2009 Ninth International Conference on Intelligent Systems Design and Applications

Optimization of Multi-classifiers for Computational Biology: Application to the Gene Finding problem

Rocío Romero-Zaliz
DECSAI, UGR
Granada, Spain
rocio@decsai.ugr.es

Coral del Val
DECSAI, UGR
Granada, Spain
delval@decsai.ugr.es

Igor Zwir
DECSAI, UGR
Granada, Spain
Howard Hughes Medical Institute
St. Louis, Missouri, USA
igor@decsai.ugr.es

Abstract

Genomes of many organisms have been sequenced over the last few years. However, transforming such raw sequence data into knowledge remains a hard task. A great number of prediction programs have been developed to address part of this problem: the location of genes along a genome. We propose a multiobjective methodology to combine algorithms into an aggregation scheme in order to obtain optimal methods' aggregations. Results show a major improvement in specificity and sensitivity when our methodology is compared to the performance of individual methods for gene finding problems. The here proposed methodology is an automatic method generator, and a step forward to exploit all already existing methods, by providing optimal methods' aggregations to answer concrete queries for a certain biological problem with a maximized accuracy of the prediction. As more approaches are integrated for each of the presented problems, de novo accuracy can be expected to improve further.

1. Introduction

Genomes of many organisms have been sequenced over the last few years. However, transforming such raw sequence data into knowledge remains a hard task. A great number of prediction programs have been developed to address one part of this problem: the location of genes along a genome [2, 4, 1, 9]. Unfortunately, finding genes in a genomic sequence is far from being a trivial problem. Computational gene prediction methods have yet to achieve perfect accuracy, even in the relatively simple prokaryotic genomes [11]. Gene prediction is one of the most important problems in computational biology due to the inherent value of the set of protein-coding genes for other analysis.

Despite the advances in the gene finding problem, existing approaches to predicting genes have intrinsic advantages and limitations [11]. Furthermore, there is no program that can provide perfect predictions for any given input. The gene-finding problem can be interpreted as a simple decision between which section of a sequence is protein-coding and which not. Many different programs are available which give distinct solutions. Our methodology combines these approaches into an aggregation scheme to provide better predictions by taking advantage of the different methodologies' starknesses and avoiding their weaknesses. Moreover, we use a multiobjective approach to extract the best aggregation of methods by maximizing the specificity and sensitivity of their predictions.

We applied our methodology to a reference dataset in gene prediction containing 570 multi-species DNA sequences of known genes [5].

2. Materials and Methods

The aggregation of methods is accomplished by using the union -$\cup$- and intersection -$\cap$- operator [8]. All potential aggregations conform a space of potential hypotheses, which can be represented as a lattice structure (Figure 1). We search for the best aggregation of methods, moving from hypothesis to hypothesis towards the most general (i.e., the union of all methods) and the most specific (i.e., the intersection of all methods) which are located at the top and the bottom of the lattice, respectively [12] (Figure). In the gene finding problem we explore nine methods, n = 9, termed M1 to M9, conforming a total set of 512 potential aggregations.

We selected the dataset from Guigó et al. [5] which is a reference for assessing the quality of gene prediction programs. This set contains only sequences representing only one complete spliceable functional product of a gene in the

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.70

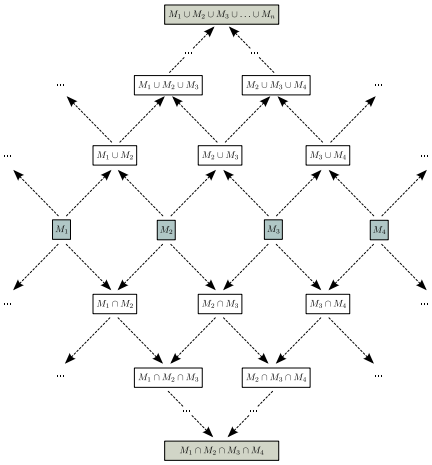
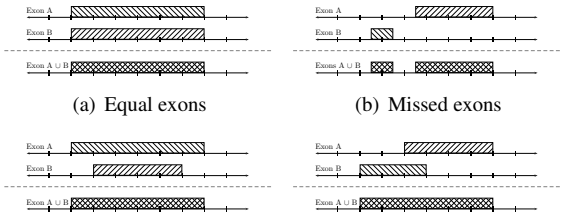


