

---

# Gene Prediction Algorithms

---

Ramiro Pablo Costa

The main purpose of this paper is to conduct a literature review on gene prediction algorithms, i.e. algorithms devoted to predict protein coding regions within the genome of an organism. As soon as genomic sequence data becomes available, correct identification of the gene structure is essential since it constitutes a prerequisite for future biological experiments. However, gene annotation is far from self-evident and in general prone to errors. The widely used approach for genome annotation consists of first employing homology methods, also called *extrinsic methods*, and then gene prediction methods or *intrinsic methods*.

---

## Introduction

The genomic era is here. Every day in labs around the globe new sequence data becomes available and currently whole-genome sequences of roughly 800 organisms are either complete or being determined. This outstanding result leads us to an even more challenging task, the characterization of existing data and to gene prediction. In recent years many computational approaches have come to scene to shorten the gap and the aim of this review is to briefly discuss the different proposed models.

## Comparative genomics methods

The idea of homology comparison between two sequences falls within the concept of evolution, in which divergence between two homologous sequences is the result of continuous accumulation of mutations over the evolutionary time separating them.

Basically the model for homology comparison assumes that only point-mutations are possible, namely only *substitutions*, *insertions*, and *deletions* occurred; and that no particular site was mutated more than once. The latter is sometimes regarded as the *infinite sites model*. In order to quantify the difference between two given sequences, the model defines a cost scheme for each type of mutation (for ex. BLOSUM and PAM matrices) and the total distance between the sequences is obtained as the sum of individual costs. Variations of this model include different penalties for point mutations and for accumulated insertions or deletions (gap penalty). The algorithm proposed by Smith-Waterman is widely used for local sequence alignment, i.e. for search of a stretch of sequence (the query sequence) against a database with sequences of a large number of different organisms.

Once the alignment is produced and its score is reported, the statistical significance is tested. To this end, the query sequence is randomized and a distribution of alignment scores is obtained. From this distribution the P-value is calculated as the probability that the score exceeds the score of the optimal alignment. The *Expect value* (E parameter) is also computed and represents the number of times a match equivalent or a better to the one obtained would be expected to occur in a database search by chance. The lower the E value, the more significant the score. The availability of closely related genomes makes it possible to carry out genome-wise comparisons and analyses of synteny. When two genomes have only

recently diverged, the order of many genes, gene numbers, gene positions and even gene structures (exon–intron organization, splice site usage, and so on) remain highly conserved. New genes can also be identified from direct genome comparisons. By comparing the genomes of several closely related species, conserved regulatory regions can also be easily identified. For these reasons, making use of comparative genomic data is a key challenge for the gene-prediction field. One problem with this approach is that no homologue will be found if the database does not contain a sufficiently similar sequence. Genes may indeed be species-specific and may not have a homologue in another species. Nevertheless, in some cases long divergent species can be of great aid in finding previously overlooked genes. An example of this situation follows, where newly available genomic sequence of the *Fugu rubripes* (Pufferfish) aided in the identification of new human genes.

## Identification of Novel Human Putative Gene Loci [1]

Although three-quarters of predicted human proteins have a strong match to *Fugu*, approximately a quarter of the human proteins have highly diverged from or have no pufferfish homologs, highlighting the extent of protein evolution in the 450 million years since both organisms diverged. All the predicted *Fugu* proteins were searched against the human Ensembl peptides, resulting in matches for 27,779 *Fugu* proteins with a Blast expect score threshold of less than  $10^{-3}$ . This accounted for 22,386 Ensembl human peptides. Of the 8761 *Fugu* proteins below this threshold, a further 1800 matched against the masked human genomic sequence when tblastn was used. Of these, a large number were short matches, which may represent missing exons from gene predictions; however, some represent potentially novel human gene loci. To establish the relation between the matching proteins and existing human gene loci, these putative proteins from *Fugu* gene predictions were used as input to attempt to build human genes through an Ensembl human pipeline. Predictions that overlapped with or were contained within existing loci of human Ensembl were eliminated, resulting in 1260 predictions that were apparently novel. After filtering for low complexity peptides, the remainder were further searched against the National Center for Biotechnology Information (NCBI) nonredundant protein database. A

