computing techniques lies in the fact that they are complementary, not competitive. In many cases, a problem can be solved by using a neural network, fuzzy logic, and genetic algorithm in combination rather than one technique exclusively. This section describes the application of these soft computing techniques in the area of gene prediction.

Artificial neural networks

An artificial neural network (ANN) is an information processing model that is used to represent complex input–output relationships. The main aim behind the development of neural networks is to acquire human ability to deal with the changing environment. An ANN is an interconnected group of artificial neurons [18]. The structure of a neural network is represented as multiple layers of neurons operating in parallel to solve specific problems. The main characteristic of ANNs is their ability to learn from examples and generalize this learning beyond the examples supplied. A 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 them suitable for predicting genes in DNA sequence. Neural networks can be divided into different architectures on the basis of learning algorithms [20]. Various neural network architectures employed for splice site and gene prediction are discussed here.

Splice site prediction

The most common functional sites in DNA sequence are splice sites because they define exon/intron boundaries and, thus, define the exact content of coding regions. Neural networks have been used successfully for splice site prediction during the past two decades. One of the earliest attempts at splice site prediction using neural networks is described in Ref. [21]. The method is based on the statistical technique of discriminant analysis. The perceptron algorithm is represented in the form of a variable that can be used in the discrimination function. The results show that a combination of methods within the discrimination analysis framework provides a reliable method for splice junction prediction. Another method that demonstrates use of the perceptron algorithm is described in Ref. [22]. This method uses base composition surrounding the splice sites as a means to preprocess the sequences that further can be fed into neural networks in encoded form. The method does not work in the case of noncanonical splice sites.

The aforementioned methods are based on local sequence information. A new method called NetGene, which combines both local and global sequence information in neural networks, is presented in Ref. [23]. Here a joint prediction approach, where prediction of transition regions between coding and noncoding helps in splice site assignment, is applied. The resulting method obtained better results than the methods that used only local information. The method does not use the constraints of ORFs in the selection of compatible splice sites. This would have helped in reducing the false positive predictions. This method is extended further to predict splice sites in plants. The new system, NetPlantGene [24], provides better results than NetGene. An enhanced prediction system called NetGene2, which incorporates branch point consensus to improve acceptor site prediction, is reported in Ref. [25]. The system also reduces false positive predictions.

A method that takes into account pairwise correlation of the dinucleotides at the splice site consensus is described in Ref. [26]. The tool NNSplice proposed here is incorporated into a gene-finding system called Genie. The resulting system reduces false negatives and improves the overall gene prediction performance. The predictions of this new version of Genie are better than those of the old version [27], which uses neural networks for splice site prediction in a manner similar to that of NetGene. No attempt is made here to predict noncanonical splice sites. Another approach based on a hierarchical neural network simulator is presented in Ref. [28]. The proposed method also analyzes the effect of point mutation on splicing. Although the system succeeds in extracting the particular features of the splice sites to some degree, it does not obtain the explicit expressions of the features.

A neural network-based hybrid approach to predict splice sites is proposed in Ref. [29]. The BRAIN algorithm used here infers Boolean formulas from examples and considers splicing rules as disjunctive normal form (DNF) formulas. The predictions of this algorithm are refined by a neural network and combined using a discriminant analysis procedure. The proposed method confirms low error rates and shows better results than other stand-alone methods. The higher order Markov model (HOMM) can be implemented using a neural network because of its ability to learn complex interactions of

nucleotides through nonlinear mapping. A novel hybrid technique based on this concept is described in Ref. [30]. To implement this, Markov encoding is used at the first stage and the obtained conditional probabilities are fed into the neural network. The method outperforms other splice site detection programs developed during that time. The model is further used to predict transcription and translation initiation sites [31].

The technique used to encode neural network input plays an important role in splice site prediction. Most of the methods mentioned above are based on the orthonormal encoding method (OEM). A complementary encoding method (CEM) for splice site prediction is illustrated in Ref. [32]. The new approach considerably reduces the false positive predictions and training time of the neural networks. Another hybrid technique based on the inhomogeneous Markov chain model (IMCM) and neural network is presented in Ref. [33]. The sequence data are preprocessed using an inhomogeneous Markov chain before feeding into the neural network. The proposed technique requires less computation and outperforms all other splice site prediction methods with prediction accuracy greater than 98%. The neural network can also predict splice site locations without prior knowledge of signals such as GT and AG. Such a technique designed for plants is discussed in Ref. [34]. The input is encoded using orthonormal encoding. This method achieves a significant level of accuracy but not better accuracy than other popular techniques.

A summary of neural network-based splice site prediction methods presented above is provided in Table 1. The table enlists different splice site prediction methods, their references, the training algorithm used by these methods, the years of their development, organisms for which they train, the information used, and the available links. Most of the methods discussed in this section use the feedforward neural network trained using the back-propagation algorithm. Some more work to predict splice sites using different neural network architectures is described in Refs. [35–37]. Discussing the literature related to other functional sites, such as transcription and translation, is beyond the scope of this article and is summarized in Ref. [38].

