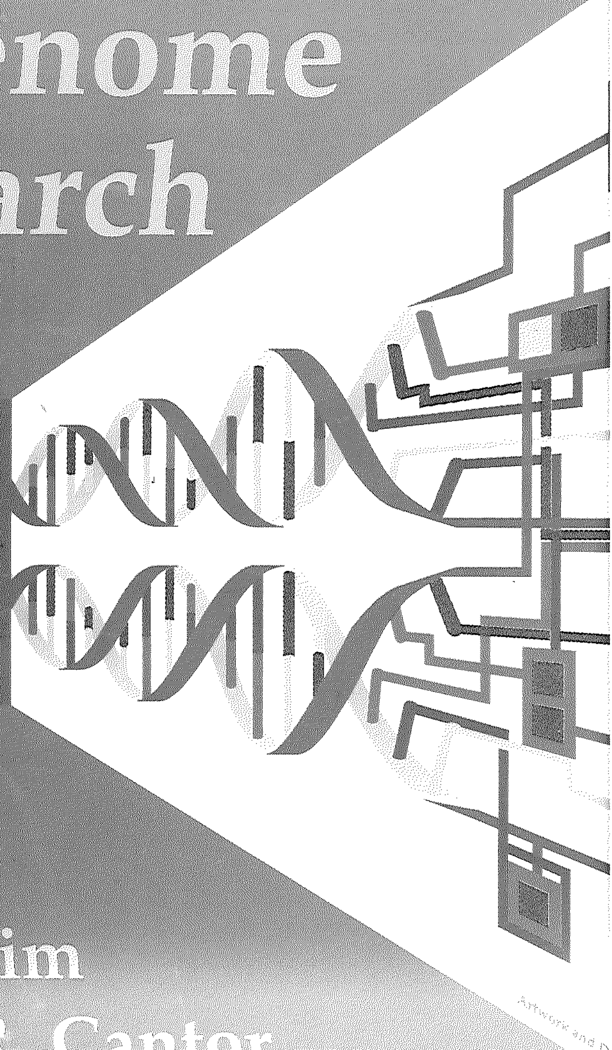
Bioinformatics & Genome Research



Editors

Hwa A. Lim
Charles R. Cantor

World Scientific

Lim
Cantor

Bioinformatics & Genome Research

ISBN 981-02-2401-X

9 789810 224011

Proceedings of the Third International Conference on

Bioinformatics & Genome Research

June 1 – 4, 1994
at Augustus Turnbull III
Florida State Conference Center
Tallahassee, Florida

Scientific Editors

Hwa A. Lim, Ph.D.
Computational Genetics & Biophysics
Supercomputer Computations Research Institute
Florida State University
Tallahassee, Florida, USA

Charles R. Cantor, Ph.D.
Center for Advanced Biotechnology
Boston University
Boston, Massachusetts, USA

World Scientific
Singapore • New Jersey • London • Hong Kong

THE ANALYSIS OF K-TUPLE DISTRIBUTION IN THE EUKARYOTIC PROMOTER DATABASE USING THE WINNER-TAKE-ALL SYSTEM

Patrizio Arrigo*, F. Giuliano*, and L. Milanesi†

* C.N.R, Istituto per i Circuiti Elettronici, via De Marini 6 – 16143 Genova, Italy
† C.N.R, Istituto Tecnologie Biomediche Avanzate, via Ampere 56–Milano, Italy

Abstract

One of the possible ways to identify a putative functional motif on genomic DNA is based on the identification of statistically relevant features on the primary sequence. This paper presents a new statistical approach that involves an Unsupervised Neural Classifier. This methodology appears capable of extracting statistically relevant information from the sequence. In this paper we test this methodology on a set taken from the Eukaryotic Promoter Database.