## Gene-prediction terms and concepts [6]

### Linear Discriminant Analysis and Quadratic Discriminant Analysis

Statistical pattern-recognition methods that are used to categorize samples into two classes. Once samples have been represented as points in space, linear discriminant analysis (LDA) finds an optimal plane surface that best separates points that belong to two classes. Quadratic discriminant analysis (QDA) finds an optimal curved (quadratic) surface instead. Both methods seek to minimize some form of classification error.

### Perceptron Method

A machine learning algorithm for pattern recognition or classification. A perceptron method is based on a simple neural network that begins with an arbitrary initial plane and then iteratively moves the plane in a way that tries to reduce the classification error at each step.

### Hidden Markov Models

Hidden Markov Models (HMMs) represent a system as a set of discrete states and as transitions between those states, each of the possible transitions having an associated probability. Markov models are “hidden” when one or more of the states cannot be observed directly. HMMs are valuable in bioinformatics because they allow a search or alignment algorithm to be built on firm probability bases, and it is straightforward to train the parameters (transition probabilities) with known data.

### Hexamer-Coding Measures

Some methods interpret sequences as successions of *words* (so-called because nucleotides are not independent of each other, but tend to occur together as if in a word) of length  $k$  ( $k$ -tuples); 6-tuples are called hexamers. In-frame hexamer frequencies in a region of DNA have traditionally been used as a powerful way of discriminating coding regions from non-coding regions, as some *words* are more likely to be present in either type of DNA.

### Weight Matrix Method and Weight Array Method

Used for scoring a signal motif site. In the weight matrix method (WMM), a score  $s(x, b)$  is assigned to each position  $x$  for each base pair  $b$ , such that the total score of a motif site can be calculated as the sum of scores at all positions in the site. In the weight array method (WAM), a score  $s(x, w)$  is assigned to each position  $x$  for each word  $w$  of length  $k$  (when  $k = 1$ , the two methods are the same).

### Maximal-Dependence Decomposition (MDD) Donor Matrices

A set of donor splice-site weight matrices that are generated using the WMM, each of which is built for a different class of splicing donor sites in such a way that the dependence between nucleotide positions is minimized.

### Decision Tree

A classification scheme, which can be used, for example, to split a sample into two subsamples according to some rule (feature variable threshold). Each subsample can be further split, and so on.

### Artificial Neural Networks

The key element of the artificial neural network (ANN) model is the novel structure of the information processing system. It is composed of many highly interconnected processing elements that are analogous to neurons and are tied together with weighted connections that are analogous to synapses. Once it is trained on known exon or intron sample sequences, it will be able to predict exons or introns in a query sequence automatically.

total of 961 predictions remained that did not overlap with existing human proteins. About half have some nonhuman match in the NCBI nonredundant database; the remainder were not classifiable by homology. These predicted proteins represent novel putative gene loci in human.

### Linear Discriminant Analysis

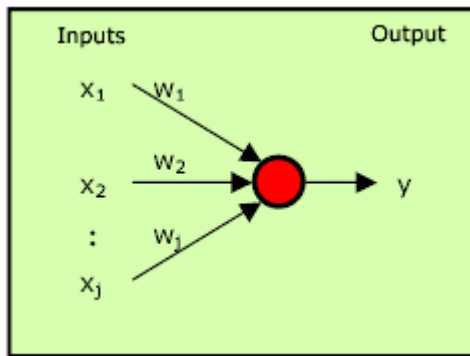
Linear Discriminant Analysis is a classical statistical approach that falls within the Supervised Learning Paradigm. It is widely used for classifying samples of unknown classes, based on training samples with known classes. In order to distinguish one class object from another, two things are needed [7]. Firstly, a set of

feature variables  $X = \{x_\alpha: \alpha=1, \dots, p\}$  and secondly a decision rule (i.e. classifier)  $C$  such that given the measured values  $x_i$  for the  $i^{\text{th}}$  object,  $C$  would be able to map it into either class I or class II. In practice, choosing the set of feature variables that is most discriminative with respect to the two classes is the key. For example, the sex hormone level is a much better discriminative feature variable than the color of skins when classifying people into males and females. Although there are many systematic methods for selecting better feature variables, it is still like a black art, which depends heavily on the master's insight to the nature of the subject. We can represent the  $N$  objects to be classified as  $N$  sample points