Coding region and gene prediction

The first attempt to locate coding regions of genes using neural networks is described in Refs. [39,40]. The method combines the output of seven sensor algorithms into a neural network from which coding regions are identified. The coding recognition module (CRM) of Grail uses a sequence window of fixed length. The gene modeling module used by this system is presented in Ref. [41]. The results of this technique are quite promising, but it easily misses short exons. The performance of Grail is improved further by using a variable length sequence window that results in a new system known as Grail II [42–44].

A system based on the Grail framework to predict genes in *Drosophila* is illustrated in Ref. [45]. The latest version GrailEXP [46] has incorporated homology information from database sequences to improve the prediction results of Grail. This method is further extended in Ref. [47] to predict multiple gene structures based on ESTs. A similar system called Codex [48], based on Grail, has been developed to improve the prediction performance. Unlike Grail, it uses a series of neural networks for the prediction of exons. The technique helps in reducing the number of false predictions, but it classifies some sequences as "don't know." Moreover, it does not predict the complete gene structure. An attempt to discriminate coding regions from noncoding regions based on di-codon statistics is investigated in Ref. [49]. The system shows higher accuracy on small coding segments but does not recognize complete exons.

An approach to predict coding regions by combining neural networks and dynamic programming is discussed in Ref. [50]. The

Table 1
Neural network-based splice site predictors.

Method [reference(s)]	Neural network algorithm used (year)	Organism	Type of information used	Link (where available)
Discrimination based [21]	Perceptron algorithm (1985)	Human	Consensus sequence patterns, base composition, and periodicity of coding and noncoding regions, free energy of small nuclear RNA and mRNA base pairing	–
Perceptron-based neural network [22]	Perceptron algorithm (1988)	Human	Base composition surrounding the splice sites	–
NetGene [23]	Back-propagation algorithm (1991)	Human	Consensus sequence around splice sites and base composition of coding and noncoding regions	–
NetPlantGene [24]	Back-propagation algorithm (1996)	*Arabidopsis thaliana*	Consensus sequence around splice sites and base composition of coding and noncoding regions and sequence based rules	http://www.cbs.dtu.dk/services/NetPGene
NetGene2 [25]	Back-propagation algorithm (1997)	Human, *A. thaliana*, *Caenorhabditis elegans*	Consensus sequence around splice sites and base composition of coding and noncoding regions, sequence-based rules, and branch point consensus sequence	http://www.cbs.dtu.dk/services/NetGene2
NNSplice 0.9 [26]	Back-propagation algorithm (1997)	*D. melanogaster*, human, other	Pairwise correlation of dinucleotides at the splice site consensus	http://www.fruitfly.org/seq_tools/splice.html
Hierarchical neural network simulator [28]	Back-propagation algorithm (1997)	Mammals	Consensus splice site sequences	–
BRAIN [29]	Perceptron algorithm (1998)	Primate	Formulas inferred from consensus splice site sequences	ftp://ftp.ebi.ac.uk/pub/softwares/dos under nnbrain$.exe
HOMM-based neural network [30,31]	Back-propagation algorithm (2003)	Human	Consensus sequence around splice sites and composition of coding and noncoding regions	–
CEM-based neural network [32]	Back-propagation algorithm (2005)	Human	Consensus splice site sequences	–
IMCM-based neural network [33]	Back-propagation algorithm (2007)	Primates	Consensus splice site sequences	–
Feedforward neural network [34]	Back-propagation algorithm (2009)	*A. thaliana*	DNA sequences	–

main aim behind the development of GeneParser is error tolerance. The method perform well for short internal exons but is unable to predict terminal exons. The problem of terminal exons is solved in the enhanced version of GeneParser [51]. Another attempt to discriminate coding regions of eukaryotic genes using neural networks is reported in Ref. [52]. Various organisms' sequences have been taken to build a discrimination model. The performance of this method depends on coding length.

A new method based on a modular system of neural networks to identify eukaryotic gene structure is proposed in Ref. [53]. The prediction task is divided into the detection of distinct signals based on different neural network architectures. A novel approach of using homology information for multiple gene prediction is presented in Ref. [54]. The technique is designed primarily to reduce false positive predictions. Although GIN (gene identification using neural nets and homology information) predicts genes with reasonable accuracy, it does not work well in the absence of homology information. A knowledge-based neural network system, Exon–ENet, is described in Ref. [55] for exon prediction. This system outperforms Grail on short DNA sequences. However, its performance is not very good on long DNA sequences. Most of the methods discussed above predict genes in human DNA sequences. A neural network-based method to predict coding regions in the yeast genome is reported in Ref. [56]. The prediction accuracy of this method is better than that of the previous method designed for yeast. Nearly all neural network approaches for gene prediction discussed above use the back-propagation algorithm to train DNA sequences.