Figure 1. Lattice of potential hypothesis, methods' aggregations of $M_1, \cdots M_n$ using the -∪- and -∩- operators. The solid arrows show the direction of the search in the space of hypothesis.

forward strand. It contains as well representatives from all vertebrate genomes in 570 sequences totaling 2,892,149 bp. There are 2649 coding exons, corresponding to 444,498 coding bp, which gives a coding density of about 15%. The programs used in this study were designed to predict gene structure, or at least a set of spliceable exons, in vertebrate or pre-human genome sequences: GeneID [7], GeneID+, SORFIND [10], GeneParser2, GeneParser3 [14], GRAIL 2 [18], GenLang [3], FGENEH [15] and Xpound [17]. Genscan is a Generalized hidden markov model (GHMM) and one of the most successful gene prediction programs. It was the most accurate individual method in the reference dataset but, since most of the genes in the dataset were used to train it, it was not included in the aggregation assessment. All of them are *de novo* gene predictors using a single genome sequence. GeneID combines different algorithms using Position Weight Arrays to detect features such as splice sites, start and stop codons and Markov Models to score exons and Dynamic Programming (DP) to assemble the gene structure [7]. GeneParser2 employs a DP algorithm and a simple feed-forward Neural Network (NN) to maximize the number of correct predictions [14]. GeneID+ and GeneParser3 extend their respective original versions by using the potential similarity between the query sequence and the known amino acid sequences as evidence in gene identification. GenLang uses grammar rules and a parser for eukaryotic protein encoding genes [3]. GRAIL uses a set of NNs to evaluate candidate exons and a DP algorithm to build the best possible single gene model [18]. Sorfind and Xpound use both a variety of statistical models, primary Markov Chain models to score exons [10, 17]. FgeneH uses linear discriminant analysis to best discern between two functional classes of sequences, combined with a DP algorithm to predict optimal gene models from the list of potential exons [15].

The aggregation of the different methods in the Gene Finding problem is performed at a nucleotide level. This aggregation joins two overlapping or adjacent exons into a larger new exon (Figure 2).



(a) Equal exons (b) Missed exons

(c) Included exons (d) Overlapped exons

Figure 2. Example of exons aggregation by the union operator.

We measured the accuracy of a prediction on a test sequence by comparing the predicted coding value (coding or non-coding) with the true coding value for each nucleotide along the test sequence. This has been one of the most widely used approaches in evaluating the accuracy of coding region identification and gene structure prediction methods. Nucleotide level accuracy is calculated as a comparison of the annotated nucleotides with the predicted nucleotides. Sensitivity (Sn) (Equation 1) is the proportion of annotated nucleotides (as being coding or part of an mRNA molecule) that is correctly predicted, and specificity (Sp) (Equation 2) the proportion of predicted nucleotides (as being coding or part of an mRNA molecule) that is so annotated. As a summary measure, we have computed the correlation coefficient (CC) (Equation 3) between the annotated and the predicted nucleotides [5].

$$Sn = \frac{TP}{TP + FN} \quad (1)$$

$$Sp = \frac{TP}{TP + FP} \quad (2)$$

$$CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN_F P) \times (TP + FP) \times (TN + FN)}} \quad (3)$$