$x_i$  in the  $p$ -dimensional feature space. Discriminant theory is to provide us with the mathematical tools for finding the optimal classifier in the sense of minimizing the classification errors. The optimal classifier in LDA is a hyperplane in  $p$ -dimensional space. In particular cases it is desirable to choose more complicated classifiers. To this end, the original feature space is transformed to a new (possibly infinite dimensional) space through a *kernel function* and then the optimal separating hyperplane is computed. In general various kernel functions are used and the one that achieves best discrimination is kept. This corresponds to finding the optimal separating hyperplane in a new (transformed) space.

### Perceptron Method

A single-layer perceptron network consists of one or more artificial neurons in parallel. Each neuron in the single layer provides one network output, and is usually connected to all of the external (or environmental) inputs. The following graph shows a one neuron single-layer perceptron:



The output is obtained as follows:

$$f(x) = \text{sign}\left(\sum_{j=1}^p w_j x_j\right)$$

where  $p$  is the number of inputs.

The perceptron learning algorithm was originally developed by Frank Rosenblatt in the late 1950s and it works as follows:

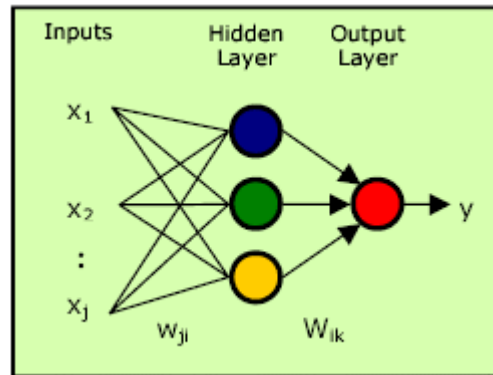
- Initialize the weights  $w_j$  to small random numbers.
- Present a vector to the neuron inputs.
- Update the weights according to:

$$w_t = \begin{cases} w_{t-1} & \text{if } y_t(w_{t-1}x_t) > 0 \\ w_{t-1} + y_t x_t & \text{if } y_t(w_{t-1}x_t) \leq 0 \end{cases}$$

where  $x_t$  is the input vector at time  $t$ ,  $w_t$  are the weights at time  $t$  and  $y_t$  is the desired output ( $y_t = -1$  if  $x_j$  is in class I and  $y_t = +1$  if  $x_j$  is in class II) for the input vector. Learning only occurs when an error is made,

otherwise the weights are left unchanged. The classifier produced is equivalent to finding a separating plane between the objects of different classes.

Single-layer perceptron networks have many limitations and are not computationally complete. They find application in the special type of patterns characterized as *linearly separable*. To solve this problem a multi-layer perceptron can be constructed. For example, a 2-layer perceptron looks like this:



This scheme is capable of producing more complex classifiers and then achieve correct sorting of the objects in the feature space. It is comprised of a hidden layer of neurons connected to the inputs  $x_j$  through a weights matrix  $w_{jk}$ , and also connected to the output neurons (in this case only one) through the weights matrix  $W_{ik}$  (a vector if there is only one output neuron).