Genetic algorithms

Genetic algorithms are heuristic search algorithms based on the process of natural evolution [57]. They are efficient, adaptive, and robust processes and have a large degree of parallelism. Therefore, genetic algorithms are suitable for solving those problems that need optimized, fast, and close approximate solutions. Furthermore, erroneous bioinformatics data can be handled with the robust property of genetic algorithms. They often encode candidate solutions as fixed length bit-strings called chromosomes [58]. Due to these characteristics, the use of genetic algorithms in the field of gene prediction seems appropriate.

A novel attempt based purely on a genetic algorithm for gene prediction is described in Ref. [59]. Here fitness function is calculated using site and content statistics based on in-frame hexamer frequency and positional weight matrix. The experimental results show that the system achieves moderately good results at the nucleotide level. By adding a little more flexibility to the system, it will be able to deal with many gene prediction issues such as alternative splicing, noncanonical functional sites, ignored stop codons, and pseudo-genes. The performance of the system is not up to the mark, but it proves the validity of genetic algorithms as a tool in gene prediction. An enhanced version of this method is reported in Ref. [60]. Here support vector machines are used for functional site prediction. The results obtained from this system are better than those from the previous approach. The proposed method has provided a new direction of applying genetic algorithm in the field of gene prediction.

Hybrid systems

A hybrid system integrates two or more techniques to solve a problem. The most common examples include a neural network combined with a genetic algorithm and a neural network combined with fuzzy logic. Fuzzy logic is a relatively new technique based on multi-valued logic that allows multiple values to be defined between conventional values such as 0 and 1. It provides a method to deal with imprecision and uncertainty [61]. The main idea behind fuzzy logic is to approximate human decision making by using natural language terms instead of quantitative terms [62]. One of the biggest advantages of fuzzy logic is that it simplifies complex systems. Hybrid systems are more popularly known as neuro-fuzzy and neuro-genetic. In neuro-fuzzy systems, fuzzy input is provided to the neural network. In neuro-genetic systems, a neural network calls a genetic algorithm to optimize its structural parameters. Neuro-genetic systems have been used in gene prediction during the past 10 years. However, neuro-fuzzy systems have been observed more recently.

Neuro-genetic systems

As mentioned previously, an exon recognition method generally includes both signal and content sensors. This approach is used in an evolved ANN [63], a system designed to predict coding and noncoding regions in DNA sequences. In an evolved ANN, genetic algorithms are used to determine appropriate parameters for neural network structure. The neural network is provided with nine inputs based on different coding measures. Other structural parameters of the network such as interconnection weights are evolved using genetic algorithms. Finally, the output of the neural network is refined in a postprocessing step to predict exons. The performance of this neuro-genetic system is better than that of Grail, which uses only a neural network to predict exons.

Many gene-finding programs have been developed during the past 20 years. A common approach since the early days of gene prediction is to combine the results of several existing gene-finding tools to achieve high prediction accuracy. A recent method called RBFN (radial basis function network) Combining, which combines the predictions of three popular gene finding tools (GENSCAN [64,65], HMMgene [66,67], and Glimmer [68]), is proposed in Ref. [69]. An ANN is used here to combine the accuracy parameters of these tools. Using a genetic algorithm, the equitable weighted parameters of the RBFN is calculated. Finally, the integrative evaluation of the tools is done with the help of a trained neural network. The results show that the proposed method is effective in combining gene-finding programs and achieves higher accuracy at the exon level than as a single gene prediction tool.

Neuro-fuzzy systems

A new approach to predict splice sites based on an adaptive network-based fuzzy inference system (ANFIS) is discussed in Ref. [70]. Here the sequence data are divided into three datasets using five preprocessing strategies: encoding the nucleotides, extracting the statistical properties, ignoring the low correlated features, normalizing the patterns, and reducing the redundant features. Finally, the network is trained using different learning algorithms. The ANFIS outperforms well-known classification algorithms. A recent attempt of using the neuro-fuzzy network to predict splice sites is presented in Ref. [71]. The neuro-fuzzy network is also based on the ANFIS. In contrast to the previous approach, examples of true and false splice site sequences are used here to define fuzzy rules. The largest contribution of this method is to achieve high prediction accuracy using smaller neuro-fuzzy networks.

Comparative analysis of gene prediction techniques

In a recent study [72], the performance of different neural network architectures for splice site prediction is measured with a 10-fold cross-validation. The experiment is performed on perceptron, back-propagation network (BPN), knowledge base ANN (KBANN), multilayer perceptron (MLP), radial basis function (RBF), and generalized regression neural network (GRNN). The results of this experiment show that GRNN is more successful than other networks used for splice site prediction. The performance of six splice

Table 2
Performance of programs on HMR195 dataset.

Gene prediction program	Nucleotide level			Exon level			
	Sn	Sp	AC	ESn	ESp	ME	WE
RBFN combining	0.92	0.94	0.91	0.79	0.80	0.11	0.06
AUGUSTUS	0.94	0.90	0.90	0.73	0.82	0.17	0.08
FGENESH	0.95	0.94	0.94	0.83	0.84	0.07	0.07

Note. Sn, sensitivity; Sp, specificity; AC, approximate correlation; ESn, sensitivity; ESp, specificity; ME, missed exons; WE, wrong exons.

