A Survey on Gene Prediction using Neural Network

S. Lakshmi[#1], A. Shahin[*2]

[#]Computer Science Department, Auxilium College, Thiruvalluvar University
Vellore, Tamil Nadu

Abstract**—** Neural network is traditionally used to refer to a network or circuit of Gene in the research traditionally. Despite a high number of techniques specifically dedicated to networks problems as well as many successful applications, we are in the initiation process to massively integrate the aspects and experiences in the different core subjects such as medicine, computer science, engineering and mathematics. Currently, a large number of gene identification tools are based on computational intelligence approaches. Here, we have revealed the existing conventional as well as computational methods to classify genes and various gene predictors are compared. My paper includes some drawbacks of the presently available methods and also, the feasible instructions for future directions are discussed.

Keywords**—** DNA, Gene, Artificial Neural Networks, SVM, etc.

I. INTRODUCTION

In the field of Neural Networks [1], gene identification from large DNA sequence is known to be a significantly tragedy. The exact number of genes encoded by the human genome is still unknown [2, 3]. Hence, genome annotation is a necessity and a multi-step process in itself. The steps involved in genome annotation can be grouped into three categories: nucleotide level (gene prediction or identification), protein-level (structure determination of proteins), and process-level annotation (mechanism of biochemical reactions). Among these three categories, nucleotide-level annotation is the most significant, because it primarily deals with gene annotation, a fundamental step in molecular biology [4]. Therefore, partitioning them into promoters, genes, intergenic region, regulatory elements, etc. for interpreting long unidentified genomic sequence are required to be modified from the conventional techniques became essential [5]. Consequently, the mathematical approach in the segment of molecular biology and genomics is gaining a lot of attention and is an interesting research area for many scientists [3, 6, 7]. The methods for gene-finding which are being used now a day are more precise and reliable as well than the earlier tactics. The advances in gene finding through dynamic programming, decision trees and Hidden Markov Model (HMM), are also studied [7]. The available gene prediction programs and methods are also reported and summarized [8-10]. The existing methods for gene prediction are also studied and compared [10, 11]. A comprehensive review of prediction methods for functional sites, protein coding genes, tRNA etc. is also reported [12]. A summary of a few techniques based on computational gene identification tools is also reported [13]. Catherine provided a review of the existing approaches of gene identification in eukaryotic organisms, their advantages as well as the limitations [14-16]. The gene identifier Combiner integrates multiple gene prediction programs and a large number of evidences are available in a typical annotation pipeline including substance from proteins, ESTs, cDNAs and splice site predictions [18]. Other approaches consisting of multiple evidence types can be found in the Eu-Gene [19] and GAZE [20] systems

In this paper, various conventional approaches of gene identification, via Bayesian Networks are explained along with the review of some of the computational intelligence techniques. In most of the previous reviews on this topic, the drawbacks of the classical methods are not described. We have highlighted the recent developments in gene identification tools, especially those based on computational intelligence techniques like Neural Networks and Decisions tree.

II. OVERVIEW OF THE GENERAL GENE PREDICTION TECHNIQUES

The general techniques for gene prediction can be differentiate into identifying the evidence for gene and integrating the various evidences of genes for predicting the gene structure as shown in Fig. (1) [14].

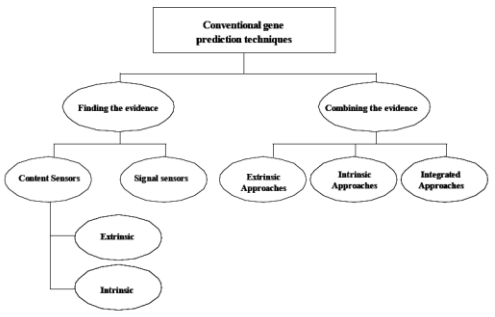

Figure 1: Gene Prediction using Conventional Techniques

A. Evidence Discovery

Here, we are considering the problem of obtained the genes coding for a protein sequence with eukaryotes only. The problem to finding genes in prokaryotes presents several types of difficulties (there are no introns and the intergenic regions are small, but genes will often overlap each other and the translation starts are difficult to predict correctly). Dynamically, a eukaryotic gene may be defined as being setup to a transcribed region and of regions that is allocate the gene expression, that as the promoter region which controls both the site and the expansion of transcription and is mostly found in the 5' part of the gene. The currently existent gene prediction software looks only for transcribed region of genes, which is called `the gene'. Signal sensors and content sensors are two fundamental types of information those are presently locate genes in genomic sequence

III. An Overview of Gene Techniques

In this section, brief overviews of some endorsed gene identification techniques are discussed without going deeply into their mathematical parts and algorithm.

