

# Research proposal: Improvement of *ab initio* gene annotation

**Alexandre Tchourbanov**

Computer Science Department, University of Nebraska at Omaha,  
6001 Dodge Street, PKI 175A, Omaha, NE 68182-0500, USA

E-mail: [achurbanov@mail.unomaha.edu](mailto:achurbanov@mail.unomaha.edu)

May 5, 2003

## **Abstract**

While genomes of many organisms were sequenced over the last few years, transforming the sequences into meaningful data remains a difficult task. The most important task in reading human genome is to identify genes, its structure and certain functional units, such as exonic and intronic structures, CDS, 3'UTR, 5'UTR, splicing patterns and other features. There was a significant progress made in the last 5 years, but many problems remain unaddressed, or quality of the best software available nowadays is still far from desirable. This document describes different approaches to build gene recognition software, discusses test runs information available, surveys literature and proposes further destination for the research aimed to Ph.D. degree in bioinformatics.

## **1 Introduction**

There are several methods used for the experimental discovery of genes, but they are time consuming and very costly. Accordingly, for the last 15 years researchers have been developing computational methods for gene finding that can automate, or facilitate, the identification of genes.

Correct gene recognition and annotation are among the most important problems to be resolved in the new millennium.

According to Genome Technology, issue No.17, January, 2002, there are two urgent problems in the top 10 list:

- Precise, predictive model of transcription initiation and termination: ability to predict where and when transcription will occur in a genome.

- Precise, predictive model of RNA splicing/alternative splicing: ability to predict the splicing pattern of any primary transcript in any tissue.

There are generally recognized two methods - *homology* based and *ab initio* to approach the problems mentioned above.

Homology methods rely on alignment algorithms, combined with information from *signal sensors* information, to find a gene structure.

*Ab initio* approach determines the gene structure solely based on signals in a sequence plus pattern recognition, without information on homology.

Also, recently researches started combining both methods together to get even stronger results (Mathé *et al.*, 2002; Birney & Durbin, 2000).

Software, currently available, looks only for transcribed regions of a chromosome, which is then called *gene*. A gene is further divided into *exons* and *introns*. Introns are removed by *splicing* mechanism after *transcription* to form a mature *mRNA*. Assembly of the gene into mature mRNA is not always the same; (Mironov *et al.*, 1999) have found that at least 35% of human genes are alternatively spliced - having more than one possible exon assembly. Recent findings estimate more than 300,000 alternative transcripts from 30,000 human genes are available (Bracco, 2002; Black, 2000).

The region between two genes is called *intergenic* and may contain several interesting regulatory motifs, undiscovered genes and other elements function of which we yet to know.

Inside or at the boundaries of various genomic regions, specific functional sites or signals are documented to be involved in a various levels of protein encoding gene expression, i.e. transcription (transcription factor binding sites and TATA boxes), splicing (donor and acceptor sites and branch point), polyadenylation site, translation (initiation ATG site and stop codon), ribosomal binding sites, topoisomerase II binding sites, topoisomerase I cleavage sites, and various transcription factors binding sites.

Sequence similarity (homology) search is a well-established computational method for gene discovery which has been used extensively with considerable success. The alignment procedures are usually implemented through dynamic programming alignment algorithms, described in (Durbin *et al.*, 1998; Waterman, 1995). *Sim4* is a widely used dynamic programming software (Florea *et al.*, 1998) for gene annotation by alignment of cDNA with genomic DNA.

Although sequence similarity search has been proven useful in many cases, it has been shown that only a fraction of newly discovered sequences have identifiable homologs in the contemporary databases. The proportion of vertebrate genes with no detectable similarity in other phyla is estimated to be 50% (Claverie, 1997; Fickett, 1996). This is supported by a recent analysis of human chromosome 22 (Dunham *et al.*, 1999) where only 50% of the proteins are found to be similar to previous known proteins. It is obvious that sequence similarity search within vertebrates is currently limited.

At the same time cDNA alignments provide the best way to find genes nowadays - *ab initio*

gene finders are able to find only a coding part of genes along with the intervening introns, and usually miss the pattern-less non-coding exons and UTRs. At the same time Gene annotation by alignments suffers from contamination, a cDNA library contains only sequences of genes that were actively transcribed under certain conditions; otherwise, no mRNA would be found. Besides, the cDNA sequences do hardly even span the complete mRNA; the single-stranded RNA is easily disrupted or digested before the reverse transcription has reached the opposite end. Alignment produces gene structure only for a few available alternative transcripts and does not depict the whole structure of a gene.

For the reasons stated above, in this proposal I focus entirely on analysis of *ab initio* approaches and methods of sequence analysis based on signal processing.

## 2 Literature survey

There are numerous papers on gene structure recognition and annotation written by computer scientists and biologists. Recent articles demonstrate a well-balanced bioinformatics approach (Yeh *et al.*, 2001; Fairbrother *et al.*, 2002; Burge *et al.*, 1998).

The recent review articles (Rogic *et al.*, 2001; Mathé *et al.*, 2002) discuss general approaches for the gene finding, deficiencies and strengths of certain methods. The papers explicitly demonstrate that there is still plenty of space for improvement on gene finder's performance.

The review article dated back to 90's (Fickett, 1996) admitted that in order to improve performance of gene finders significantly, an extensive collaboration between computer scientists and biologists is needed, as well as deep insight into biological processes. More recent review (Haussler, 1998) bolsters this statement.

In (Black, 2000) discussion refers to the future challenges in alternative splicing, which requires combination of knowledge of global and local cellular mechanisms.

On the biological side the article (Hastings & Krainer, 2001) discusses the splicing research and various problems the biologists address in the new millennium. It is important to realize is that research is moving into direction of exact understanding of how the splicing works.

Thesis (Rutz, 2000) gives an extensive treatment of an early spliceosome assembly - a valuable information for further modelling.

There are seven major pure *ab initio* approach programs available after 1996 Burset and Guigó test (Burset & Guigó, 1996) used for the gene prediction (Rogic *et al.*, 2001), mainly FGENES, GeneMark.hmm, Genie, Genscan, HMMgene, Morgan and MZEF. Some of these programs allow user to change some of the program's parameters (e.g., prior probability for MZEF and exon size in Morgan, depending on the properties of the input sequences. All seven programs might be installed and run locally except for Genie which could be accessed through the Genie Web server [http://www.fruitfly.org/seq\\_tools/genie.html](http://www.fruitfly.org/seq_tools/genie.html).

For each program (Mathé *et al.*, 2002) gives a short description of methods used by the

program, information about its training set, parameter files used when running it, and some characteristics of output format.

**FGENES** version 1.6 Information about this program can be found on the Sanger Center Computational Genomic Group Web site at <http://genomic.sanger.ac.uk/gf/>; details about an earlier version of the program, **FGENEH** can be found in (Solovyev *et al.*, 1995). **FGENES** uses dynamic programming to find the optimal combination of exons, promoters, and polyA sites detected by a pattern recognition algorithm, constructing a set of gene models along a given sequence. The model is very flexible and allows prediction of single and multi-genes in a sequence, that are either complete or partial. The program has been trained on a nonredundant dataset of 660 human sequences extracted from GenBank release 100. Details about the dataset can be found in (Salamov *et al.*, 1998). The type (first, internal, last, single) and location of each exon is specified in the output of the program, and for each exon there is an associated score for the prediction.

**GeneMark.hmm** version 2.2 (Lukashin & Borodovsky, 1998). This program was initially developed for bacterial genefinding (Lukashin & Borodovsky, 1998) and only recently modified to predict gene structure in eukaryotic organisms. A paper about the eukaryotic version of the program has not been published, but from the programs web site at <http://genemark.biology.gatech.edu/GeneMark/> it can be concluded that it uses explicit state duration HMM, which is often used in gene-finding programs (**Genie** and **Genscan**). The optimal gene candidates selected by the HMM and dynamic programming are further processed by a ribosomal binding site recognition algorithm. The dataset used for training is not described. The output is similar to that of **FGENES** but no scores are given.