Table 3
Performance of programs by nucleotide on human sequences.

Gene prediction program	Nucleotide level		
	Sn	Sp	CC
FGENESH	0.93	0.93	0.92
AUGUSTUS	0.88	0.93	0.89
GENSCAN	0.94	0.89	0.9
HMMgene	0.88	0.9	0.87
GeneMark.hmm	0.87	0.89	0.87
GeneParser	0.71	0.72	0.68
Grail-I	0.56	0.85	0.65
Evolved ANN	0.74	0.38	0.46
Evolutionary algorithm	0.44	0.67	0.56

Note. Sn, sensitivity; Sp, specificity; CC, correlation coefficient.

Table 4
Performance of programs by exon on human sequences.

Gene prediction program	Exon level			
	ESn	ESp	ME	WE
FGENESH	0.81	0.8	0.09	0.11
AUGUSTUS	0.72	0.84	0.2	0.08
GENSCAN	0.78	0.74	0.08	0.14
GrailExp	0.9	0.93	0.06	0.008
Grail-II	0.77	0.78	0.07	0.06
Modular NN	0.88	0.83	0.11	0.17
Codex	0.81	0.9	0.19	0.09
GeneParser	0.69	0.63	0.31	0.37
Grail-I	0.59	0.91	0.4	0.09

Note. ESn, sensitivity; ESp, specificity; ME, missed exons; WE, wrong exons.

site tools (HMMgene, NetGene2, HSPL [73,74], NNSplice, SpliceView [75,76], and Gene–Id3 [77,78]) is analyzed in Ref. [79]. The results indicate that the programs that use both local and global coding information along with splice signals (HMMgene and NetGene2) perform better than the other four methods.

In a different evaluation, the accuracy of GeneSplicer [80] is compared with six leading splice site predictors (NetGene2, NetPlantGene, NNSplice, HSPL, GENIO [81–83], and SpliceView). For donor sites NetGene2 performs the best, whereas for acceptor sites GeneSplicer, NetGene2, and HSPL perform comparably. From these analyses, it is clear that neural network-based splice site predictors perform well. A number of splice predictors are evaluated in these studies. However, in practice it is very difficult to compare the performance of splice site and gene prediction tools. Prediction tools and their datasets are not all freely available. They are trained on different genomes. The accuracy is evaluated using different parameters.

In this study, a comparative analysis of soft computing-based gene prediction programs is performed. During this analysis, five programs are tested on sequences taken from the HMR195 dataset [84]. The performance evaluation parameters used in this study are the same as those defined in Ref. [85]. At the nucleotide level the performance is evaluated on the parameters Sn (sensitivity), Sp (specificity), AC (approximate correlation), and CC (correlation coefficient), whereas at the exon level performance is evaluated on ESn (sensitivity), Esp (specificity), ME (missed exons), and WE (wrong exons). First, the performance of RBFN Combining is compared against that of two popular gene prediction programs (AUGUSTUS [86–88] and FGENESH [89,90]). The programs are evaluated on all sequences of human, mouse, and rat from this dataset. The results shown in Table 2 indicate that gene prediction tools combined with using soft computing techniques perform comparably to popular gene predictors. To evaluate the nucleotide-level accuracy, 100 human sequences are taken from the HMR195 dataset. The performance is compared with that of five popular gene prediction programs (GENSCAN, HMMgene, GeneMark.hmm [91,92], AUGUSTUS, and FGENESH).

The exon-level accuracy is computed on 50 human sequences extracted from the same dataset. To compare these results, three gene prediction programs (GENSCAN, AUGUSTUS, and FGENESH) are used. Due to the unavailability of the soft computing-based gene prediction programs, the results are calculated from the data present in their respective articles. The programs tested on sequences other than human are excluded in this analysis. The nucleotide- and exon-level results are summarized in Tables 3 and 4, respectively. At the nucleotide level, programs other than soft computing methods perform better. However, homology-based soft computing methods are not included here. The exon-level results

Table 5
Soft computing-based gene prediction programs.

Program	Year(s)	Reference(s)	Organism(s)	Prediction type	Link (where available)
Grail I/Grail II	1991–1996	[39–45]	Human, mouse, *Drosophila*	Exon, single gene	http://compbio.ornl.gov/grail-1.3
GeneParser	1993–1995	[50,51]	Human	Exon	http://beagle.colorado.edu/~eesnyder/GeneParser.html
Codex	1995	[48]	Human, mouse, plant	Exon	–
GrailExp	1996–1997	[46,47]	Human	Exon, multiple genes	http://compbio.ornl.gov/grailexp
Modular neural network system	1996	[53]	Human	Exon	–
GIN	1998	[54]	Vertebrates	Exon, multiple genes	http://www.bork.emblheidelberg.de/fmilpetz/GIN
Exon–ENet	1999	[55]	Human, primates	Exon	–
MLF–ANN	2003	[56]	Yeast	Open reading frame	–
Evolutionary algorithm	2011	[59,60]	Human	Exon	–
Evolved ANN	2002	[63]	Human	Exon	–
RBFN combining	2007	[69]	Human, mouse, rat, *Escherichia coli*, *A. thaliana*	Exon	–

indicate that GrailExp performs better than all other methods. The results shown here can vary with the dataset used. A detailed comparative study of popular gene prediction methods is carried out in Ref. [85]. The evaluation is done on 570 vertebrate sequences. These results, when compared with GIN [54], signify that it performs similar to GENSCAN, one of the best gene prediction tools. The above analysis indicates that gene prediction tools based on soft computing techniques have good performance.