A. Bayesian Networks

A Bayesian network [20] is a graphical model that encodes probabilistic relationships among variables of interest. When we used in conjunction with the statistical techniques, the graphical model have different advantages for data analysis.

1) Since, the model encodes dependencies upon all variables, it easily handles situations where some data entries are missing.

2) A Bayesian network may be used to learn causal relationships, and hence, it can be used to gain understanding about a problem domain and to predict the consequences of intervention.

3) Since the model has both causal and probabilistic semantics, it is an ideal representation for combining prior knowledge (which often comes in causal form) and data.

4) A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - a conditional distribution for each and every node given its parents:

P $(X_i \mid \text{Parents}(X_i))$

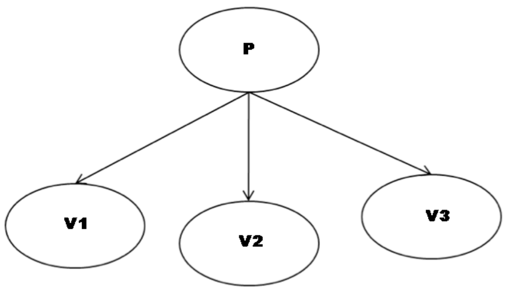

Figure 2: Bayesian Network

A Bayesian network for a set of variables V = {V1, . . . , Vn} consists of:

1) A network structure S that encodes a set of conditional independence assertions about variables in V and 2) A set P of local probability distributions associated with each variable. Put together, those components define the joint probability distribution for V. The network structure S is directed acyclic graph. The nodes which S are in one-to-one correspondence with the variables V. Vi argue both the variable and its corresponding node, and P argue the parents of node Vi in S as well as the variables corresponding to these parents. The lack of possible arcs in S encodes conditional independencies. In specific, it given structure S, the joint probability distribution for V is given

By p(V) = Πni =1p(Vi |Pai).

The local probability distributions P are the distributions corresponding to the rule the product of the previous equation. Accordingly, the pair (S;P) encodes the joint distribution p(V). Methods of learning probabilities in a Bayesian network, and techniques for learning by incomplete data, are studied in detail in [21]. A Bayesian network framework for combining gene predictions from multiple systems is given in [22], where the approach adopted is that of combining the advice of multiple experts.

B. Artificial Neural Networks

Artificial neural networks is computer algorithms based loosely on modelling the neuronal structure of natural organisms [23].

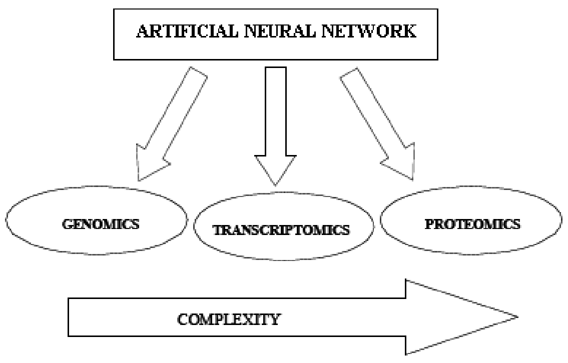

Figure 3: Major branch of Artificial Neural Network.