**Genie** version 1.x and version 2.1, October 1999 (Kulp *et al.*, 1996) Similarly to **GeneMark.hmm** **Genie** uses generalized HMM with arbitrary length distributions associated with some states of the model. The system is described as modular, since each state is trained separately and new states can be easily added. The mechanisms underlying some states are neural networks for splicing sites, with Markov chains for coding regions. The training set is assembled from the human sequences extracted from GenBank release 89.0 (1995), and details describing sequences and filtering processes can be found at <http://www.fruitfly.org/sequence/human-datasets.html>. This dataset has also been used for training other gene-finding systems (**HMMgene** and **Genscan**). **Genie** can predict single or multiple-exon genes and any number of them in the sequence. The **Genie** web site is at [http://www.fruitfly.org/seq\\_tools/genie.html](http://www.fruitfly.org/seq_tools/genie.html).

**Genscan** version 1.0 (Burge & Karlin, 1997; Burge, 1997) In this program, the structure of the genomic sequence is modeled by explicit state duration HMM. The states of this HMM are probabilistic models themselves. Signals are modeled by weight matrices, weight arrays, and maximal dependence decomposition (Burge, 1997), a new technique used for recognition of donor sites. **GenScan** model can predict the absence of genes or the presence of a single gene

or multiple genes, which can be either complete or partial. It also has the option to predict suboptimal exons, which are defined as potential exons with a probability higher than a certain threshold but which are not contained in the optimal parse of the sequence. This type of exon can potentially represent alternatively spliced exons. **Genscan** was trained on Kulp and Reeses dataset of human genomic sequences, and an additional set of 1999 human cDNA sequences was used for training the coding region HMM. The maximal length of the input sequence for this version of **Genscan** is 200 kBp. The output of **Genscan** is similar to the output of the other programs, giving information about exon location and their probabilistic score, but scores for other sequence features such as splicing sites are also given. The Web version of **Genscan** is at <http://genome.mit.edu/GENSCAN.html>.

**HMMgene** version 1.1d (Krogh, 1997) The program is based on HMM, and is trained using a criterion called conditional maximum likelihood, which maximizes the probability of correct prediction. If the sequence analyzed already has some subregions identified (hits to EST or protein database, repeated elements), those regions can be locked as coding or noncoding and then submitted to **HMMgene**. The underlying gene structure model can predict both partial and complete genes in sequence and any number of them. The program has the option to give more than one prediction, which could indicate alternative splicing of the gene in the sequence. The dataset of human single and multi-exon genes collected by Kulp and Reese was used for the training of this program. The output is given in GFF format, slightly different from that used by **Genie** it does not give the location of the splicing sites, but only of the exons, whose type is also specified. **HMMgene** Web site is at <http://www.cbs.dtu.dk/services/HMMgene/>.

**Morgan** version from April 1998 (Salzberg *et al.*, 1998) [version from June 1997]. The underlying method behind **Morgan** is a combination of decision trees, dynamic programming and Markov chains. The most distinctive technique used is a decision tree classifier that classifies subsequences into different classes: initial, internal, final exon. **Morgan** has been trained on the Burset and Guigó dataset of 570 sequences containing only multi-exon genes, and for that reason its prediction is limited to only this class of genes. In addition, it is not capable of analyzing sequences that contain symbols other than A, C, G, and T (e.g. N, M, R, Y). **Morgan** has the standard output with exon locations and probability scores. The recommended length of DNA sequence is up to 200 kBp.

**MZEF** version from April 1998 (Zhang, 1997) **MZEF** uses a quadratic discriminant function to distinguish between two classes: coding and noncoding. Its training set consists of 3440 human exons extracted from GenBank release 87.0, and the program is trained to predict only internal coding exons. The output of the program gives the location of every internal exon predicted, along with a probability score for it and some other measures for different reading frames. **MZEF** can only analyze sequences shorter than 200 kBp. The program has an option to set the prior probability for the sequence analyzed which depends on gene density and G+C content of the sequence. The programs web site is at <http://sciclio.cshl.org/genefinder/>.

### 3 Methods of *ab initio* gene annotation

According to (Burge, 1997), there are four generations of *ab initio* gene finders:

**The first** generation used statistics on coding and non-coding regions to approximately locate exons;

**The second** generation combined these statistics with models for splice sites to exactly locate exons;

**The third** generation combined multiple exons in a model for a single gene;

**The fourth** generation was finally able to predict multiple and partial genes on both sides of a long genomic sequence.

One can imagine two opposing schools of thought on gene modelling - *pragmatic* and *biochemical* according to (Burge, 1997):

**Pragmatic** point of view says that one should combine all known discriminatory properties of introns, exons, and other sites into composite function for prediction, weighting each property by an appropriate factor derived by certain trial and error heuristic, some statistical or machine learning approach.

**Biochemical** point of view is to mimic biochemical machinery on computer system by building an appropriate model. The model parameters and logical framework should be based off the real biological processes of transcription, polyadenylation, capping, splicing (including alternative expression) and other functional transformations.

Recent gene finders consist of *signal sensors* that identify pattern that positionally conserved nucleotides such as the splice sites found at the intron/exon boundaries, and of *content sensors* that identify regions with a statistically significant compositional bias, such as coding exons.

There two ways to combine this information further into prediction:

**Search-by-content** algorithms identify constituent gene fragments by using probabilistic composite measures, such as codon bias, CpG contents and other features.

**Search-by-signal** algorithms make predictions based on the detection of certain consensus signals such as GT/AG rule, branch site, polypyrimidine tract and etc.

The *search-by-content* could easily be an additional source of information needed to resolve weak, indistinct or ambiguous set of signals. A very common source of additional information to predict splicing is coding region biases (3 rd codon position: 90% are A/T; 10% are G/C). The measurement of codon bias probably has almost nothing in common with the way a cell

recognizes and expresses genes (Fickett, 1996). Another source of information, CpG islands, often serve as a clue in promoter finding for a fraction of genes (Ohler, 2001). DNA methylation has no direct effect on the chromatin structure, and no direct evidence for specific protein interactions with methylated regions that are associated with chromatin structure has been reported (Ohler, 2001). Since the CpG region is consequence of biological processes during transcription, the feature is mostly non-biological. The search-by-content methods result into approximate exon boundary recognition, they are incapable of contributing into recognition of non-coding or short exons(Ohler, 2001; Mathé *et al.*, 2002). Other probabilistic bias features hardly provide any deep insight into biological machinery.

On the other hand, *search-by-signal* based recognition is the closest to the biological process, since most of the hexamer signals correspond to binding sites of certain transcription factors (Fairbrother *et al.*, 2002), snRNP binding locations, catalytic centers and highly conserved stacked regions for self-spliced introns (Cech, 1989). A power calculation (number of words of size  $w$  is  $4^w$ ) dictates the use of a word size of six, which is comparable in size to the binding sites of many known RNA binding factors (Fairbrother *et al.*, 2002).

According to (Hu *et al.*, 2000) one way to formulate our gene recognition problem is that an unknown genomic sequence has been generated by some unknown, highly complex evolutionary stochastic process in nature  $G$ . We can abstractly view  $G$  as a stochastic generator of genomic sequences with some distribution:

$$G(s_1, \dots, s_n, c_1, \dots, c_n) = p, \quad 0 \leq p \leq 1 \quad (1)$$

where

$S = s_1, \dots, s_n$  - sequence of nucleotides  $n$  symbols long,

$C = c_1, \dots, c_n$  - sequence of states, where  $s_i$  symbol is in the  $c_i$  state

An example of this approach could be the GenScan model (Burge, 1997) shown in Figure 1.

### 3.1 Sinal sensors

There are numerous signal sensors of different efficiency. The most efficient for donor splice signal recognition is MDD and recently developed Bayesian Tree Method (Cai *et al.*, 2000)

**Consensus string** Based on most frequently observed residues at a given position.

**Pattern recognition** Flexible consensus strings.

**WMM** One of the earliest and most influential approaches to modelling the acceptor splice signal and other biological signals has been the Weight Matrix Method (WMM).