35.1 Introduction

A *word* is a symbolic string of length ℓ obtained by the concatenation of ℓ symbols extracted from an alphabet [1]. A text can be decomposed into a set of *words* of different lengths. In accordance with the formal language theory we can consider a genomic sequence as a text constituted by a monodimensional juxtaposition of N symbols extracted from an alphabet of four letters. We consider only the canonical bases $\mathcal{A} = \{A, T, C, G\}$; so a gene can be considered as a text constituted by words arranged in sentences. Syntactically we can describe a DNA sequence in the following way:

$$G = \{s_1 \cup s_2 \ldots \cup s_n\}$$

Each s_n is a 'word'. This representation is greatly limited because at present it is not possible to separate the words into different classes based on their functionality. This is the main difference from the natural language processing field and it is for this reason reliable *dictionaries* containing words and their specific functional relation are not available. A system describing the primary sequence has been developed which operates in the same ways as a linguistic parser in a genetic text [2]. The first step in the analysis of syntactical and semantical structure of the gene requires the identification of more relevant words. Many different approaches are applied in order to perform this task: e.g., dynamic programming [3] and statistical mechanics. In 1984 Stormo applied the **Perceptron** model, considered the oldest connectionist model to nucleotide sequence analysis. Generally, only limited regions of the genomic sequence are analyzed. Instead we believe that it is very interesting to analyze the whole sequence in order to detect the presence of singular and more frequent substrings and their relative placements. The aim of this work is to analyze a subset of the Eukaryotic Promoter Database **EPD** [4] by using an Unsupervised Neural Classifier.

35.2 The Eukaryotic Promoter Database

The promoter region is a nucleotide domain related to the regulation of the starting point of the transcriptional process. The eukaryotic promoter region is not as well known as the bacterial promoter. At present all the information related to promoters is collected in a specific database (Eukaryotic Promoter Database). The knowledge about the functional signals present in this region has not yet been fully investigated. In our study we analyzed a subset of EPD (about 40% of the full database) extracted from the region −200 to +200 around the starting point of the transcriptional site.

35.3 The Neural Network

In order to partition the data set we applied a self-organizing neural classifier based on the *Winner Take All* (WTA) methodology [5]. At present we do not know how many k-tuple classes there are nor their occurrence frequencies and for this reason we chose *unsupervised* neural classifiers. Figure 1 shows a schema of the applied network. The neurons reside on a square lattice ($NK_r \times NK_c$) (the maximal dimension is 10^4 neurons). The synaptic weight matrix is a 3D array $[(NK_r \times NK_c \times I)]$ where I is the dimensionality of the input data vector. The WTA uses a distance parameter for the assignment of each pattern to the neuron, from a statistical point of view the distance can be considered as a *dissimilarity* measure. In order to identify the activated neuron, the widely applied measure is the Euclidean metric *Euclidean* ($r = 2$) or *city block* ($r = 1$) metric; following previous papers, we use the Euclidean metric here:

$$d^r(x^n, w_{nk}) = [\sum_{i=1}^{n}(x_i^n - w_{nk,i})^r]^{1/r} \quad (1)$$

We define $\mathbf{X}$ as the input vector data set; $\forall \mathbf{x_n} \in \mathbf{X}$ each neuron of the lattice will present a different level of activation (η). The d^2 is computed $\forall nk \in (NK_r \times NK_c)$; the *winner* neuron must satisfy the following minimization constraint:

$$nk : MIN\{d^2\} \rightarrow \eta_{nk} = 1.0.$$

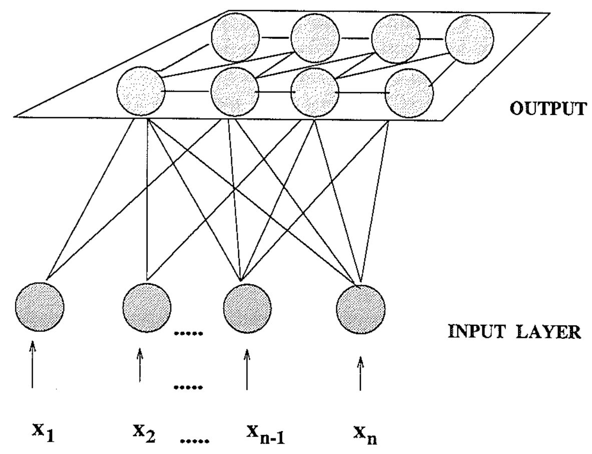


Figure 1. WTA Network.