They are stimulus-response transfer functions that accept some input and yield some output. They are generally used to learn an input–output mapping over a set of examples. In general, if they given sufficient complexity, there exists an ANN that can map every input pattern to its appropriate output pattern, as long as the input output mapping is not one-to-many. ANNs are, therefore, well suited for use as detectors and classifiers. Multi layer perceptrons, also sometimes described as feed forward networks, are the most common architecture used in supervised learning applications .Each computational node sums N weighted inputs, subtracts a threshold value, and passes the result through a logistic (sigmoid) function. Single perceptrons form decision regions separated by a hyper plane. If the different data classes being input are linearly separable, a hyper plane can be positioned between the classes by adjusting the weights and bias terms. If the input data are not linearly separable, a least mean square (LMS) solution is typically generated to minimize the mean square error (MSE) between the calculated output of the network and the actual desired output. An earlier attempt at computer-aided gene recognition, such as the well-known GRAIL software, used an ANN to combine a number of coding indicators calculated within a fixed sequence window [24]. Fickett and Tung [25] noted that at the core of most gene recognition algorithm are one or more coding measures: functions that calculate, for several window of the sequence, a number or vector intended to measure the “codingness” of the sequence. Common examples of those measures include codon usage, base composition vector, etc. An exon-recognition method includes both a coding measure and a method of deciding between “coding” or “non coding” regions for each vector. Such an approach to evolve ANNs capable of identifying coding and non coding regions is available in [26]. The classification process using evolved ANNs proceeded as follows. A classification of DNA was interrogated using a window of 99 nucleotides. The ANN was used to classify the nucleotide in the center of the window as either coding or non coding. For this analysis, the neural network architecture was fixed and consisted of nine input nodes (corresponding to nine features), 14 hidden nodes, and one output node. The output decision was normalized from −1 (non coding) to +1 (coding) for each position in the sequence. If the output value was less than −0.5 (or +0.5), it was classified as coding (or, non coding). In evolved ANN, genetic algorithms (GAs) have been used for determining the appropriate network architecture. “Offspring” ANN architectures are created from the parent networks through random alteration. The number of layers, nodes, and the values for the associated parameters (e.g., weights and biases of a MLP, weights, biases, means, and standard deviations of a radial basis function network) are encoded in the chromosomes, and their appropriate values are evolved using GAs. The architecture is fixed to nine input nodes, 14 hidden nodes, and one output node, while the interconnections weights and the biases are evolved using GAs. The coding indicators of the system, used has the set of input features, are Frame bias matrix, Fickett feature, coding sextuple word preferences, word preferences are frame 1, word preferences in frame 2, word preferences in frame 3, maximum word preferences in frames, sextuple word commonality, and repetitive sextuple word [26]. A flowchart of the entire gene identification procedure is given in Fig. 5. In the post processing step, spikes in the output vector is first removed by replacing the value of a central nucleotide to the minimum over its own value and those of its left and right neighbors (theMin3 processing step). In the exon-identification step, a set of continuous coding nucleotides was predicted as a putative exon, and its start to end positions were noted. Across each such exon, a window of 50 nucleotides was considered, and statistical measures for intron filtering (based on the frequency of occurrence of nucleotides at the intron/exon boundaries) was used to adjust the location of the putative exon to a more appropriate location within the window. Finally, the domain knowledge that a majority of exons in human DNA was more than 15 nucleotides long was used to reject all predicted exons of length less than 15. ANNs combined with a rule-based system has been used for splice site prediction in human Arabidopsis thaliana by using a joint prediction scheme where the prediction of transition regions between introns and exons regulates a cut off level for local splice site assignment [27]. This is followed by a rule-based refinement that uses splice site assurance values, prediction scores, coding context, and distances between potential splice sites. This has been further improved by the incorporation of information regarding the branch point consensus sequence found by a noncircular approach using HMM [28]. The application of a time-delay-architecture-based feed forward neural network for analysis of the Drosophilia melano gaster genome has been presented in [29]. It was tested on the Adh region of 2.9 Mbases of the Drosophila genome, where it was found to provide a recognition rate of 75% with a false positive rate of 1/547 bases. Recently, a neural-network based multi classifier system has been proposed for gene identification in E. coli by locating the promoters of the genes [30]. A set of 324 known E.coli promoters and 429 known non promoters was coded using four different coding techniques, and four different neural classifiers were trained on each set. The final classification was an aggregate of the individual classifications obtained using a variant of the logarithmic opinion pool method. Some other applications of neural networks for gene finding are also discussed in [31]. An attempt to employ ANNs to predict the regions that overlap with the first exon of a gene or lies in its close proximity was reported in [32] and [33]. Dragon gene start finder (DGSF) [32], [33] combines three systems to achieve this goal. First, it uses a promoter finder system to estimate the transcription start site (TSS). Another system estimates the presence of CpG islands (which are stretches of DNA containing a significantly high frequency of CG sequence, often located around the promoters of genes that perform general cell functions) on the DNA strand. Several signals are extracted from its predicted TSS and CpG islands. The data are then normalized and transformed using principal component analysis. Thereafter, a four-layer neural network is used to make predictions whether the combination of the CpG island and the predicted transcription start site indicates the presence of gene starts.