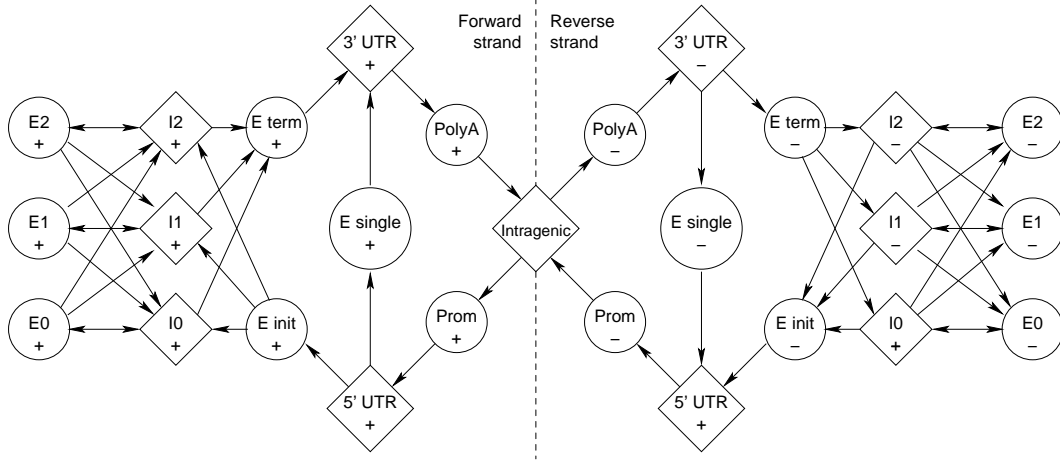


Figure 1: GenScan model of a gene (Burge, 1997). Each circle or diamond in the figure represent a particular functional element type (state) of a gene or genomic region.  $E_k$  ( $0 \leq l \leq 2$ ) - phase  $k$  internal exon,  $I_k$  ( $0 \leq l \leq 2$ ) - phase  $k$  intron

In this approach, nucleotides in a signal of length  $\lambda$  are assumed to be generated independently according to position-specific probability distributions. Under such a model, the probability of generating a particular sequence  $X = x_1, x_2, \dots, x_\lambda$  under the signal model is given by

$$P(X) = \prod_{i=1}^{\lambda} p_{x_i}^{(i)},$$

where  $p_j^{(i)}$  is the probability of generating nucleotide  $j$  at position  $i$  of the signal, which is typically estimated from the positional frequency  $f_j^{(i)}$  observed in a set of aligned signal sequences.

WMM scoring example:

$$P(AGCTGGACTCG) = P(A) \times P(G) \times \dots \times P(C) \times P(G)$$

**WAM** A natural generalization WMM method, termed Weight Array Model (WAM). The WAM model is essentially an inhomogeneous first-order Markov model which differs from the WMM model in that it allows for dependencies between adjacent positions. Under this model, the probability of generating the sequence  $X$  is:

$$P(X) = p_{x_1}^{(1)} \prod_{i=2}^{\lambda} p_{x_{i-1}, x_i}^{(i-1, i)}$$

where  $p_{j,k}^{(i-1, i)}$  is the conditional probability of generating nucleotide  $k$  at position  $i$ , given nucleotide  $j$  at position  $i - 1$ . This quality is typically estimated from the ratio  $f_{j,k}^{(i-1, i)} / f_j^{(i-1)}$ , where  $f_{j,k}^{(i-1, i)}$  is the frequency of the dinucleotide  $j, k$  at positions  $i - 1, i$  of the signal.

WAM scoring example:

$$P(AGCTGGACTCG) = P(A) \times P(G|A) \times P(C|G) \times \dots \times P(C|T) \times P(G|C)$$

**WWAM** In order to have enough data to describe potential higher-order biases, the approach chosen in (Burge, 1997) was to pool data from a "window" of adjacent signal positions, constructing what might be termed a Windowed Weight Array Model (WWAM). A second-order WWAM was therefore constructed in which data from positions  $i - 1, i - 1, i, i + 1$  and  $i + 2$  were averaged to give the second order Markov transition probabilities at position  $i$  for  $-38 \leq i \leq -21$ .

**MDD** If there are strong dependencies between non-adjacent as well as adjacent positions, then we need Maximum Dependence Decomposition (MDD), which basically a decision tree bifurcating at the most influential residuals.

**HMM** Hidden Markov Models (HMM) uses a probabilistic framework to infer the probability that a sequence correspond to a real signal.

**BN** Bayesian Network (BN) outperforms the MDD (Cai *et al.*, 2000). The Bayesian tree Model is obtained:

1. Computing the correlation between all residuals,
2. Finding the maximum spanning tree by linking positions of high correlations,
3. Choose root variable and orient the tree,
4. Computing the conditional probability for each linked position.

**ANN** Artificial Neural Network (ANN) are trained with positive and negatives example and "discover" the features that distinguish the two sets. Example of neural network is shown in Figure 2.

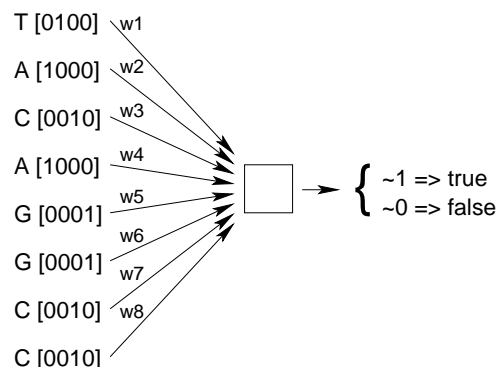


Figure 2: Neural network sensor example from (Horton & Kanehisa, 1992)

The Markov model of step length 5, considered the most accurate, is based on phase-specific counts for oligonucleotides of length 6 (Fickett & Tung, 1992). For the purposes of exonic classification, the most efficient measure, according (Fickett & Tung, 1992), appears to be the hexamer oligonucleotide frequency, which usually corresponds to a protein binding site (Fairbrother *et al.*, 2002).

### 3.2 Gene annotation methods

There are several methods for gene annotation available nowadays. We describe the most significant of them below. Most of the methods include classification algorithms usually implemented through a Dynamic Programming (DP) framework combining several features to make the most plausible prediction fig. 3.

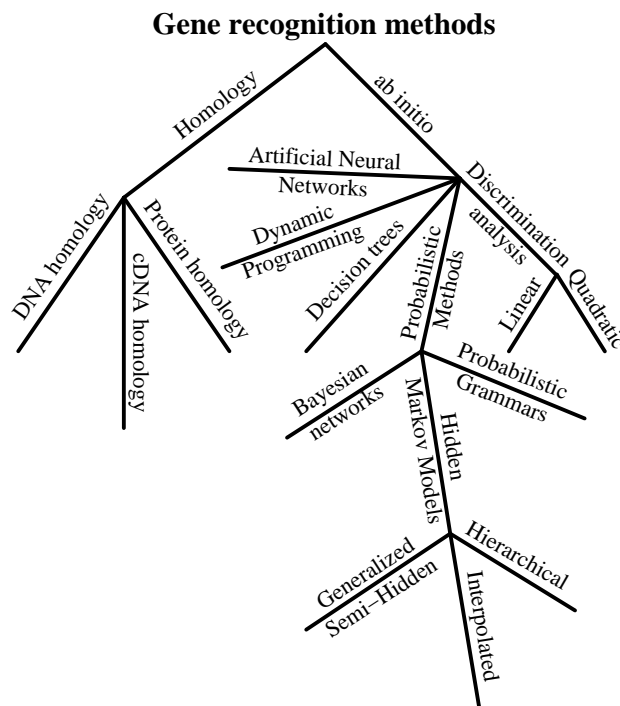


Figure 3: Methods available for gene structure prediction

Here we present non-exhaustive list of the major *ab initio* methods:

**Hidden Markov Models (HMMs)** There are numerous Hidden Markov Models sharing the same feature - we can decompose complex joint distribution over a set of variables using the chain rule of probabilities combining the Conditional Probability Distributions (CPD) of limited sizes. We need to limit the number of interdependencies in the model, keeping the size of CPDs reasonable to keep model tractable. For example, if we have a chain of nodes  $A, B, C, D$ :

$$A \rightarrow B \rightarrow C \rightarrow D$$

we get factoring

$$P(A, B, C, D) = P(A) \times P(B|A) \times P(C|B) \times P(D|C)$$

Thanks to this assumption, and in theory at least, the gene parsing can be defined as the search for an optimal path in a directed acyclic graph. This search is done using the famous Viterbi algorithm (Durbin *et al.*, 1998; Rabiner, 1989), which produces a most likely gene structure and can be considered as a specific instance of the older Bellman shortest path algorithm (Cormen *et al.*, 2001).