### Hidden Markov Models

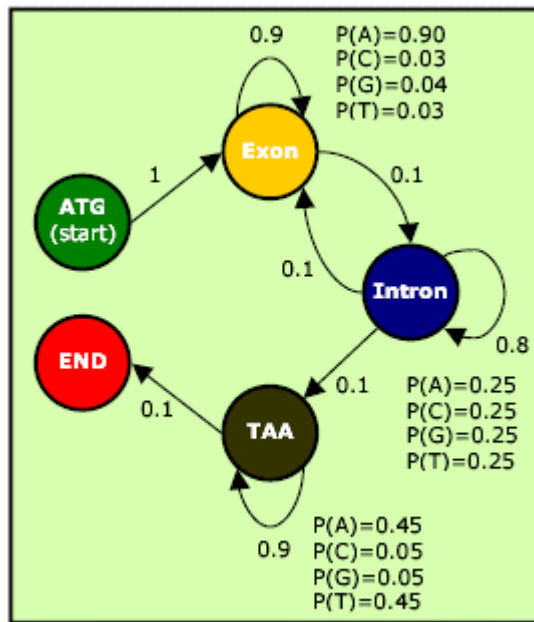
A *Hidden Markov Model* (HMM) is a particular class of statistical model for sequences of discrete symbols. The model consists of a finite set of *states*, each of which can emit a symbol from a finite alphabet with a fixed probability distribution over those symbols, and a set of *transitions* between states, which allow the model to change its state after a symbol is emitted. The transition and emission probabilities may differ between states. Parsing a natural biological sequence into non-coding versus coding region simply consists in determining if a given region is more likely to be generated by the coding versus the non-coding Markov models (previously built using training sets). The model is conceptualized as starting in a designated *start state*, transitioning stochastically from state to state for some variable number of time units, and then terminating when a designated *final state* is reached, all the while emitting symbols (one per state) which when concatenated together in time order form the output sequence of the model.

In the case of gene prediction using a HMM, a DNA sequence is partitioned into disjointed fragments (the states), namely exons and introns. If the conditional probability  $P(s|q)$  of finding a base  $s$  in state  $q$  (which might depend on neighboring bases as specified by the probability model) and the transition probability  $T(q|q')$

of finding state  $q$  after state  $q'$ , for any possible assignment (called a parse  $\Phi$ ) of states  $\{q_i: i = 1, 2, \dots, N\}$  are known, the joint probability is given by:

$$P(\Phi(S)) = P(s_1 | q_1) T(q_1 | q_2) P(s_2 | q_2) \cdots \\ \cdots T(q_{N-1} | q_N) P(s_N | q_N) P_0(q_N)$$

The Viterbi algorithm can be used to find the most probable parse  $\Phi^*$  [8] corresponding to the optimal transcript (exon or intron) prediction. The advantage of HMMs is that more states (such as intergenic regions, promoters, UTRs, poly(A) and frame- or strand-dependent exons and introns) can be added, as well as flexible transitions between the states, to allow partial transcripts, intron-less genes or even multiple genes to be incorporated into a model. Multiple transcript predictions (which might correspond to alternatively spliced transcripts) can also be obtained by using sub-optimal parses. Here is an example of a simple HMM for gene prediction:



Some positive aspects of the HMM approach include:

- Flexible model for dealing with probabilistic processes.
- Does not directly rely on significant similarities with known genes.

Negative aspects include:

- Complicated models need a great deal of training data.
- Gene prediction is biased toward genes with similar features to those used as training set.

### Hexamer-Coding Measures

A wide variety of protein coding measures were proposed and applied to the analysis of genomic sequences [10]. The amount of sequence data available led to the discovery that exons and introns exhibit a distinct usage of nucleotide *words*. This global property probably results from the combination of codon preference with other characteristic periodicities. The contrast in the usage of six nucleotide words (hexamers) was found to be the best single property to predict whether a window of vertebrate genomic sequence was coding or non-coding. The accuracy of the best coding measure was ~70% (i.e., 1/3 of the coding exons were missed, and 1/3 of the ones predicted are not real) for coding windows of at least 50 nucleotides in length. With little prospect of finding better coding measures, scientists in the field began to try various combinations of the existing methods, hoping to improve the overall accuracy of predictions. A straightforward, but effective, way of implementing this concept was through a visual interface, simultaneously displaying graphical representations of the selected coding measures as well as *signal* information (such as start/stop codons and splice sites). This approach, pioneered by Staden, Legouis et al., used a semi-automated protocol to successfully identify the gene for Kallmann syndrome from a 67 kb genomic contig containing only two internal exons (141 + 222 coding nucleotides). The protocol combined (i) the selection of all ORFs larger than 50 bp and flanked by reasonable consensus acceptor and donor splice sites, (ii) ranking the candidate exons according to the hexamer coding measure and (iii) scanning the candidate exons for similarity against protein sequence databases.