3. Results

The number of genes correctly predicted was calculated according to the average CC for the individual (Table 1) and the aggregation of methods (Table 2). We express the accuracy of the methods' aggregation by considering a gene correctly retrieved when its CC > 0.7. When using different thresholds (e.g., CC > 0.5) (data not shown) the ranking of individual methods or methods' aggregations was not affected. Out of all gene prediction programs analyzed, GParser3 achieved the highest number of correctly

predicted genes. However, its average CC for all genes was not the highest differing in more than 0.05 from the highest one (values ranking in the [0-1] interval with the best value in 1). GeneID failed to predict 130 genes, almost 23% of the dataset though it achieved the best performance attending to the average CC. Its sensitivity value was close to the average (0.694) but its specificity value was much lower than the rest (Table 1). FgeneH obtained the best specificity, sensitivity and CC, but predicted correctly two genes less than GParser3. These results show that a high average CC does not imply a good performance, and viceversa, since the average might hide some low CCs for specific genes. Some programs predict specific genes with high CCs close to 1, but the same program is not able to predict correctly other genes (CCs below 0.7 or even 0.5) (Table 2).

Method	Sp	Sn	Average Correlation Coefficient	#Genes correctly predicted
GParser3	0.759	0.724	0.714	*413*
FgeneH	*0.847*	*0.772*	*0.768*	411
GeneID+	*0.713*	0.718	0.694	393
GRAIL	0.837	0.724	0.731	389
Sorfind	0.834	0.697	0.705	349
genlang	0.747	0.719	0.673	323
XPound	0.825	0.611	0.652	302
GParser2	0.778	0.652	*0.642*	299
GeneID	0.806	*0.632*	0.649	*283*

Table 1. Results obtained by each of the nine individual methods. Gene finding methods are ordered by descending number of genes correctly retrieved, where a gene is considered correctly retrieved when its correlation coefficient is over 0.7**. The best result for each column is highlighted in italic and color-coded in blue, while the worst result is highlighted in italic and color-coded in grey.**

All 502 potential methods' aggregations of the nine used gene finding programs were performed and evaluated. The top 10 methods' aggregations are shown in Table 2, where aggregations including GeneID+, FgeneH, GParser3 and GRAIL generally improve individual methods performance in terms of accuracy. The best aggregation is the union of GeneID+ ∪ GRAIL ∪ FgeneH, which correctly predicts 90% of the dataset. FgeneH is present in the best ten methods' aggregations, providing supplementary predictions to the other methods. For instance, when FgeneH is combined with GParser3, the number of genes that are correctly predicted increases from 413 to 494. This change is due to the fact that FgeneH detects several genes, especially those with length over 8000 nt, that GParser3 was unable to identify while increasing the prediction of others. This behavior is consistent with the other aggregations with FgeneH. XPound appears in several sets of best aggregations even though is one of the worse individual methods (Table 1), this is because XPound predicts correctly three genes from the entire data set (HRSPRMI, HUMHMGIY, S59780) that are not correctly predicted by any other method.

The levels of specificity and sensitivity obtained by all methods' aggregations using the union operator to predict each of the 570 genes are shown in Figure 3(a) and Figure 3(b). The range of colors between green and red in both images is not uniformly distributed, but centered in a threshold of 0.8 to easily distinguish good values from bad specificity and sensitivity levels. The specificity of each methods' aggregation to retrieve each gene expression profile is shown in Figure 3(a). The top rows of the plot correspond to methods' aggregations with fewer numbers of methods, while bottom rows correspond to methods' aggregations composed by a large number of methods. From the top rows we can infer that some genes are not predicted with very high specificity levels when we use only one or two methods' aggregations. We see that there are several genes, those in the first columns (e.g., HUMDZA2G, BOVGAS, HS2OXOC), for which almost any methods' aggregation obtains good specificity values, while there are other genes showing black/red areas meaning that in their prediction a large set of methods' aggregations obtains good specificity values (e.g., HUMRPS6B, S66606, HUMMKXX). We can also see a set of genes in the center of the figure with similar behavior for all methods' aggregations (e.g., RATIGFBA, MUS100B, RNGMTG). Another noticeable feature of the graphic is that some methods' aggregations share a similar behavior when predicting the genes, as it can be appreciated in large horizontal areas. The sensitivity of each methods' aggregation to predict each gene is shown in Figure 3(b). The top rows of the plot correspond to methods' aggregations with fewer numbers of methods, while bottom rows correspond to methods' aggregations composed by a large number of methods. We see that most of the columns have sensitivity levels near to 1 for the majority of the methods' aggregations examined. There are some other columns with green areas as well as black/red areas, meaning that such genes are predicted with successful levels of sensitivity by only some of the methods' aggregations. The lower rows (methods' aggregations composed by a large number of methods) correspond to methods' aggregations with lower sensitivity values as there is a higher proportion of green areas over black/red ones.