C. Decision Trees

A decision tree is a decision supporting tool which uses a graph or decision model and their possible consequences, resource cost and utility including chance event outcomes. Decision trees are helpful to identify a strategy most likely to reach an objective and commonly used in operation research, especially in decision analysis [106]. They accurately differentiate between coding and non-coding DNA for sequences ranging from 54 to 162 base pairs in length [107].An advantage of decision trees over techniques such as linear discriminent analysis is that they perform more functions of feature selection automatically, the user can enter a large number of features, including irrelevant data, and the decision tree algorithm will use only a subset in building the tree. In that observation, the task of distinguishing between subsequences that are either entirely encoding or entirely non-coding was addressed. An integrated system MORGAN [42] is a tool to identify genes in the vertebrate DNA sequences that include its decision tree routine and algorithms for splice site identification and its performance on a standard database. It uses an OC1 decision tree system made for separating coding and non-coding DNA. Depending on a separate scoring function, the optimal segmentation takes a subsequence and indicates whether an exon is present in the given sequence or not. In MORGAN, the scoring functions are the collection of decision trees which are combined to give the estimate of a probability. Internal nodes of a decision tree are property values that are tested for each sub sequence passed to the tree which can be various coding measures (e.g., hexamer frequency) or signal strengths. MORGAN correctly identifies 58% of the coding exons, i.e. both the beginning and the end of the coding regions in a DNA sequence. Another well-known gene finder, Glimmer M [35] developed specifically for eukaryotes, uses decision trees hybridized with Interpolated Markov Model (IMM) and dynamic programming. This system is based on bacterial gene finder Glimmer. It selects the best combination from all the possible exons using dynamic programming to consider for inclusion in a gene model. The best gene model is a combination of the strength of the splice sites and the scores of the exons produced by IMM. A scoring function is built on the basis of decision trees to estimate the probability that a DNA subsequence is coding or not. The types of subsequences, which are estimated, are: introns, initial exons, internal exons, final exons and single exons. The average value of the probabilities obtained with the decision trees is calculated and used to produce a smoothed estimate of the probability that the given subsequence is of a particular type. When the IMM score over all coding sequences exceeds a preset value (threshold), only then the gene model is accepted.

D. Support Vector Machine

Support Vector Machines (SVMs) are the set of related supervised learning methods used for classification and regression [36]. They belong to a family of generalized linear classifiers. In other terms, Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize the predictive accuracy while automatically avoiding over-fit to the data. SVMs can be defined as the systems which use hypothesis space of linear functions in a high dimensional feature space, trained among learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. SVM became famous when, using pixel maps as input; it gave accuracy comparable to sophisticated neural networks with elaborated features in a handwriting recognition task [34]. It is also being used for many applications, like as hand writing analysis, face analysis and so forth, especially for pattern classification and regression based applications. The SVMs have been developed by Vapnik [36] and gained popularity due to many promising features such as better empirical performance.