There are different variations of the model:

**Hidden Markov Model (HMM)** Hidden Markov Models could be of different order, starting from 0. The most basic case is the first order topology, shown on fig. 4. Even the first order networks are surprisingly good at dna signal recognition (Burge, 1997). Networks of order 5 or allow catching hexamer biases, which is an important source of additional information on human genome parsing (Fickett & Tung, 1992; Burge, 1997).

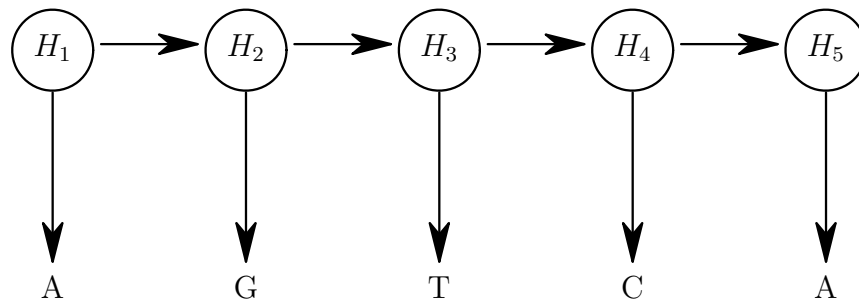


Figure 4: First order Hidden Markov Model

**Interpolated Markov Model (IMM)** Consider the example sequence

AGCTGGACTCG. The fifth order score for a nucleotide A at seventh position will be  $P(A|GGTCC)$ . Sometimes we do not have a significant size of a training set to estimate probabilities of order higher than 5 (it requires estimate of 4096 conditional probabilities put into table as shown in table 1).

If we have biases in oligonucleotides of size greater than 5, we could be missing some data from the training set. To avoid the pitfall, we can make a weighted sum of conditional probabilities. For the example mentioned above IMM score for the nucleotide A at position 7 is the linear combination of  $P(A)$ ,  $P(A|G)$ ,  $P(A|GG)$ ,  $P(A|TGG)$ ,  $P(A|CTGG)$  and  $P(A|GCTGG)$ .

Example the scoring of sequence fragment using IMM:

$$S(AGCTGGACTCG) = S(A) \times S(G|A) \times S(C|GA) \times$$

	A	C	G	T
AAAAA	$P(A AAAAA)$	$P(C AAAAA)$	$P(G AAAAA)$	$P(T AAAAA)$
AAAAC	$P(A AAAAC)$	$P(C AAAAC)$	$P(G AAAAC)$	$P(T AAAAC)$
AAAAG	$P(A AAAAG)$	$P(C AAAAG)$	$P(G AAAAG)$	$P(T AAAAG)$
...			...	
TTTTT	$P(A TTTTT)$	$P(C TTTTT)$	$P(G TTTTT)$	$P(T TTTTT)$

Table 1: Fifth order dependencies in the CPD table of size  $4^5 = 4096$

$$\dots S(C|TCAGGTCG) \times S(G|CTCAGGTC)$$

We illustrate how  $S(C|GA)$  is estimated. If there are enough number of AGC in the training set, use estimate of  $P(C|GA)$ . Estimate of  $P(C|GA)$  is simply number of AGC divided by number of AG in training set.

Rationale is that the higher the Markov model is the better will be the estimate. Otherwise  $P(C|GA)$  alone will not be a good estimate; incorporate lower order model. So compare the number of AGA, AGC, AGT, and AGG with GA, GC, GT, and GG. Use  $\chi^2$  test to obtain weight  $w_1$ . Now use  $S(C|GA) = (1 - w_1)S(C|G) + w_1P(C|GA)$ . This is iterative procedure.

If there are enough number of GC in the training set, use  $P(C|G)$  to estimate  $S(C|G)$  estimate of  $P(C|G)$  is simply number of GC divided by number of G in training set. Otherwise compare the number of GA, GC, GT, and GG with A, C, T, and G. Use  $\chi^2$  test to obtain weight  $w_2$ . Use  $(1 - w_2)S(C) + w_2P(C|G)$  to estimate  $S(C|G)$ ; note that  $S(C)$  is just  $P(C)$ .

$$S(C|GA) = P(C|GA)$$

or

$$S(C|GA) = (1 - w_1)P(C|G) + w_1P(C|GA)$$

or

$$S(C|GA) = (1 - w_2)(1 - w_1)P(C) + w_2(1 - w_1)P(C|GA) + w_1P(C|GA)$$

**Semihidden Hidden Markov Model (semiHMM)** In an HMM, suppose that  $p$  is the probability of the transition from any state to itself. The probability that the process stays in the state for  $n$  steps is  $p^{n-1}(1 - p)$ , which corresponds to 1-shifted geometrically distributed state length only. semiHMM is a stochastic process whose successive state occupancies are governed by a Markov transition matrix (with the restriction that probability of transition from a state to itself is zero) but where the duration is time spent in each state is a (positive) integer valued random variable described by a separate probability distribution  $\tau_i$  which depends on a state type  $A_i$

(in some cases on  $A_{i+1}$  as well). The SMM is strictly more general than HMM. Such a model has been referred to as an explicit state duration HMM (Rabiner, 1989) or generalized HMM (Kulp *et al.*, 1996).

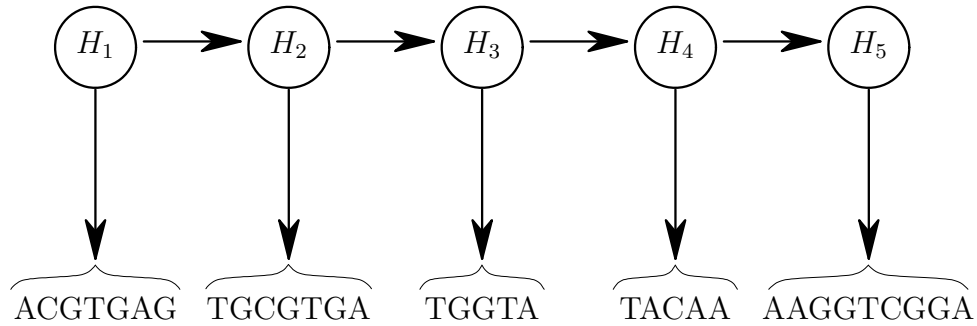


Figure 5: Generalized Hidden Markov Model

We should also mention Three periodic Markov Model, Hierarchical Hidden Markov Model, Variable length Markov Chain, Stochastic Segment Model which could be used. Theory behind these models is beyond the scope of this introduction.

**Discriminant analysis** Dynamic programming is used in to combine the inferred exons in both linear and Quadratic discrimination.

- Linear discrimination analysis Linear discrimination analysis is a standard technique in multivariate analysis. Linear discrimination analysis is used to linearly combine several measures in order to perform the best discrimination between coding and non-coding sequences.
- Quadratic discriminant analysis is similar to linear discrimination analysis, but uses a quadratic discriminant function.

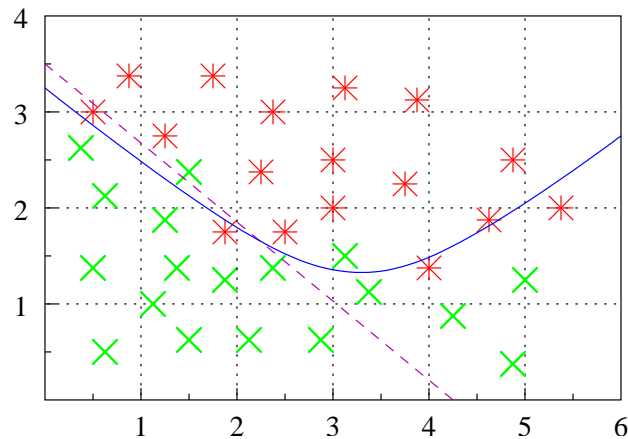


Figure 6: Linear and quadratic discrimination

**Decision tree** Internal nodes of a decision tree are property values tested for each subsequence passed to the tree. Properties can be various coding measures (e.g. hexamer frequencies) or signal strengths. Bottom nodes (leaves) of the tree contains class labels to be associated with the subsequences. Dynamic programming is used to deduce the complete gene structure.