### Decision tree

Well established machine learning techniques, decision tree classifiers have been introduced by Salzberg [12] for solving the problem of discriminating coding and non-coding DNA [10]. The internal nodes of a decision tree are property values that are tested for each sub sequence passed to the tree. Properties can be various coding measures (e.g., hexamer frequency) or signal strengths. The bottom nodes (leaves) of the tree contain class labels to be finally associated with the sub sequence. Once classified, the various components are assembled into an optimal gene model using a dynamic programming approach. Briefly, the dynamic programming algorithm is a well established recursive procedure for finding the optimal (e.g., minimal cost or top scoring) pathway among a series of weighted steps. For example, coding measures and signal strengths can be used to compute scores for all subintervals in the test sequence. There are cases in which a neural network is used to combine the various measures into a log-likelihood ratio for each subinterval to exactly represent an intron or exon. A dynamic programming approach is then used to find the optimal combination of introns and exons.

### Weight Matrix Method and Weight Array Method

Numerous models of biological signal sequences such as donor and acceptor splice sites, promoters, etc, have been constructed in the past years [11]. One of the earliest and most influential approaches has been the *Weight Matrix Method* (WMM) introduced by Staden [13], in which the frequency  $p_j^{(i)}$  of each nucleotide  $j$  at each position  $i$  of a signal of length  $n$  is derived from a collection of aligned signal sequences and the product

$P\{X\} = \prod_{i=1}^n p_{x_i}^{(i)}$  is used to estimate the probability

of generating a particular sequence  $X = \{x_1, x_2, \dots, x_n\}$ . A generalization of this method, termed *Weight Array Model* (WAM), was applied by Zhang & Marr [14], in which dependencies between adjacent positions are considered. In this model, the probability of generating a particular sequence is  $\Pr\{X\} = p_{x_1}^{(1)} \prod_{i=2}^n p_{x_{i-1}, x_i}^{i-1, i}$

where  $p_{j,k}^{(i-1, i)}$  is the conditional probability of generating nucleotide  $X_k$  at position  $i$ , given nucleotide  $X_j$  at position  $i-1$  (which is estimated from the corresponding conditional frequency in the set of aligned signal sequences). Of course, higher-order WAM models capturing second-order (triplet) or third-order (tetranucleotide) dependencies in signal sequences could be used in principle, but typically there is insufficient data available to estimate the increased number of parameters in such models. WMM models are used for certain types of signals and modified WAM models can be derived for acceptor splice sites. Here, another model termed as *Maximal Dependence Decomposition* (MDD), is introduced to model donor splice sites.

### Maximal Dependence Decomposition

The goal of the Maximal Dependence Decomposition procedure is to generate, from an aligned set of signal sequences of moderate to large size (i.e. at least several hundred or more sequences), a model which captures the most significant dependencies between positions [11] (allowing for non-adjacent as well as adjacent dependencies), essentially by replacing unconditional WMM probabilities by appropriate conditional probabilities provided that sufficient data is available to do so reliably. Given a data set  $D$  consisting of  $N$  aligned sequences of length  $k$ , the first step is to assign a consensus nucleotide or nucleotides at each position.  $C_i$  is the *consensus indicator variable* (1 if the nucleotide at position  $i$  matches the consensus at  $i$ , 0 otherwise) and the  $X_j$  is the *nucleotide indicator*, identifying the nucleotide at position  $j$ . Then, for each pair of positions, the  $X^2$  statistic is calculated for  $C_i$  versus  $X_j$ , for each  $i, j$  pair with  $i \neq j$ . If no significant dependencies are detected (for an appropriate P-value), then a simple WMM should be sufficient. If significant dependencies are detected, but they are exclusively or predominantly between adjacent positions, then a WAM model may be