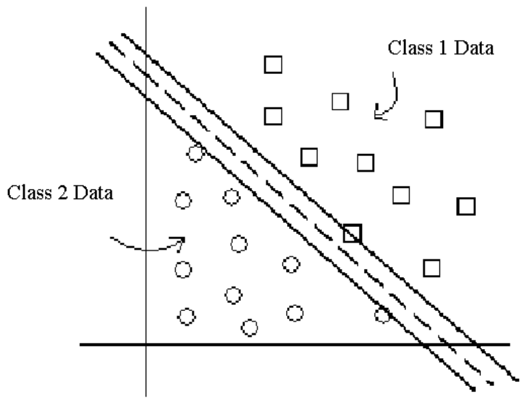

Figure 4: Support Vector Machine

The formulation uses the Constitutional Risk Minimization (SRM) principle, which has been shown to be superior [37] to traditional Empirical Risk Minimization (ERM) principle, used by conventional neural networks. SRM minimizes an upper bound upon the expected risk, where as ERM minimizes the error on the training data. It is difference which equips SVM with a greater ability to postulate, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but promptly they have been extended to solve regression problems [38]. Learning with structural risk minimization is the central idea behind SVMs, and this is seductively accomplished by obtaining the separating hyper plane between the binary labeled data sets (± 1) that separates the labeled data sets with a maximum possible margin [39]. SVM has been found to be successful when used for pattern classification problems. Applying the Support Vector access to a particular practical problem involves resolving a number of questions based on the problem definition and the design involved with it. One of the major challenges is that of choosing an appropriate kernel for the given application [37]. Fig. (4) shows support vector machine with hyper plane and margin. There are standard choices such as a Gaussian or polynomial kernel that are the default options, but if these prove ineffectual or if the inputs are discrete structures more elaborate kernels will be needed. By essentially defining a feature space, the kernel contributes the description language used by the machine for examing the data. Once the choice of kernel and optimization criterion has been made the key components of the system are in place. The major strengths of SVM are the training is relatively easy. No local optimal, similar in neural networks. It scales relatively known to high dimensional data and the trade-off between classifier complexity and error may be controlled explicitly. The weakness excludes the need for a good kernel function [40]

III. Conclusion

There are several methods to identify the gene in Neural Network but still it is a very difficult task and need improvement in finding the gene prediction for large genomes. In last few decades, various strategies of gene identification based on HMM and dynamic programming are developed. As gene identification leads to a structural annotation of the genomes that is then used for experimentation, the worth addition to the identifications are going to be given for every expected gene. Given the difficulty of the problem, machine intelligence based strategies have also been applied in recent times because of their hardiness and skill to handle clanging and incomplete/uncertain knowledge. FGENES/FGENESH (species specific gene prediction tool estimation programs) uses Viterbi algorithmic program to go looking for optimum path. GRAIL and GRAIL 2 uses neural network for gene prediction, GRAIL 2 being the advancement on grail. GeneMark uses one homogeneous model for protein coding DNA and homogeneous Markov Model for non-coding DNA finding strategies. But, it has been found that only 2 chronicles book of human desoxyribonucleic acid is coding and also the rest is non-coding. Recently, the promoter is considered as showing in the intergenic region (immediately upstream of the gene), and not overlapping with it, therefore simplifying the fact. There is a demand of the databases that aren't redundant contain reliable and relevant annotations, and supply all necessary links to more knowledge. Though there exist various problems in gene finding, the comparative genome approach seems to be a really promising not only in the field of gene prediction however also for the identification of regulatory sequences and the decoding of junk DNAs. GenomeScan use integrated approaches in database similarities whereas MORGAN uses decision trees and dynamic programming. GenScan and UNVEIL use Hidden Markov Model for the aim. Genie use GHMM and SPLICEVIEW and SplicePredictor uses signal sensor methods. AAT use integrated approach in knowledge similarities while DAGGER Gene recognition is based on DAG shortest path. An equal number of coding and non-coding nucleotides are contained in the training sets used for varied gene.

Acknowledgment

We are very thankful to the reviewers and those who helped in improving the manuscript's quality.

References

[1] Limin Fu Neural Networks in Computer Intelligence,Tata MCGraw-Hill Edition

[2] Ghosh Z, Mallick B. Bioinformatics Principles and Applications. 2nd ed. Oxford University Press: Delhi 2009.

[3] van Wieringen WN, Kun D, Hampel R, Boulesteix AL. Survival prediction using gene expression data: A review and comparison. Comput Stat Data Anal 2009; 53(5): 1590-1603.

[4] Stein LD. End of the beginning. Nature 2004; 431(7011): 915-916.

[5] Stormo GD. Gene-finding approaches for eukaryotes. Genome Res 2000; 10(4): 394-397.

[6] Zhang MQ. Computational prediction of eukaryotic protein-coding genes. Nat Rev Genet 2002; 3(9): 698-709.

[7] Cawley SL, Pachter L. HMM sampling and applications to genefinding and alternative splicing. Bioinformatics 2003; 19 (Suppl 2): ii36-ii41.

[8] Kumar M, Raghava GP. Prediction of nuclear proteins using SVM and HMM models. BMC Bioinformatics 2009; 10: 22.

[9] Claverie JM. Computational methods for the identification of genes in vertebrate genomic sequences. Hum Mol Gen 1997; 6(10): 1735-1744.

[10] Eddy SR. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. BMC Bioinformatics 2002; 3:18.

[11] Burset M, Guigo R. Evaluation of gene structure prediction programs. Genomics 1996; 34(3): 353-367.