The comparison between the prediction accuracy of the individual methods and the aggregation strategy can be seen in Figure 4(a). This figure shows that the aggregation of methods increases the sensitivity of the predictions, without lost of specificity. Figure 4(b) depicts the percentage of genes correctly retrieved by all aggregations. It can be con-

Method	Sp	Sn	Average Correlation Coefficient	#Genes correctly predicted
GeneID+ ∪ GRAIL ∪ FgeneH	0.810	0.967	0.853	*504*
GeneID+ ∪ FgeneH	*0.862*	0.925	*0.863*	493
GParser3 ∪ FgeneH ∪ XPound	0.818	0.949	0.845	491
GeneID+ ∪ GParser3 ∪ GRAIL ∪ FgeneH	0.783	*0.975*	0.837	488
GeneID+ ∪ GParser3 ∪ FgeneH	0.831	0.947	0.854	486
GeneID+ ∪ FgeneH ∪ XPound	0.818	0.958	0.851	485
GeneID+ ∪ GParser3 ∪ FgeneH ∪ XPound	0.792	0.969	0.837	479
GeneID ∪ GeneID+ ∪ GRAIL ∪ FgeneH	0.768	0.975	0.827	478
GeneID+ ∪ Sorfind ∪ GParser3 ∪ GRAIL ∪ FgeneH	0.754	0.981	0.817	465
GeneID+ ∪ GParser3 ∪ GRAIL ∪ FgeneH ∪ Xpound	0.754	0.980	0.817	464

Table 2. Ten best aggregation of methods. Gene finding methods are ordered by descending number of genes correctly retrieved, where a gene is considered correctly retrieved when its correlation coefficient is over 0.7. The best result for each column is highlighted in italic and color-coded in blue.