appropriate. If, however, there are strong dependencies between non-adjacent as well as adjacent positions, then we proceed as follows: (i) Calculate, for each position  $i$ , the sum  $S_i = \sum_{j \neq i} X^2(C_i, X_j)$ , which is a measure of

the amount of dependence between the variable  $C_i$  and the nucleotides at the remaining positions of the site; and (ii) choose the value  $i_1$  such that  $S_{i_1}$  is maximal and partition  $D$  into two subsets:  $D_{i_1}$  all sequences which have the consensus nucleotide(s) at position  $i_1$ ; and

$D_{i_1}^-$  all sequences which do not. Now repeat steps (i)

and (ii) on each of the subsets,  $D_{i_1}$  and  $D_{i_1}^-$  and on

subsets thereof, and so on, yielding a binary subdivision tree.

### Artificial Neural Networks

One can simply view a neural network [15] as a parallel computational model comprised of a large number of adaptive processing units (neurons). The neurons communicate through a large set of interconnections with variable strengths (weights) in which the learned information is stored.

Neural networks have several unique characteristics and advantages as tools for the molecular sequence analysis problem. A very important feature of these networks is their adaptive nature, where *learning by example* replaces conventional *programming* in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved, but where training data are readily available. Owing to the large number of interconnections between their basics processing units, neural networks are error-tolerant, and can deal with noisy data. Neural network architecture encodes information in a distributed fashion. This inherent parallelism makes it easy to optimize the network to deal with a large volume of data and to analyze numerous input parameters. Flexible encoding schemes can be used to combine heterogeneous sequence features for network input. Finally a multilayer network is capable of capturing and discovering high-order correlations and relationships in input data.

A neural network is characterized by (i) its pattern of connections between the neurons (called its architecture), (ii) its method of determining the weights on the connections (called its training, or learning, algorithm) and (iii) its activation function.

### Neural Networks Architecture

A neural network consists of a large number of simple processing elements called *neurons*. The arrangement of neurons into layers and the connection patterns within and between layers is called the *network architecture*. Each neuron is connected to other neurons by means of



directed communication links, each with an associated weight. The weights represent information being used by the net to solve a problem. Each neuron has an internal state, called *activation level*, which is a function of the inputs it has received. An *activation function* is used to map any real input into a usually bounded range, often 0 to 1 or -1 to 1.

In *feedforward* (FF) nets, the signals flow from the input units to the output units, in a forward direction: the input units receive signals from the outside world; the output units present the response of the net. A multilayer FF net is a net with one or more *hidden layers* between the input units and the output units. In a *fully connected* net every node in each layer is connected to every other node in the adjacent forward layer. If, however, some of the communication links are missing from the network, we say that the network is *partially connected*.

### The Simplest Neural Network

The simplest example of a neural network is the *Perceptron* (Rosenblatt), used for the classification of the special type of patterns characterized as *linearly separable*. A perceptron has only two layers—input and output layers. It computes a linear combination of the network inputs and applies the net input to produce the output using a threshold output function. An elementary perceptron consists of a single output neuron with *adjustable synaptic weights* and a *threshold*. The threshold can be treated as a synaptic weight connected to a fixed input of value 1. Such a fixed input unit is called a *bias unit*. One can use the elementary perceptron to solve a pattern classification problem with only two classes. To perform classification with more than two classes requires the use of more output neurons.

The weights of the perceptron can be adapted on an iteration-by-iteration basis, using an error-correction rule known as the *perceptron convergence theorem* (Minsky and Papert). The theorem guarantees that if a solution exists, the perceptron learning rule will, in a finite number of steps, converge to the correct weights that produce correct output values for all training patterns.