Discussion and conclusion

Gene prediction is an important open problem in bioinformatics. Since the early 1980s, a large number of gene prediction programs based on traditional hard computing techniques (e.g., HMM, DP) have been developed. Because of the difficult nature of this problem, soft computing techniques have been applied in this field due to their ability to handle uncertain and noisy data. This article has presented an extensive review of soft computing-based gene prediction methods. A list of gene prediction programs based on soft computing techniques is compiled in Table 5. For each program listed, the table gives the year in which it was reported, reference, organism used, prediction type, and important URLs available.

Although significant advances have taken place in this area, some problems still need to be addressed. The current splice site predictors based on soft computing techniques show improved performance over those available in the past. However, they are unable to solve the problem of noncanonical splice sites. Moreover, splice site prediction programs result in a large number of false positives, reducing their performance and making gene prediction extremely difficult. Various issues exist in the case of gene prediction, as mentioned before. The current gene prediction programs depend heavily on sequences from existing databases either for training or for homologue sequences [6]. The databases themselves are of poor quality. The dataset used to assess the performance of gene prediction methods does not model real sequences completely, which sometimes overestimates their performance.

Another problem in gene prediction is to locate short exons, especially those bordered by long introns. Short exons are easily missed due to a lack of discrimination characteristics present in these segments. To predict boundary exons with high accuracy is still problematic. One major issue in gene prediction is to predict alternative splice sites. Some programs have tried to predict suboptimal exons (e.g., GENSCAN), but a more effective mechanism is required to deal with this problem. In addition, the presence of noncoding RNA (ncRNA) genes in the human genome further complicates the problem due to their poor conservation. Moreover, the presence of overlapping genes makes the problem even more difficult.

Keeping in mind the above issues, gene prediction is a hard to solve problem. Some of these issues are addressed by soft computing techniques, but there is still a long way to go. 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 depends mainly on the type of information used as 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 have achieved a significant level of accuracy in identifying genes, the output results are still far from perfect because most of the methods are developed for specific genomes. In the future, techniques such as fuzzy logic, genetic algorithms, and neuro-fuzzy and neuro-genetic methods need to be explored. Neural networks alone are incapable of incorporating already known sequence information from biological knowledge. So they can be combined with traditional gene prediction techniques such as HMM to achieve better results. Nowadays, ncRNA gene prediction is becoming a promising area of research and, thus, can be further explored using soft computing techniques.