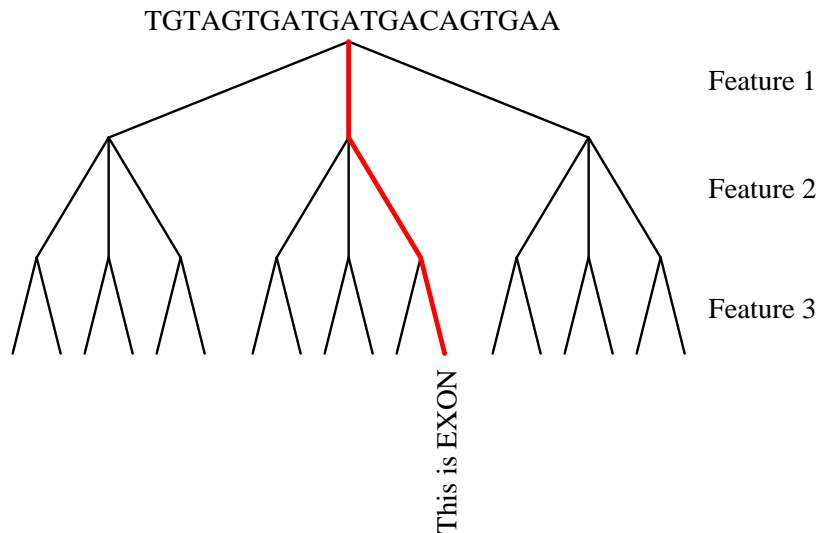


Figure 7: Decision Tree classification

**Neural Network** The neural network is trained with a set of true positives and true negatives examples. For each training example, the neurons are tuned to return the right answer. Dynamic programming is used to deduce the complete gene structure.

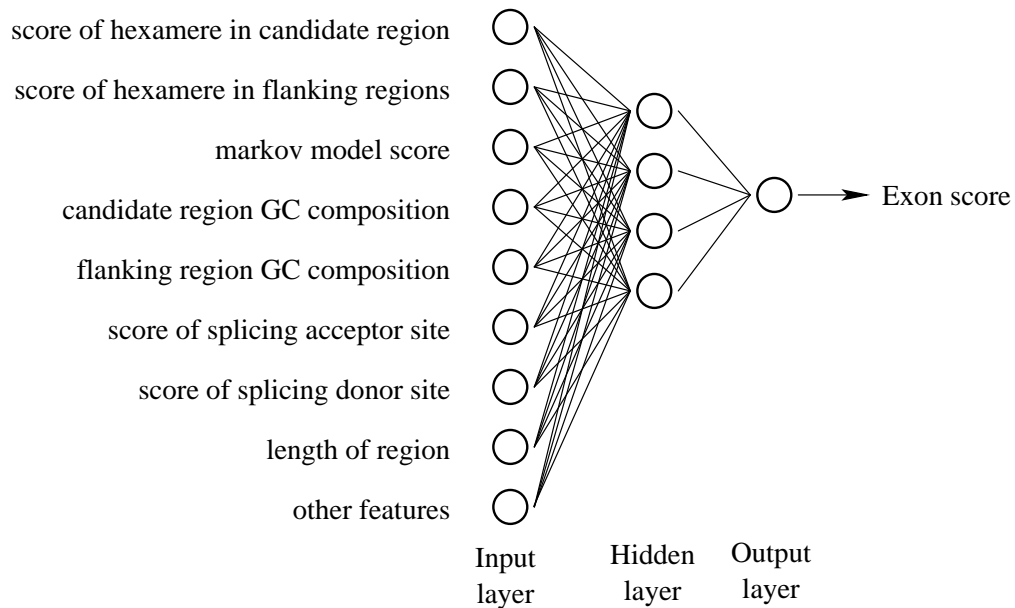


Figure 8: Neural network classification

## 4 Problems with existing approaches

The currently available genefinding performance results must be approached with extreme caution. The primary reason is that they depend very strongly on the difficulty of the genes in the test set, and for some genefinders, on the homology overlap between the genes in the test set and those in the training set that is used to optimize the parameters of the models (Haussler, 1998).

As new category of *completely correct gene prediction* has been added to the list of performance measurements, the **GeneScan** achieves an accuracy of about 40% on the Buset and Guigó dataset in this category (Burge & Karlin, 1997).

As an illustration (Dunham *et al.*, 1999) identified 94% at least partially predicted genes on human chromosome 22 using **GenScan**, but only 20% of genes had all exons predicted exactly.

With the accuracy measures at the nucleotide level as high as 0.91 for the *AC* value for both **GeneScan** and **HMMgene** as well as 0.90 for **GeneWise** with wrong gene prediction equal to 0 (Birney & Durbin, 2000), we might conclude that the problem of computational gene finding is almost solved. However, looking at the results for exon sensitivity and specificity and their average, we see that the goal is still far away (Rogic *et al.*, 2001).

Because of our ignorance of the rules governing splice site choice, today's tools for analyzing genomic sequence provide a picture of these gene products that is highly indistinct (Black, 2000).

There are number of problems, excluding the current far-from-perfection gene finding approaches, that need to be addressed, according to (Rogic *et al.*, 2001; Mathé *et al.*, 2002; Haussler, 1998; Fickett, 1996):

- Existing programs do not find correlation between certain signals, such as the possible correlation between splice site strength and exon size (Zhang, 1998). Usually signals do correlate and this information may improve the performance of a gene finder.
- Problem annotating very long introns. In the human dystrophin gene, some introns are > 100Kbp, and 99% of the gene is composed of introns.
- Very conserved introns and 3'UTR adversely affects homology gene finders.
- Very short exons: some exons are only 3 bp long in *Arabidopsis* genes and probably even 1 bp for the coding part of exons at either end of the coding sequence, meaning that start and stop codons may be interrupted by introns (Mathé *et al.*, 2002).
- Short exons of length of multiple of 3 (typically 3, 6 or 9bp) may be easily missed, since such an omission does not introduce any changes in a frame.
- Overlapping genes. Overlapping usually involves 3'UTR part of genes, but it can also happen that there is a gene within an intron of another gene.

- Polycistronic gene arrangement: albeit this situation was initially thought to occur only in prokaryotes, polycistronic genes have been found in eukaryotes. Many such cases are known for snoRNA genes in plants.
- Frameshifts: some sequences stored in databases may contain errors leading to frameshift, which may result into nonsense prediction.
- Introns in non-coding regions: there are genes for which a genomic region corresponding to the 5' or 3' UTR in the mature mRNA is interrupted by one or more intron(s).
- Non-canonical sites, such as non - GT/AG splice sites : they are not really handled by any program yet.
- Cases of alternative biological processing
  - Alternative transcription start: e.g. three alternative promoters.
  - Alternative splicing. Recent studies report an estimated 35-59% of human genes showing evidence for at least one alternatively spliced form (Mironov *et al.*, 1999). Some of the gene finding programs try to address this issue through sub-optimal exons or suboptimal gene structures (Mathé *et al.*, 2002). However, a more relevant approach would consist of improving the identification of the intronic and/or exonic signals that dictate the choice of the alternative sites (Hastlings & Krainer, 2001; Mathé *et al.*, 2002)
  - Alternative polyadenylation: recent findings estimate 20% of human genes being alternatively polyadenylated (Rogic *et al.*, 2001).
  - Alternative initiation of translation: finding the right AUG is still major concern for gene predicting methods. No current programs consider such alternative form (Mathé *et al.*, 2002).

Gene regulation is another issue to be addressed. The abundant regulatory signals flanking genes, and appearing in introns (and sometimes in exons (Nagel *et al.*, 1998)), combined with regulatory proteins specific to the cell type and cell state, determine the expression of the gene. Gene annotation is not complete until these signals are identified, and the cellular conditions that give rise to differing expression levels for different transcripts are worked out.

This implies, among other things, that future genefinders will need to explicitly take into account experimental data relating to differential expression, along with the other types of data (Kolchanov & et al., 1998). It may be anticipated that this task will occupy genefinding researchers for some years to come (Haussler, 1998).

Signal and content sensors alone cannot solve the genefinding problem, since the signals are weak and unrelated (Haussler, 1998). However, all the necessary information *is available* for

the cell machinery to process a gene in a right way (Lewin, 1999). Nowadays problem is that people do not understand exactly how the machinery works and try to mimic it using different approaches, sometimes irrelevant to the real life (Mathé *et al.*, 2002).

On the example of **GenomeScan**, let us consider the problems affecting performance of all gene finders to a certain degree.