### Application to the Gene Identification Problem

One important area of application of the neural network model is for gene identification. The gene identification problem is tackled by two complimentary approaches: gene search by signal and gene search by content (Staden). The *search by content* methods use various coding measures to determine the protein-coding potential of sequences. The *search by signal* methods identify signal sequences, such as splice sites, which delimit coding regions. Neural networks provide an attractive model in which sequence features for both signal and content can be combined and weighted to improve predictive accuracy.

The identification and analysis of other signals, binding sites or regulatory sites, such as promoters, ribosome-binding sites, and transcriptional initiating and

terminating sites, are also important for the studies of gene regulation and expression. Common approaches to finding functional signals include the *consensus sequence* method, the *weight matrix* method and the neural network method. Neural networks allow the incorporation of both positive and negative examples, the detection of higher-order and long-range correlations, and are not based on the assumption of positional independence. As a result, neural networks are found to be superior to other methods in many studies.

### A Different Approach

Another approach to the problem of gene prediction was borrowed [2] and [3] from the field of Digital Signal Processing (DSP). Genomic information is digital in a real sense, it comes in sequences (character strings) where each element is one out of a finite number of possible entities. DSP techniques deal with numerical sequences rather than character strings, so a proper map from characters to numbers is needed before DSP tools can be used. Of course there are infinite many possible mappings available but, since the goal is to predict the location of coding regions within a portion of DNA sequence, an optimization problem is set in which the parameters (mapping) are determined as the set of values maximizing the predictive power.

### The spectra of DNA

Assume that we assign the numbers **a**, **t**, **c**, **g** to the characters **A**, **T**, **C**, **G**, respectively. Then a DNA sequence of length N can be represented [3] as follows:

$$x[n] = a \cdot u_A[n] + t \cdot u_T[n] + c \cdot u_C[n] + g \cdot u_G[n]$$

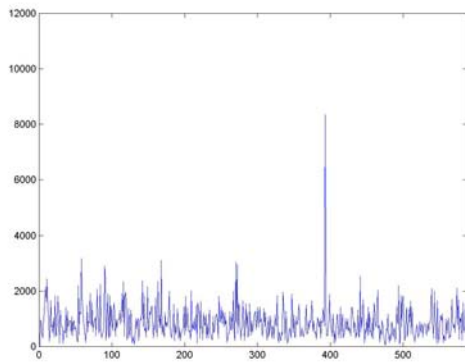
$$n = 0, 1, 2, \dots, N-1$$

Where  $u_x[n]$  represents a *binary indicator function* for the corresponding nucleotide, it takes the value 1 at index n if the corresponding nucleotide is present at that position, and takes the value 0 otherwise.

For pure DNA character strings (i.e. without assigning numerical values) the *Discrete Fourier Transform* (DFT) of the indicator sequences (designated as  $U_x[k]$ ) represent the frequency content of each nucleotide. Combining all four contributions we get:

$$S[k] = |U_A[k]|^2 + |U_T[k]|^2 + |U_C[k]|^2 + |U_G[k]|^2$$

This quantity can be used as a measure of the total spectral content of the DNA sequence at frequency k. In particular, the frequency  $k = N/3$  corresponds to a period of three samples (the length of each codon) and it has been shown [2], [4], [5] that a protein coding region in DNA typically has a peak at that frequency, while non-coding regions have a much smaller value. For example, the figure in the next page shows the spectrum of a length N=1176 coding region in the genome of *Caenorhabditis Elegans*, showing the peak at  $k=392$ .