References

[1] Z. Wang, Y. Chen, Y. Li, A brief review of computational gene prediction methods, Genomics Proteomics Bioinform. 2 (2004) 216–221.
[2] J.H. Do, D. Choi, Computational approaches to gene prediction, J. Microbiol. 44 (2005) 137–144.
[3] R.D. Sleator, An overview of current status of eukaryote gene prediction strategies, Gene 461 (2010) 1–4.
[4] C. Mathe, M.F. Sagot, T. Schiex, P. Rouze, Current methods of gene prediction, their strengths, and weaknesses, Nucleic Acids Res. 30 (2002) 4103–4117.
[5] W. Li, A bibliography on computational gene finding, 2007, <http://www.nslij-genetics.org/gene>.
[6] J.M. Claverie, Computational methods for the identification of genes in vertebrate genomic sequences, Hum. Mol. Gene 6 (1997) 1735–1744.
[7] A. Krogh, Gene finding: putting the parts together, in: M. Bishop (Ed.), Guide to Human Genome Computing, Academic Press, San Diego, 1998, pp. 261–274.
[8] G.D. Stormo, Gene-finding approaches for eukaryotes, Genome Res. 10 (2000) 294–297.
[9] S. Bandyopadhyay, U. Maulik, D. Roy, Gene identification: classical and computational intelligence approaches, IEEE Transact. Syst. Man Cybern. C 38 (2008) 55–68.
[10] J.C. Setubal, J. Meidanis, Introduction to Computational Molecular Biology, PWS, Boston, 1996.
[11] S. Shenoy, B. Jayaram, N. Latha, P. Narang, T. Jain, K. Bhushan, S.A. Shaikh, S. Bose, P. Sharma, P. Singhal, A. Gandhimathi, P. Agarwal, V. Pandey, S. Dutta, G. Sandhu, A. Gupta, S. Shekhar, S. Tripathi, From gene to drug: a proof of concept for a plausible computational pathway, in: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, IEEE Computer Society, Jinan, China, 2006, pp. 1147–1152.
[12] K. Davies, Eukaryotic gene prediction, 2009, <http://biochem218.stanford.edu/Projects%202009/Davies%202009.pdf>.
[13] M.J. Schellenberg, D.B. Ritchie, A.M. Macmillan, Pre-mRNA splicing: a complex picture in high definition, Trends Biochem. Sci. 33 (2008) 243–246.
[14] J.M. Claverie, From bioinformatics to computational biology, Genome Res. 10 (2000) 1277–1279.
[15] M. Yandell, D. Ence, A beginner's guide to eukaryotic genome annotation, Nat. Rev. 13 (2012) 329–342.
[16] R.K. Jena, M.M. Aqel, P. Srivastava, P.K. Mahanti, Soft computing methodologies in bioinformatics, Eur. J. Sci. Res. 26 (2009) 189–203.
[17] A.B. Kurhe, S.S. Satonkar, P.B. Khanale, S. Ashok, Soft computing and its applications, BIOINFO Soft Comput. 1 (2011) 5–7.
[18] S. Rajasekaran, G.A.V. Pai, Neural Network, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications, Prentice Hall, Englewood Cliffs, NJ, 2005.
[19] S.S. Ray, S. Bandyopadhyay, P. Mitra, S.K. Pal, Bioinformatics in neurocomputing framework, IEEE Proc. Circuits Device Syst. 152 (2005) 556–564.
[20] C.H. Wu, Artificial neural networks for molecular sequence analysis, Comput. Chem. 21 (1997) 237–256.
[21] K. Nakata, M. Kanehisa, C. DeLisi, Prediction of splice junctions in mRNA sequences, Nucleic Acids Res. 13 (1985) 5327–5340.
[22] A. Lapedes, C. Barnes, C. Burks, R. Farber, S. Sirotkin, Applications of neural networks and other machine learning algorithms to DNA sequence analysis, in: G.I. Bell, T.G. Marr (Eds.), Proceedings of the Interface between Computation Science and Nucleic Acid Sequencing Workshop, Addison–Wesley, Santa Fe, NM, 1988, pp. 157–182.
[23] S. Brunak, J. Engelbrecht, S. Knudsen, Prediction of human mRNA donor and acceptor sites from the DNA sequence, J. Mol. Biol. 220 (1991) 49–65.
[24] S.M. Hebsgaard, P.G. Korning, N. Tolstrup, J. Engelbrecht, P. Rouze, S. Brunak, Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information, Nucleic Acids Res. 24 (1996) 3439–3452.
[25] N. Tolstrup, P. Rouze, S. Brunak, A branch point consensus from *Arabidopsis* found by non-circular analysis allows for better prediction of acceptor sites, Nucleic Acids Res. 25 (1997) 3159–3163.
[26] M.G. Reese, F.H. Eeckman, D. Kulp, D. Haussler, Improved splice site detection in genie, in: First Annual International Conference on Computational Molecular Biology (RECOMB), ACM Press, New York, 1997, pp. 232–240.
[27] D. Kulp, D. Haussler, M.G. Reese, F.H. Eeckman, A generalized hidden Markov model for the recognition of human genes in DNA, Proc. Int. Conf. Intell. Syst. Mol. Biol. 4 (1996) 134–142.
[28] H. Ogura, H. Agata, M. Xie, T. Odaka, H. Furutani, A study of learning splice site of DNA sequence by neural networks, Comp. Biol. Med. 27 (1997) 67–75.
[29] S. Rampone, Recognition of splice junctions on DNA sequences by BRAIN learning algorithm, Bioinformatics 14 (1998) 676–684.
[30] L.S. Ho, J.C. Rajapakse, Splice site detection with a higher-order Markov model implemented on a neural network, Genome Inform. 14 (2003) 64–72.
[31] J.C. Rajapakse, L.S. Ho, Markov encoding for detecting signals in genomic sequences, IEEE/ACM Transact. Comput. Biol. Bioinfom. 2 (2005) 131–141.