[12] Gelfand MS. Prediction of function in DNA sequence analysis. J Comput Biol J Comput Mol Cell Biol 1995; 2(1): 87-115.

[13] Haussler D. Computational genefinding. TIBS, Suppl Guide Bioinform 1998; 23(1): 12-15.

[14] Mathe C, Sagot MF, Schiex T, Rouze P. Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Res 2002; 30(19): 4103-4117.

[15] Fields CA, Soderlund CA. Gm: a practical tool for automating DNA sequence analysis. Comput Appl Biosci 1990; 6: 263-270.

[16] Gelfand MS. Computer prediction of the exon-intron structure of mammalian pre-mRNAs. Nucleic Acids Res 1990; 18: 5865-5869.

[17] Flicek P. Gene prediction: Compare and CONTRAST. Genome Biol 2007; 8(12):233.

[18] Allen JE, Pertea M, Salzberg SL. Computational gene prediction using multiple sources of evidence. Genome Res 2004; 14(1): 142- 148.

[19] Schiex T, Moisan A, Rouze P. EuGène: An eucaryotic gene finder that combines several sources of evidence. Lect Notes Comput Sc 2001; 2066: 111–125.

[20] R. E. Neapolitan, Learning Bayesian Networks. Upper Saddle River, NJ: Prentice-Hall, 2003.

[21] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Mach. Learn., vol. 20, pp. 197–243, 1995.

[22] V. Pavlovi´c, A. Garg, and S. Kasif, "A Bayesian framework for combininggene predictions," Bioinformatics, vol. 18, no. 1, pp. 19–27, 2002

[23] Xiaolin L, Parizeau M, Plamondon R. Training hidden Markov models with multiple observations-a combinatorial method. IEEE T Pattern Anal 2000; 22(4): 371-377.

[24] Pedersen JS, Hein J. Gene finding with a hidden Markov model of genome structure and evolution. Bioinformatics 2003; 19(2): 219-227.

[25] Snyder EE, Stormo GD. Identification of protein coding regions in genomic DNA. J Mol Biol 1995; 248(1): 1-18.

[26] Gelfand MS, Roytberg MA. Prediction of the exon-intron structure by a dynamic programming approach. Biosystems 1993; 30(1-3): 173-182.

[27] Gelfand MS, Astakhova TV, and Roytberg MA. An algorithm for highly specific recognition of protein-coding regions. Genome Inform 1996; 7: 82-87.

[28] Gelfand MS, Podolsky LI, Astakhova TV, Roytberg MA. Recognition of genes in human DNA sequences. J Comput Biol 1996; 3(2): 223-234.

[29] Xu Y, Einstein JR, Mural RJ, Shah M, Uberbacher EC. An improved system for exon recognition and gene modeling in human DNA sequences. Proc Second Intl Conf ISMB 1994; 2: 376-384.

[30] Chen XW, Anantha G, Wang X. An effective structure learning method for constructing gene networks. Bioinformatics 2006; 22(11): 1367-1374.

[31] Han B, Chen XW. bNEAT: a Bayesian network method for detecting epistatic interactions in genome-wide association studies. BMC Genomics 12(Suppl 2): S9.

[32] Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Mach Learn 1995; 20(3): 197-243.

[33] Pavlovic V, Garg A, Kasif S. A Bayesian framework for combining gene predictions. Bioinformatics 2002; 18(1): 19-27.

[34] Mitchell TM, Moore AW. Machine Learning, 10-701 and 15-781, School of Computer Science, Carnegie Mellon University, Pennsylvania, USA. http://www.cs.cmu.edu/~awm/15781/2003/ (Accessed June 17, 2012)

[35] Salzberg SL, Pertea M, Delcher AL, Gardner MJ, Tettelin H. Interpolated Markov models for eukaryotic gene finding. Genomics 1999; 59(1): 24-31.

[36] Vapnik V. The Nature of Statistical Learning Theory. Springer: New York 1995.

[37] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 1998; 2(2): 121-167.

[38] Suykens JAK. Support vector machines: A nonlinear modeling and control perspective. Eur J Control 2001; 7(2-3): 311-327.

[39] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995; 20(3): 273-297.

[40] Ojeda F, Suykens JAK, De Moor B. Low rank updated LS-SVM classifiers for fast variable selection. Neural Networks 2008; 213): 437-449.