**Spectra of a coding region in the worm genome**

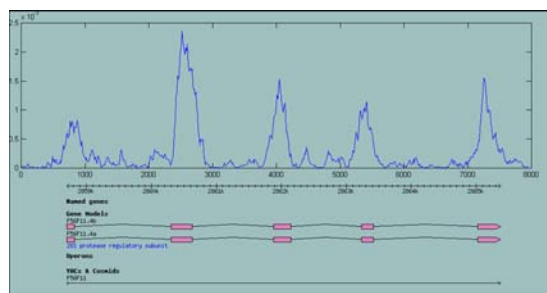
If we define the following normalized DFT coefficients (random variables) at frequency  $k=N/3$ :

$$A = \frac{1}{N} U_A \left[ \frac{N}{3} \right] ; T = \frac{1}{N} U_T \left[ \frac{N}{3} \right]$$

$$C = \frac{1}{N} U_C \left[ \frac{N}{3} \right] ; G = \frac{1}{N} U_G \left[ \frac{N}{3} \right]$$

$$W = a \cdot A + t \cdot T + c \cdot C + g \cdot G$$

Then for each segment of length  $N$  (multiple of three) of a DNA sequence there corresponds a complex number  $W$ . We can think of  $|W|$  as being a random variable and for properly chosen values of  $a, t, c$  and  $g$  it has proved [3] to be a good predictor of whether or not a given DNA segment belongs to a coding region. We can obtain the statistical properties of  $A, T, C$  and  $G$  (their mean and standard deviation) collecting a large number of samples from protein coding regions of DNA and maximize the discriminatory capability of  $|W|$  setting the correct optimization problem. The following example tests the ability of the predictor when applied to a DNA segment of *C. Elegans* which contains the gene designated as F56F11.4a. The graph shows the magnitude of the predictor in the top part and the exon/intron structure of the gene in the bottom part.



**Exon prediction over F56F11.4a of *C. Elegans***

We see from the picture the power of the predictor in this particular case.

## References

- [1] S. Aparicio et al., "Whole-Genome Shotgun Assembly and Analysis of the Genome of *Fugu rubripes*", *Science*, Vol. 297, Aug 2002.
- [2] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes by Fourier analysis of genomic sequences," *CABIOS*, vol. 113, pp. 263-270, 1997.
- [3] D. Anastassiou, "Genomic Signal Processing", *IEEE Signal Processing Magazine*, July 2001.
- [4] V.R. Chechetkin and A.Y. Turygin, "Size-dependence of three-periodicity and long-range correlations in DNA sequences," *Phys. Lett. A*, vol. 199, pp. 75-80, 1995.
- [5] J.W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucleic Acids Res.*, vol. 10, pp. 5303- 5318, 1982.
- [6] M. Q. Zhang, "Computational Prediction of Eukaryotic Protein-Coding Genes", *Nature Reviews, Genetics*, Vol. 3, Sep. 2002.
- [7] M. Q. Zhang, "Discriminant analysis and its application in DNA sequence motif recognition", *Henry Stewart Publications, Briefings in Bioinformatics*, Vol. 1, No. 4, Nov 2000.
- [8] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition" *Proc. IEEE* 77, pp. 257-286, 1989.
- [9] Burset, M. and Guigo, R., "Evaluation of Gene Structure Prediction Programs", *Genomics*, Vol. 34, pp. 353-367, Jun. 1996.
- [10] J.M. Claverie, "Computational Methods for the Identification of Genes in Vertebrate Genomic Sequences", *Human Molecular Genetics*, Vol. 6, No. 10 Review, pp. 1735-1744, 1997.
- [11] C. Burge and S. Karlin, "Prediction of Complete Gene Structures in Human Genomic DNA", *Journal of Molecular Biology*, pp. 78-94, Apr. 1997.
- [12] S. Salzberg, "Locating Protein Coding in Human DNA Using a Decision Tree Algorithm", *Journal of Computational Biology*, Vol. 2, pp. 473-485, 1995.
- [13] R. Staden, "Computer Methods to Locate Signals in Nucleic Acid Sequences", *Nucl. Acids Res.* 12, pp. 505-519, 1984.
- [14] M. Zhang, T.G. Marr, "A Weight Array Method for Splicing Signal Analysis", *Comput Appl Biosci*, Vol. 9 No. 5, pp. 499-509, Oct. 1993.
- [15] C.H. Wu, "Artificial Neural Networks for Molecular Sequences Analysis", *Computers & Chemistry*, Vol. 21, No. 4, pp. 237-256, 1997.