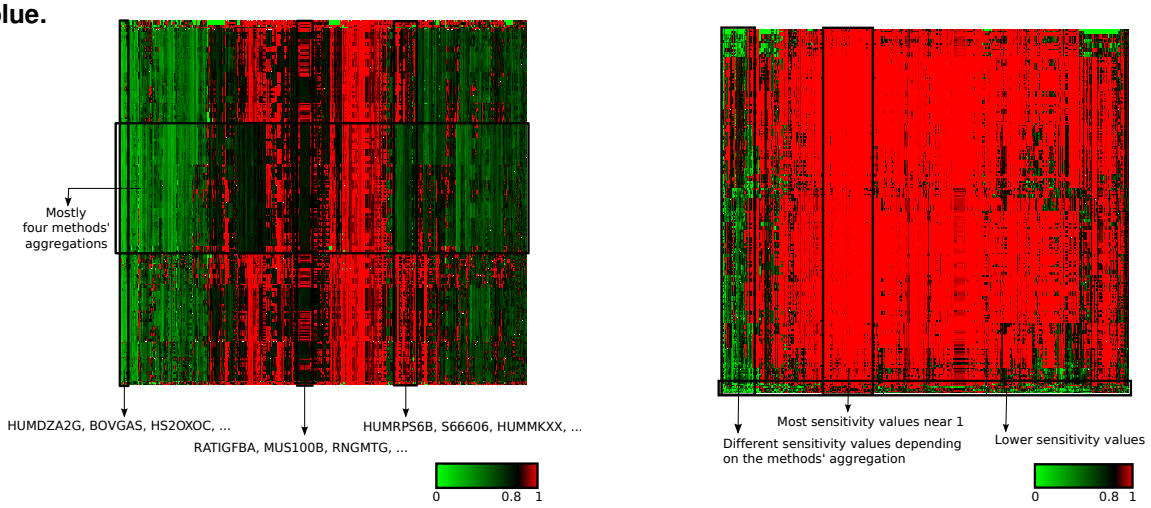
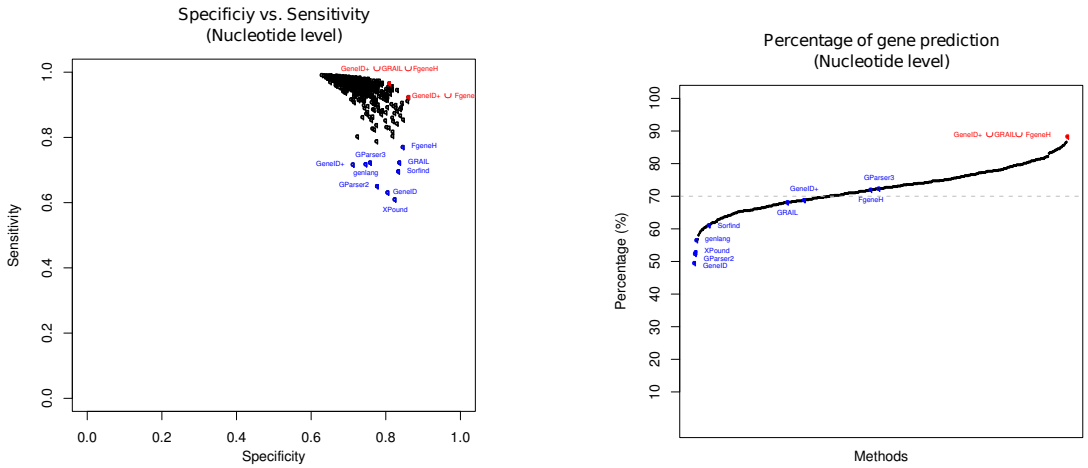


Figure 3. Graphical representation of the specificity (a) and sensitivity (b) obtained by the methods' aggregations using the ∪ operator to predict genes. Only methods' aggregations predicting the same or more genes than the best individual method (GeneParser3) are shown. Each column represents a gene (570 columns) and each row a methods' aggregation including individual methods. Color is coding values from 0 -green- to 1 -red-. Therefore, green points represent low specificity while red points correspond to those methods' aggregations achieving a high specificity level. Black points correspond to specificity levels around 0.8.

cluded that almost half of all methods' aggregations have a better performance than the individual ones. Moreover, the best methods' aggregation predictor, GeneID+ ∪ GRAIL ∪ FgeneH (see Table 2), increases the percentage of prediction by a 15% approximately when compared with the best individual predictor, Gparser3.

In the case of the methods' aggregations obtained using the intersection operator, we can see a different behavior. Results are included in Figure 5. The intersection operator obtains good results when the approaches used in the intersection predict the same exons. Therefore, the greater the number of methods intersected, the lowest the number of genes correctly retrieved. FgeneH is present in many of the best aggregations, but it never appears near the GeneID or GeneID+ methods, because their results are very different from the former. We can conclude that most of the individual methods have a better performance than any aggregation of methods using the intersection operator for the gene finding problem, but the intersection operator could be use to find consensus exons predicted by all programs (usually they coincide with canonical exons) and use this information for added quality value.