**GenomeScan** (Yeh *et al.*, 2001) and **GeneScan** (Burge & Karlin, 1997) are the state of the art programs nowadays, according to the test runs in (Rogic *et al.*, 2001; Yeh *et al.*, 2001).

Performance of the **GenomeScan** has been tested on the set of CDK gene family, which was under extensive study by UNO bioinformatics group during the summer 2002. The group were able to get precise annotation for the genes, obtaining exons boundaries. Also, the protein sequences necessary for **GenomeScan** program to run properly were obtained. The results of the test run are shown in table 2.

Apparently, the results are not satisfactory, 35% of the boundaries are annotated wrong. Especially the terminal exons, they are either entirely missing or annotated with both boundaries wrong. The CDS portions of the genes are annotated with much better precision, but still occasional misses of boundaries happen.

Exons are	CDK1	CDK2	CDK3	CDK4	CDK5	CDK9	CDK10
Correct	5	4	5	6	6	7	11
Missing	1	-	-	1	-	1	-
Added	-	1	-	-	-	1	-
Both ends are incorrect	1	1		1	1	-	1
One end incorrect	2	2	2	1	5	-	1
Ends predicted correct	12	10	12	13	17	14	23
Ends predicted incorrect	6	6	2	5	7	4	3
Number of annotated exons $AE$							76
The number of exactly predicted exons (true exons) $TE$							33
Number of predicted exons $PE$							63
Sensitivity $ES_n = TE/AE$							0.43
Specificity $ES_p = TE/PE$							0.52
Proportion of annotated exons that are correctly predicted $CR_a$							0.84
Proportion of predicted exons that are exactly correct $CR_p$							0.58
Proportion of predicted exons that are partially correct $PC_p$							0.17
Proportion of missed exons $ME$							0.04
Proportion of wrong exons $WE$							0.25

Table 2: The **GenomeScan** precision for CDK family runs

There were several problems identified with the **GenomeScan** program (Yeh *et al.*, 2001)

- Software does not have an open public source available;
- Program is designed to run fast on the whole human genome for gene discovery, but does not produce exon-intron structure of sufficient quality (see the test results in table 2);

- Both **GenomeScan** and **GeneScan** have sensors of insufficient discriminating power and sensitivity. Use the strongest signals available;
- Since the program has the closed source, researchers can not modify the probabilistic model and incorporate newer evidences becoming available;
- **GenomeScan** is partially based on older **GeneScan** program and incorporates additional information on Protein alignment (**BlastX** homology) with genetic DNA, which is not a pure *ab initio* program anymore;
- Overall statistics on **GenomeScan** is biased to representation of an overall picture, especially nucleotide level one mentioned above, and misses many interesting details, particularly first and last exon(s) boundaries altogether;
- The test set consists of highly curated "Book Genes" (Burset & Guigó, 1996; Burge & Karlin, 1997). **GeneScan** test set, particularly, is based off exons of medium size, coming from genes of medium sizes, which explains why statistics (Burge & Karlin, 1997) is better than reality.

## 5 Objectives

The purpose of the project is creation of an *ab initio* model that mimics the cellular processes, which cannot take advantage of homologies with other proteins and matches to EST or cDNA sequences when deciding where to splice, initiate or terminate translation.

More detailed information on the splicing process, the selection of translation start and the process of polyadenylation may significantly improve such "purist" models (Haussler, 1998; Lewin, 1999).

## 6 What has to be done

The prediction of protein encoding genes structure in higher *eukaryote* organisms needs significant improvement to gain industrial strength.

The general idea of the work proposed is simple - write a program to generate precise gene annotation for *human* and other organisms of similar evolutionary complexity.

The annotation should include precise prediction of all exon boundaries, information on alternative transcripts as well as numerous factors binding sites (Fairbrother *et al.*, 2002). Other factor may also be present in annotation once they prove to be useful for the main purpose.

The program won't work as a scanner of a huge genomic sequences, it will rather emphasize on precision of a prediction on pre-mRNA structure.

The key to the solution of splice site prediction is to sense the context of a certain event. My model will put significant emphasis on the correct representation of a *context* of a certain splicing event.

As mentioned in Section 4, the problems of alternative splicing, first and last exon(s) finding, correct annotation of small and big exons as well as large introns is yet to be addressed. All these problems arise from the deficiency of current methods (usually these are HMMs and DP routines) and incomplete understanding of certain fundamental principles in cell mechanisms as well as a very limited contextual insight.

The result of this work will be the program, written in Java, to approach the problems mentioned in Section 4 in a way superior to the existing *ab initio* methods software packages.

Program will run new probabilistic model based upon the latest discoveries in biology of splicing mechanism as well as advances in artificial intelligence. The program will utilize the principle of context sensitive parsing, which has already demonstrated its strength in (Hu *et al.*, 2000) having performance better than the older pure HMMs. The program will emphasize on a *contextual* background rather than on *statistics* to predict certain feature in a gene.

Program will only use the DNA information to predict the exact location of important sites in a gene plus information on transcription, splicing, enhancing, suppressing and other sites with precision higher than available nowadays for the most of the *ab initio* programs.

## 6.1 Requirements

There are several requirements I have in mind, most of them are coming from (Burge, 1997):

1. Ideally, the gene recognition software needs to have a stable performance pattern, regardless of the gene structure.
2. Trade of speed for precision is desirable.
3. All model parameters should be explicit and tunable (i.e. no hidden neural network weights), have simple intuitive interpretation, and be estimable from available sets of known human genes.
4. The model should be computationally tractable for size up to 200 kBp, or some heuristic should be used (Monte Carlo and similar) in case of intractability of the model.
5. The model should be capable of assigning a measure of reliability to each predicted exon (or gene) so that, for instance, PCR primers could be designed based on the portions of the prediction which are most certain.
6. The method should be robust with respect to C+G content of the sequence. Some programs demonstrate poor performance on A+T rich sequences.

7. Program should predict alternative transcripts.
8. The method should be capable of finding new genes without protein or cDNA homology, only knowledge on certain signals is allowed to use for a prediction.
9. New evidence should be pluggable into the model.

## 6.2 Principles

There are several principles and techniques I would like to use while working on the project

- Model designed for the purposes of this thesis needs to be purely probabilistic, since the whole cellular machinery works through mechanism of *chances*.
- Since additional information on gene structure and particular signals becomes periodically available, an easy way needs to be found to combine these new sources of evidence. Intelligent agent infrastructure may be used to incorporate new beliefs.
- In order to make the best possible prediction of a gene structure, all available information needs to be combined in a most efficient way.
- System needs to be available for different higher eukaryotic organisms, not only for *humans*.
- Probabilistic model of the system will be based on the following principles:
  - K-th order Hidden Markov models of different topologies,
  - Semi Hidden Networks as well as Hierarcical Networks would be useful.
  - Bayesian networks to incorporate prior beliefs and encode causal relationships,
  - Stochastic grammars may be useful,
  - Different heuristics, such as Monte Carlo, will be helpful to make the probabilistic models run in a reasonable amount of time, as well as estimate for information contents.
- Program should trade precision for speed.

## 6.3 Things to do

- Build data set of the whole human genes available at a given moment in RefSeq and periodically update it.
- Study the signals available for the sequences through the biological literature.

- Design a computational method to predict the combination of signals resulting into alternative splicing behavior. The method should highlight the signals characterizing the splice sites and other structural elements. Information theory methods could be useful at this stage.
- Collect information on all probabilistic models available nowadays for sequence analysis, analyze them and extend the hypotheses to make them useful.
- Based upon signals, their correlation and models discovered, build *probabilistic model* of the splicing mechanism.
- Compare the results on a standard test set used to estimate other *ab initio* approach annotation engines (Burset & Guigó, 1996).

The research could be split into four stages.

1. The analysis of splicing mechanism from biological literature, running and testing certain signal models, identifying correlation between the signals will take the Spring Semester 2003.
2. In the summer I start analyzing models available to combine the evidences from the first stage. In parallel I start building software implementing the probabilistic model.
3. In the fall/winter 2003-2004 the program will be finished and thoroughly tested.
4. In the Spring 2004 the dissertation will be written and defended at the end of summer 2004.