[32] T. Cai, Q. Peng, Predicting splice sites in DNA sequences using neural network based on complementary encoding method, Proc. Int. Conf. Neural Networks Brain 1 (2005) 473–476.
[33] L. Liu, Y.K. Hu, S. Yau, Prediction of primate splice site using inhomogeneous Markov chain and neural network, DNA Cell Biol. 26 (2007) 477–483.
[34] O. Johansen, T. Ryen, T. Eftesol, T. Kjosmoen, P. Rnoff, Splice site prediction using artificial neural networks, in: CIBB 2008, LNBI 5488, Springer-Verlag, Heidelberg, Germany, 2009, pp. 102–113.
[35] S. Makal, L. Ozyilmaz, Determination of splice junctions on DNA by neural networks, in: International Symposium on Innovations in Intelligent Systems and Applications, IEEE Turkey, Istanbul, TR, 2007, pp. 234–237.
[36] T. Naenna, R.A. Embrechts, A modified Kohonen network for DNA splice junction classification, IEEE Reg. 10 Conf. 2 (2004) 215–218.
[37] L. Ozyilmaz, Determination of exon and intron regions on DNA sequences by artificial neural networks, in: Advances in Molecular Medicine International, Journal of Molecular Biology, Biochemistry, and Gene Technology, Istanbul, Turkey, 2005, pp. 452–453.
[38] V.B. Bajic, S. Tang, H. Han, V. Brusic, Artificial neural networks based systems for recognition of genomic regions: a review, Informatica 26 (2002) 389–400.
[39] E.C. Uberbacher, R.J. Mural, Locating protein coding regions in human DNA sequences by a multiple sensor neural network approach, Proc. Natl. Acad. Sci. USA 88 (1991) 11261–11265.
[40] Y. Xu, R.J. Mural, J.R. Einstein, M.B. Shah, E.C. Uberbacher, GRAIL: a multi-agent neural network system for gene identification, Proc. IEEE 84 (1996) 1544–1552.
[41] E.C. Uberbacher, J.R. Einstein, X. Guan, R.J. Mural, Gene recognition and assembly in the GRAIL system: progress and challenges, in: H.A. Lim, J.W. Fickett, C.R. Cantor, R.J. Robbins (Eds.), Proceedings of the International Conference on Bioinformatics, Supercomputing, and Complex Genome Analysis, World Scientific, Singapore, 1993, pp. 465–476.
[42] Y. Xu, R. Mural, E. Uberbacher, Recognizing exons in genomic sequences using GRAIL II, Genet. Eng. 16 (1994) 241–253.
[43] Y. Xu, R.J. Mural, E. Uberbacher, Constructing gene models from accurately predicted exons: an application of dynamic programming, Comput. Appl. Biosci. 10 (1994) 613–623.
[44] Y. Xu, J.R. Einstein, R.J. Mural, M. Shah, E.C. Uberbacher, An improved system for exon recognition and gene modeling in human DNA sequences, in: R. Altman, D. Brutlag, R. Karp, R. Latrop and D. searls (Eds.), Proceedings of the International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1994, pp. 376–383.
[45] Y. Xu, G. Helt, J.R. Einstein, G. Rubin, E.C. Uberbacher, *Drosophila* GRAIL: an intelligent system for gene recognition in *Drosophila* DNA sequences, in: Proceedings of the First International Symposium on Intelligent Neural Biological Systems, IEEE Computer Society, 1995, pp. 128–135.
[46] Y. Xu, E. C. Uberbacher, Gene prediction by pattern recognition and homology search, in: Proceedings of the Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, 1996, pp. 241–251.
[47] Y. Xu, E.C. Uberbacher, Gene prediction by pattern recognition and homology search, in: Proceedings of the Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, 1996, pp. 241–251.
[48] L. Roberts, N. Steele, C. Reeves, G.J. King, Training neural networks to identify coding regions in genomic DNA, IEEE Conf. Artif. Neural Netw. 409 (1995) 399–403.
[49] R. Farber, A. Lapedes, K. Sirotkin, Determination of eukaryotic protein coding regions using neural networks and information theory, J. Mol. Biol. 226 (1992) 471–479.
[50] E.E. Synder, G.D. Stormo, Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks, Nucleic Acids Res. 21 (1993) 607–613.
[51] E.E. Synder, G.D. Stormo, Identification of protein-coding regions in genomic DNA, J. Mol. Biol. 248 (1995) 1–18.
[52] Y. Cai, C. Chen, Artificial neural network method for discriminating coding regions of eukaryotic genes, Comput. Appl. Biosci. 11 (1995) 497–501.
[53] A. Hatzigeorgious, N. Mache, M. Reczko, Functional site prediction on the DNA sequence by artificial neural networks, in: IEEE International Joint Symposia on Intelligence and Systems, IEEE Computer Society, 1996, pp. 12–17.
[54] Y. Cai, P. Bork, Homology-based gene prediction using neural nets, Anal. Biochem. 265 (1998) 269–274.
[55] L. Fu, An expert network for DNA sequence analysis, IEEE Intell. Syst. Appl. 14 (1999) 65–71.
[56] C. Li, P. He, J. Wang, Artificial neural network method for predicting protein-coding genes in the yeast genome, Internet Electron. J. Mol. Des. 2 (2003) 527–538.