(a) (b)

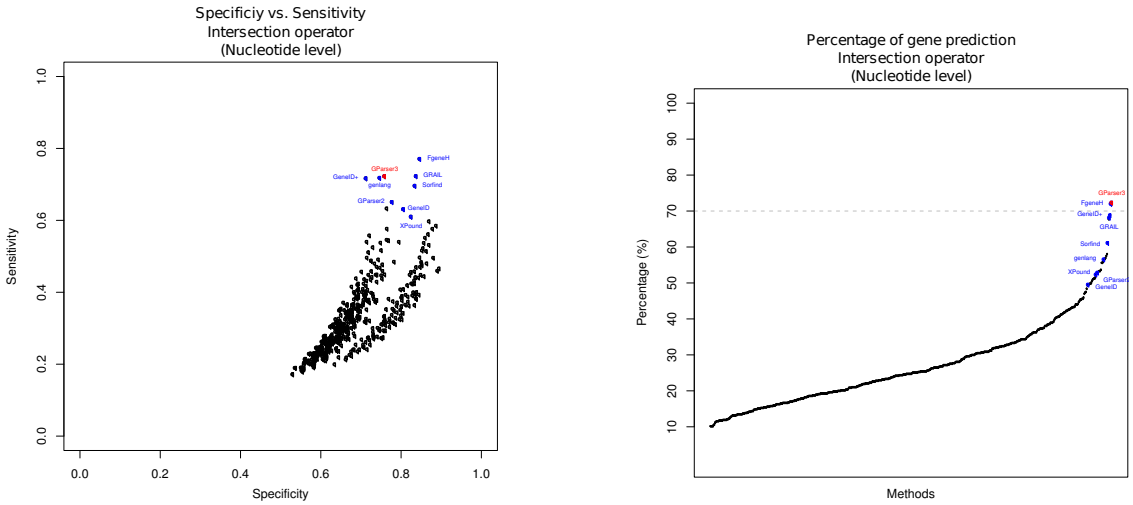
Figure 4. Gene finding prediction accuracy using the ∪ operator: (a) Results of the individual performance of methods and of each methods' aggregation. (b) Percentage of correctly predicted genes for each individual method and methods' aggregation. Individual methods are color-coded in blue, aggregations of methods in black and a few Pareto optimal solutions are shown in red. A gene is considered correctly retrieved when its CC is over 0.7**.**

4. Discussion

We propose a methodology to combine programs into a aggregation scheme. This idea provides better predictions by combining the advantages of the different methodologies used in each program. We introduced the use of a multiobjective approach to extract the best aggregation of methods by maximizing the specificity and sensitivity of their predictions. This way we avoid redundant and overlapping predictions that might be produced depending on the methodologies and the aggregation scheme used. The application of the proposed methodology to the gene finding problem to obtain optimal methods' aggregations showed a major improvement in sensitivity when compared to the performance of individual methods for each topic. The specificity levels obtained by the aggregation of gene finding methods improved or decreased depending on the methods used in the aggregation. When determining which aggregation of methods was the best one for the gene prediction problem, sensitivity and specificity were in contradiction. Nevertheless, the calculation of the correlation coefficient helped in the selection of the best methods' aggregation. The best aggregations include methods employing different algorithmic strategies that predict correctly different subset of the genes in the dataset. Although the statistical properties of coding regions allow for a good discrimination between large coding and non-coding regions, the exact identification of the limits of exons or of gene boundaries remains difficult. For instance, FGENEH and GeneID have strong constraints concerning this point. In the first case, predicted coding region limits are often incorrect, for example, short exons smaller than 40 bp tend to be difficult to locate, as discriminative statistical characteristics are less likely to appear in short strands. In the second case, the alternative splicing, a predicted structure frequently splits a single true gene into several or, alternatively, merges several genes into one. Such problems are, however, very complex, as intergenic and intronic sequences do not differ much, and specific gene boundary signals in the UTRs (e.g. the TATA box and the polyadenylation signal), are often too variable and sometimes are not even present. However, FGENEH performs very well for especially low %G+C sequences [6]. In contrast to other studies, we show that even programs with a low individual performance, such as XPound, can contribute to the accuracy improvement of a certain aggregation, in this case because it is able to identify genes with either very short or very long introns. The decrease of gene density in genome sequences and the presence of larger introns has been recently been reported to drop accuracy significantly [6].