## 7 Current progress

In this section I briefly describe current progress in gene annotation I have done. In the summer/fall 2002 we have build package **GIGOGene** to make gene annotation through the spliced alignment (Tchourbanov, 2002). In the following subsections present basics the of algorithms I used and test run results. We also have started work on Exonic Splicing Enhancer (ESE) and Exonic Splicing Silencer (ESS) use to improve gene annotation quality as described in Subsection 7.3.

### 7.1 Algorithm designed

The **GIGOGene** program designed in the summer/fall 2002 is now in a test phase. The following pieces were designed and implemented in Java programming language:

- Spliced alignment algorithm to recognize gene structure;

- Dynamic programming algorithm to make unambiguous allocation of High Scoring Pairs (HSPs) according to sequential rule of non-interrupted aligned mRNA fragments;
- Algorithm to join two sequences together introducing HSP coordinates translation.

The program searches for homology regions between mRNA and DNA using BLAST and then uses dynamic programming to combine this information in the optimal way. There are several steps involved in gene structure recognition:

1. Run the BLASTN program with curated RefSeq cDNA sequences against all DNA found in the HTG database and all DNA records in the Primate databases found at NCBI. For the best results, the low complexity BLASTN filter must be shut off.
2. Parse the BLASTN output. Identify sequences that score above a certain threshold. These sequences are the most probable candidates for the solution, containing potential exons.
3. Get rid of HSPs that are subsegments properly included into bigger matching segments. Certain noise tolerance required.
4. Disambiguate the ordered sets of HSPs. Each sequence may result into several unambiguous ordered sets of HSPs, as explained in (Tchourbanov *et al.*, 2003b).
5. Build an interval graph of overlapping sequences. Special attention is paid to the continuity of cDNA segments, following in (Tchourbanov *et al.*, 2003b). Connections between sets of HSPs coming from the same sequence is not allowed.
6. Compact the interval graph. This way we identify the biggest composite sequence, containing the maximum number of possible exons. For these purposes we run the all-pairs-longest-path algorithm, presented in (Cormen *et al.*, 2001; Tchourbanov *et al.*, 2003b).
7. We use spliced alignment to identify possible intron boundaries in the DNA sequence. In order to save running time, we use *Anchors* - short nucleotide sequences from cDNA and DNA that are supposedly contain Exon/Intron boundary fragments with Donor/Acceptor signals. Normal anchor does not have mismatches in *M* state (see (Tchourbanov *et al.*, 2003b)). Once we have a mismatch, it may mean the short exon presence. We need to extend the anchor and rerun it with two full exons to identify possible short exons as explained in (Tchourbanov *et al.*, 2003b).
8. Extract exonic structure from the ordered set of introns and write exon structure into the database in a specified format.

## 7.2 Test results

The program was tested on GENIE gene finding data set available at <http://www.fruitfly.org/sequence/human-datasets.html/>. 200 mRNAs were extracted from the test set and Exon Level accuracy was measured (Rogic *et al.*, 2001). We estimated the *sensitivity* ( $ESn$ ) and *specificity* ( $ESp$ ) first according to the formulas

$$ESn = \frac{TE}{AE} \quad ESp = \frac{TE}{PE}$$

where

$TE$  - the number of exactly predicted exons (true exons),

$AE$  - number of annotated exons,

$PE$  - number of predicted exons.

Then we calculate additional parameters characterizing the performance on exon level

$CRA$  - proportion of annotated exons that are correctly predicted,

$CRp$  - proportion of predicted exons that are exactly correct,

$PCa$  - proportion of partially predicted annotated exons,

$PCp$  - proportion of predicted exons that are partially correct,

$OL$  - proportion of predicted exons that overlap the actual exons,

$ME$  - proportion of missed exons,

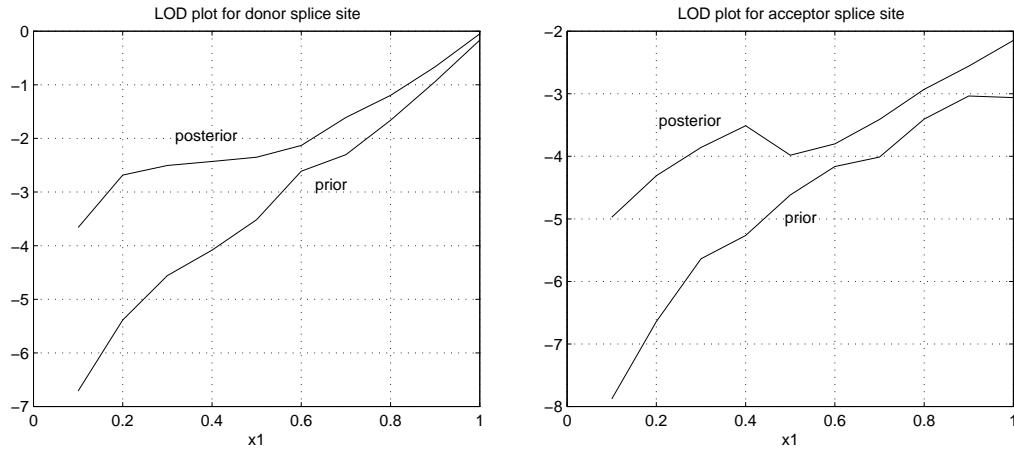
$WE$  - proportion of wrong exons.

$TE$	$AE$	$PE$	$ESn$	$ESp$	$CRA$	$CRp$	$PCp$	$OL$	$ME$
1287	1325	1323	0.97	0.97	0.98	0.97	0.0045	0.0068	0.014

Table 3: Test results for GIGOGene

## 7.3 Using Bayesian Network approach combined with ESE and ESS

In (Tchourbanov *et al.*, 2003a) we described an approach to improve precision of splice sites for human genes. The problem is known to be extremely challenging, since the splicing signals are indistinct and frequent cryptic splice sites confuse signal sensors. There are strong evidences that Exonic Splicing Enhancers and Exonic Splicing Silencers influence Commitment to Splicing at early stages. The paper proposes the use of Bayesian Network (BN), combined with scoring matrices, to improve precision of splice site prediction. We obtain important correlations between the splicing sites and ESE signals and use them to train splicing prediction BN to improve the prediction.



(a) Odds of misclassification for a donor site based solely on SS strength (b) Odds of misclassification for an acceptor site based solely on SS strength

Figure 9: Logarithmic odds of misclassification for the donor and acceptor sites

Figure 9 shows the improvement on donor and acceptor splice sites classification.

We also designed a package to visualize ESE and ESS location relative to exon/intron boundaries, as shown in Figure 10.

### GIGOGene 1.0 predicted gene structure for AL359749.7

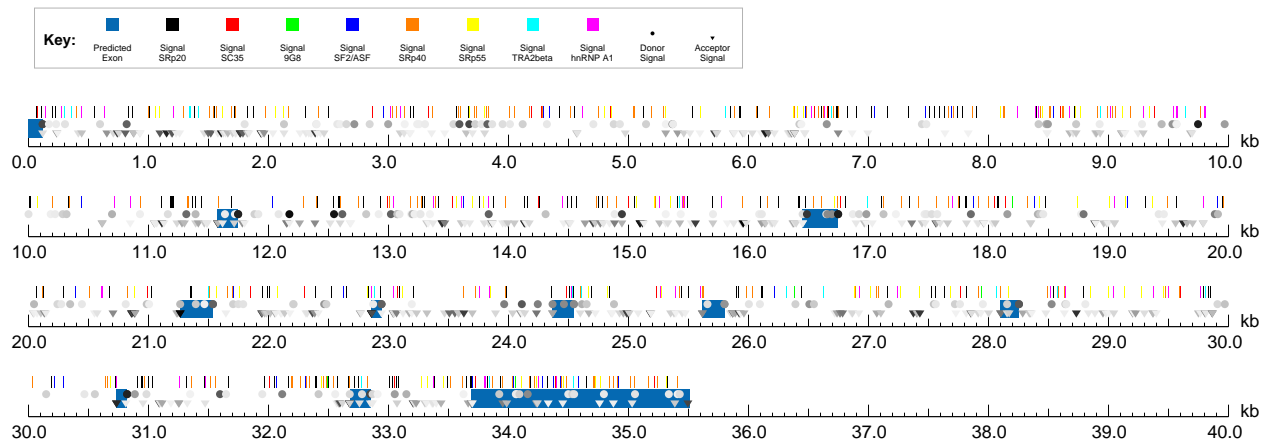


Figure 10: Visualization example of a gene annotation. The darker a donor or acceptor sign area on the diagram, the stronger the motif. Exons are located in the lower signal row, the last 3' non-coding exon is usually the biggest

## 8 Conclusion

The problem of gene structure recognition already has history of more than 15 years. There are usually identified two approaches - homology based and *ab initio*.