[57] R.C. Chakraborty, Soft computing: introduction, 2010, <http://www.myreaders.info/html/soft_computing.html>.
[58] M. Mitchell, An Introduction to Genetic Algorithm, MIT Press, Cambridge, MA, 1998.
[59] J. Perez-Rodriguez, N. Garcia-Pedrajas, An evolutionary algorithm for gene structure prediction, IEA/AIE Part II LNAI 6704 (2011) 386–395.
[60] J. Perez-Rodriguez, N. Garcia-Pedrajas, Evolutionary computation, combined with support vector machines, for gene structure prediction, in: Proceedings of the Eleventh International Conference on Intelligent Systems, Design, and Applications, IEEE Computer Society, 2011, pp. 1359–1364.
[61] J.S.R. Jang, C.T. Sun, E. Mizulani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ, 1996.
[62] J. Bih, Paradigm shift: an introduction to fuzzy logic, IEEE Potentials 25 (2006) 6–21.
[63] G.B. Fogel, K. Chellapilla, D.W. Corne, Identification of coding regions in DNA sequences using evolved neural networks, in: G.B. Fogel, D.W. Corne (Eds.), Evolutionary Computation in Bioinformatics, Morgan Kaufmann, San Fransisco, 2002, pp. 195–218.
[64] C. Burge, S. Karlin, Prediction of complete gene structures in human genomic DNA, J. Mol. Biol. 268 (1997) 78–94.
[65] GENSCAN, <http://genes.mit.edu/GENSCAN.html>.
[66] A. Krogh, Two methods for improving performance of an HMM and their application for gene finding, in: Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, 1997, pp. 179–186.
[67] HMMgene, <http://www.cbs.dtu.dk/services/HMMgene>.
[68] A.L. Delcher, D. Harmon, S. Kasif, O. White, S.L. Salzberg, Improved microbial gene identification with GLIMMER, Nucleic Acids Res. 27 (1999) 4636–4641.
[69] Y. Zhou, Y. Liang, C. Hu, L. Wang, X. Shi, An artificial neural network method for combining gene prediction based on equitable weights, NeuroComputing 71 (2007) 538–543.
[70] E. Al-Dauod, Identifying DNA splice sites using pattern statistical properties and fuzzy neural networks, EXCLI J. 8 (2009) 195–202.
[71] F. Moghimi, M.T.M. Shalmani, A.K. Sedigh, M. Kia, Two new methods for DNA splice site prediction based on neuro fuzzy network and clustering, Neuro Comput. Appl. (2012) <http://dx.doi.org/10.1007/s0052-012-1257-y>.
[72] S. Makal, L. Ozyilmaz, S. Palavaroglu, Neural network based determination of splice junctions by ROC analysis, Proc. World Acad. Sci. Eng. Technol. 43 (2008) 613–615.
[73] V.V. Solovyev, A.A. Salamov, C.B. Lawrence, The prediction of human exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames, in: R. Altman, D. Brutlag, R. Karp, R. Latrop, D. Searls (Eds.), Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, 1994, pp. 354–362.
[74] V.V. Solovyev, A.A. Salamov, C.B. Lawrence, Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames, Nucleic Acids Res. 22 (1994) 5156–5163.
[75] I.B. Rogozin, L. Milanesi, Analysis of donor splice signals in different organisms, J. Mol. Evol. 45 (1997) 50–59.
[76] SpliceView, <http://zeus2.itb.cnr.it/~webgene/wwwspliceview_ex.html>.
[77] R. Guigo, S. Knudsen, N. Drake, T. Smith, Prediction of gene structure, J. Mol. Biol. 226 (1992) 141–157.
[78] Gene–Id3, <http://genome.crg.es/geneid.html>.
[79] T.A. Thanaraj, Positional characterisation of false positives from computational prediction of human splice sites, Nucleic Acids Res. 28 (2000) 744–754.
[80] M. Pertea, X. Lin, S.L. Salzberg, GeneSplicer: a new computational method for splice site prediction, Nucleic Acids Res. 29 (2001) 1185–1190.
[81] N. Mache, P. Levi, EST/STS guided identification of genes in human genomic DNA, in: ISMB 98 [poster], Montreal, Canada, AAAI Press, 1998.
[82] N. Mache, P. Levi, GENIO: a non-redundant eukaryotic gene database of annotated sites and sequences, in: RECOMB 98 [poster], New York, ACM Press, 1998.
[83] GENIO, <http://biogenio.com/splice/splice.cgi>.
[84] S. Rogic, HMR195 dataset, <http://srogic.wordpress.com/datasets/hmr195-dataset>.
[85] M. Burset, R. Guigo, Evaluation of gene structure prediction, Genomics 34 (1996) 353–367.
[86] M. Stanke, S. Waack, Gene prediction with a hidden Markov model and a new intron submodel, Bioinformatics 19 (2003) ii215–ii225.
[87] M. Stanke, R. Steinkamp, S. Waack, B. Morgenstern, AUGUSTUS: a webserver for gene finding in eukaryotes, Nucleic Acids Res. 32 (2004) w309–w312.
[88] AUGUSTUS, <http://bioinf.uni-greifswald.de/webaugustus/prediction/create>.
[89] A.A. Salamov, V.V. Solovyev, Ab initio gene finding in *Drosophila* genomic DNA, Genome Res. 10 (2000) 391–393.
[90] FGENESH, <http://linux1.softberry.com/berry.phtml?topic=fgenesh&group=programs&subgroup=gfind>.
[91] A.V. Lukashin, M. Borodovsky, GeneMark.hmm: new solutions for gene-finding, Nucleic Acids Res. 26 (1998) 1107–1115.
[92] GeneMark.hmm, <http://topaz.gatech.edu/GeneMark/eukhmm.cgi>.