There are several previous works combining gene finding programs [13, 16], but they fail to obtain good results as they use simultaneously all programs instead of optimize their aggregation. *De novo* gene prediction for compact eukaryotic genomes is already quite accurate, although mammalian gene prediction lags way behind in accuracy. One future scope would be the application of this approach to identify ways to quickly combine many or all-existing programs trained for the same organism, and determine the upper limit of predictive power by aggregations of programs genome wide [6].

In the last ten years, the existing competitive spirit has increased the number of programs/algorithms created, updated and adapted for the two biological problems here pre-



(a) (b)

Figure 5. Gene finding prediction accuracy using the ∩ operator: (a) Results of the individual performance of methods and of each methods' aggregation. (b) Percentage of correctly predicted genes for each individual method and methods' aggregation. Individual methods are color-coded in blue, aggregations of methods in black and a few Pareto optimal solutions are shown in red. A gene is considered correctly retrieved when its CC is over 0.7**.**

sented [11, 2, 9]. On one side the development of a new algorithm always implies the sacrifice of an objective in favor of another, which makes very difficult for novel approaches to improve in absolute terms the quality of the existing ones. On the other side, the impressive amount of alternative algorithms available for different biological problems is confusing the users, who wonder what makes the programs different, which one should be used in which situation and which level of prediction confidence to expect. Finally, users also wonder whether current programs can answer all their questions. The answer is most probably no, and will remain to be negative as it is unrealistic to imagine that such complex biological processes can be explained merely by looking at one objective. The here proposed methodology is an automatic method generator, and a step forward to exploit all already existing methods, by providing optimal methods' aggregations to answer concrete queries for a certain biological problem with a maximized accuracy of the prediction.

References

[1] C. Burge and S. Karlin. Finding the genes in genomic dna. *Struct. Biol.*, 8:346–354, 1998.

[2] J. M. Claverie. Computational methods for the identification of genes in vertebrate genomic sequences. *Hum. Mol. Genet.*, 6:1735–1744, 1997.

[3] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.

[4] R. Guigó. Computational gene identification: an open problem. *Comput. Chem.*, 21:215–222., 1997.

[5] R. Guigó and M. Burset. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, 1996.

[6] R. Guigó et al. Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447, 2007.

[7] R. Guigó, S. Knudsen, et al. Prediction of gene structure. *J. Mol. Biol.*, 226:141–157, 1992.

[8] P. Halmos. *Naive Set Theory*. D. Van Nostrand Company., Princeton, NJ, 1960.

[9] D. Haussler. Computational genefinding. *Trends Biochem. Sci.*, pages 12–15., 1998.

[10] G. B. Hutchinson and M. R. Hayden. The prediction of exons through an analysis of spliceable open reading frames. *Nucleic Acids Res.*, 20:3453–3462, 1992.

[11] C. Mathé, M. F. Sagot, et al. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, 30(19):4103–4117, 2002.

[12] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[13] K. Murakami and T. Takagi. Gene recognition by combination of several gene-finding programs. *Bioinformatics*, 14:665–675, 1998.

[14] E. E. Snyder and G. D. Stormo. Identification of protein coding regions in genomic dna. *J. Mol. Biol.*, 248:1–18, 1995.

[15] V. V. Solovyev, A. A. Salamov, et al. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Res.*, 22:5156–5163, 1994.

[16] M. Tech and R. Merkl. Yacop: Enhanced gene prediction obtained by a combination of existing methods. *Silico Biology*, 3(4):441–451, 2003.

[17] A. Thomas and M. H. Skolnick. A probabilistic model for detecting coding regions in dna sequences. *IMA J. Math. Appl. Med. Biol.*, 11:149–160, 1994.

[18] Y. Xu, R. J. Mural, et al. Constructing gene models from accurately predicted exons: An application of dynamic programming. *Comput. Appl. Biosci.*, 10:613–623, 1994.