Homology based approaches are very efficient at this point, producing results of quality more than 95% (Subsection 7.2), however these methods require information on transcripts,

which is incomplete nowadays. Gene annotation by homology is a temporary mean to leverage our lack of knowledge of a cell machinery. Homology based approaches may serve as a source of exact gene annotation for the test runs (Fairbrother *et al.*, 2002).

*Ab initio* methods search got pretty mature (Rogic *et al.*, 2001) nowadays, but they still require some improvements to run with higher precision (table 2), especially for non-coding exons. The methods require additional evidence, better sensors, correlation, classification and other approaches to combine the evidence into better annotation. *Ab initio* methods are methods of the future.

In this paper I propose to start working on intelligent system for the gene structure prediction. The system will trade precision for the run time as well as incorporating advanced probabilistic model based on Bayesian Networks (HMMs are the tractable subclass of the NP-complete BN class of problems). System will annotate splice sites, start and end of a gene, translation start and end codons as well as possible transcripts.

## References

- Birney, E. & Durbin, R. (2000). Using **genewise** in the drosophila annotation experiment. *Genome Research*, **10**, 547–548.
- Black, D.L. (2000). Protein diversity from alternative splicing: a challenge for bioinformatics and post-genome biology. *Cell*, **103**, 367–370.
- Bracco, L. (2002). Exonhit therapeutics: Novel drugs and diagnostics from alternative splicing. In *Proceedings*, Splicing 2002, available at: <http://www.exonhit.com/splicing2002/data/splicing2002.pdf>.
- Burge, C. (1997). *Identification of genes in human genomic DNA*. Ph.D. thesis, Stanford University.
- Burge, C.B. & Karlin, S. (1997). Predictions of complete gene structures in human genomic dna. *Journal of Molecular Biology*, **268**, 78–94.
- Burge, C.B., Padgett, R.A. & Sharp, P.A. (1998). Evolutionary fates and origins of u12-type introns. *Molecular Cell*, **2**, 773–785.
- Burset, M. & Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.
- Cai, D., Delcher, A., Kao, B. & Kasif, S. (2000). Modeling splice sites with Bayesian networks. *Bioinformatics*, **16**, 152–158.

- Cech, T.R. (1989). Self-splicing and enzymatic activity of an intervening sequence RNA from tetrahymena. Nobel Lecture, available at <http://www.nobel.se/chemistry/laureates/1989/cech-lecture.html>.
- Claverie, J.M. (1997). Computational methods for the identification of genes in vertebrate genomic sequences. *Hum.Mol.Genet.*, **6**, 1735–1744.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. (2001). *Introduction to Algorithms*. MIT Press, 2nd edn.
- Dunham, I., Shimizu, N., Roe, B., Chissoe, S. & et al. (1999). The dna sequence of human chromosome 22. *Nature*, **402**, 489–495.
- Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (1998). *Biological sequence analysis*. Cambridge University press.
- Fairbrother, W.G., Yeh, R.F., Sharp, P.A. & Burge, C.B. (2002). Predictive identification of exonic splicing enhancers in human genes. *Science*, **297**, 1007–1013.
- Fickett, J.W. (1996). Finding genes by computer: The state of the art. *Trends in genetics*, **12**, 316–320.
- Fickett, J.W. & Tung, C.S. (1992). Assesment of protein coding measures. *Nucleic acids research*, **20**, 6441–6450.
- Florea, L., Hartzell, G., Zhang, Z., Rubin, G. & Miller, W. (1998). A computer program for aligning a cdna sequence with a genomic sequence. *Genome Research*, **8**, 967–974.
- Hastings, M.L. & Krainer, A.R. (2001). Pre-mrna splicing in the new millenium. *Current opinion in Cell Biology*, **13**, 302–309.
- Hastlings, M. & Krainer, A. (2001). Pre-mRNA splicing in a new millenium. *Curr. Opin. Cell Biol.*, **13**, 302–309.
- Haussler, D. (1998). Computational genefinding. Review, available at <http://www.cse.ucsc.edu/~haussler/genefindingpaper/paper.html>.
- Horton, P. & Kanehisa, M. (1992). An assessment of neural network and statistical approaches for prediction of e.coli promoter sites. *Nucleic Acid Research*, **20**, 4331–4338.
- Hu, M., Ingram, C., Sirski, M., Pal, C., Swamy, S. & Patten, C. (2000). A hierarchical hmm implementation for vertebrate gene splice site prediction. Tech. rep., Universty of Waterloo.

- Kolchanov, N. & et al. (1998). Genexpress: a computer system for description, analysis, and recognition of regulatory sequences in eukaryotic genome. In *In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 95–104.
- Krogh, A. (1997). Two methods for improving performance of an hmm and their application for gene-finding. In *In Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (eds.T.Gaasterland et al.)*, 179–186, AAAI Press, Menlo Park, CA.
- Kulp, D., Haussler, D., Reese, M. & Eeckman, F. (1996). A generalized hidden markov model for the recognition of human genes in dna. In *In Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology (eds.D.States et al.)*, 134–142, AAAI Press, Menlo Park, CA.
- Lewin, B. (1999). *Genes VII*. Oxford University Press.
- Lukashin, A. & Borodovsky, M. (1998). Genemark.hmm new solutions for gene-finding. *Nucleic Acids Res.*, **26**, 1107–1115.
- Mathé, C., Sagot, M.F., Schiex, T. & Rouzé, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids research*, **30**, 4103–4117.
- Mironov, A.A., Fickett, J.W. & Gelfand, M.S. (1999). Frequent alternative splicing of human genes. *Genome research*, **9**, 1288–1293.
- Nagel, R., Lancaster, A. & Zahler, A. (1998). Specific binding of an exonic splicing enhancer by the pre-mRNA splicing factor srp55. *RNA*, **4**, 11–23.
- Ohler, U. (2001). *Computational Promoter Recognition in Eukariotic Genomic DNA*. Ph.D. thesis, Technische Fakultät der Universität Erlangen - Nürnberg.
- Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, **77**, 257–286.
- Rogic, S., Macworth, A.K. & Ouellette, F.B. (2001). Evaluation of gene finding programs on mammalian sequences. *Genome Research*, **11**, 817–832.
- Rutz, B. (2000). *Commitment to splicing and pre-mRNA retention: early steps of yeast spliceosome assembly*. Ph.D. thesis, Freie Universität Berlin.
- Salamov, A., Nishikawa, T. & Swindells, M. (1998). Accessing protein coding region integrity in cDNA sequencing projects. *Bioinformatics*, **14**, 384–390.
- Salzberg, S., Delcher, A., Fasman, K. & Henderson, J. (1998). A decision tree system for finding genes in dna. *J.Comp.Biol.*, **5**, 667–680.

- Solovyev, V., Salamov, A. & Lawrence, C. (1995). Identification of human gene structure using linear discriminant functions and dynamic programming. In *In Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (eds. C. Rawling et al.)*, 367–375, AAAI Press, Menlo Park, CA.
- Tchourbanov, A. (2002). Spliced alignment through affine gap penalty global alignment. Tech. rep., UNO/PKI.
- Tchourbanov, A., Ali, H. & Deogun, J. (2003a). Using bayesian network approach for splice sites recognition enhancement, available at: <http://csce.unl.edu/~tchourba>.
- Tchourbanov, A., Quest, D., Ali, H., Pauley, M. & Norgren, R. (2003b). A new approach for gene annotation using unambiguous sequence joint, available at: <http://csce.unl.edu/~tchourba>.
- Waterman, S.M. (1995). *Introduction to Computational Biology: Maps, sequences and genomes*. Chapman and Hall/CRC.
- Yeh, R.F., Lim, L.P. & Burge, C.B. (2001). Computational inference of homologous gene structures in the human genome. *Genome Research*, **11**, 803–816.
- Zhang, M. (1998). Statistical features of human exons and their flanking regions. *Human Molecular Genetics*, **7**, 919–932.
- Zhang, M.Q. (1997). Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proc. Natl. Acad. Sci.*, **94**, 95–102.