The activation level of the winner neuron is considered equal to 1.

The second step of the learning phase involves the modification of the synaptic weights associated with the *winner* neuron; the weights adaptation is influenced by a *gain* factor, the *learning rate* ($\alpha(t)$), that linearly decreases with time. This parameter varies in the following range:

$$1e^{-7} \leq \alpha(t) \leq 1.0$$

35.3.1 Data Representation

It is possible to represent the data in different ways, the most widely used knowledge representation for the neural classifier is binary coding. On the basis of clustering theory it is possible to use different types of data representation. The nucleotides can be labeled by a finite number of values (*nominal variable*) [6] and each element is labeled by a specific value. It is important to note that the *dissimilarity* measure must be *invariant* to the data representation. The following nominal coding procedures are available:

- (molecular weights)$^{-1}$
- f_b base frequency where $b \in \mathcal{A}$
- electronic potential (Velicovic)
- ordinal numbering

The $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ was subjected to a *regularization* using the following constraint:

$$\|\mathbf{x}\| = \|\mathbf{w}\| = 1.$$

This data normalization was performed by the following method $\forall i \in d$, where **d** is the input vector dimensionality:

$$\mathbf{x}_i = \frac{x_i}{\sum_{i=1}^{d} x_i} \quad (2)$$

35.3.2 Learning Phase

We defined a set $\mathbf{X}$ of data vectors: $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ to be stored into a 2D neural lattice. The synaptic weight matrix was initialized by uniformly distributed random values

$\mathbf{W}^0 = U(0,1)$, computed by a machine-independent random number generator [7]. The learning phase was performed by a *pattern*, this term signifies that the connection weights were modified after the presentation of each single pattern $\mathbf{x_n}$. The weight vector update involves only the winner neuron, *Best Matching Unit*, indicated by nk^* and is based on the following dynamical system:

$$\mathbf{w}_{nk^*}(t+1) = \mathbf{w}_{nk^*}(t) + \alpha\eta(\mathbf{x}_n - \mathbf{w}_{nk^*}(t)) \tag{3}$$

The $\alpha(t)$ parameter linearly decreases according to the following equation:

$$\alpha(t) = \alpha(t-1) - \alpha_{step} \tag{4}$$

The step is time varying by the ratio: $\alpha_{step} = \frac{epoch}{max\ epoch}$. The activation level of the winner neuron, η, was computed in an adaptive way according to the specific Euclidean distance. The learning convergence is evaluated on the basis of the previously applied parameter [8].

In order to compare the results we considered k-tuples of $\ell = 6$; this amplitude was chosen as an average dimension of promoter signals. There are 4^6 possible combinations of symbols extracted from the nucleotide alphabet $\mathcal{A}$. In this study we only considered the frequency of occurrence for each hexamer obtained by decomposition of the promoter data set. We considered *all* the potentially relevant information stored in the net at the end of the learning phase. In order to highlight the relevant features our study simply took into account the occurrence frequency of each k-tuple. We considered a cut-off frequency of 16 occurrences. However, there are three possible approaches:

- k-tuple frequency
- neuron activation frequency
- mutual information method

35.4 Results

At the end of the learning phase it is possible to obtain a distribution of k-tuples. We considered the hexamers with the highest frequency of occurrence. The following tables report the subsets filtered by the network. Our basic training set consisted of 652 Eukaryotic Promoter region sequences; in this way 16911 six-base-long, non-overlapping patterns were considered. In order to reduce the effect of signal cutting, we processed the training sets obtained from the original set by shifting either one, two, three, or four bases, starting from the initial position.

Table 1. Hexamers extracted from each training set.

k-tuples extracted from each EPD dataset	
k-tuple	Number of k-tuples
1) Without shift	28
2) Shift 1 base	43
3) Shift 2 bases	43
4) Shift 3 bases	45
5) Shift 4 bases	47

Table 1 shows the resulting fraction of filtered hexamers for each run of the program with a cutoff frequency of 16; we extracted only patterns which occur in the dataset more than

Table 2. Most frequent hexamers.

Maximal frequency of k-tuple		
EDP set	k-tuple	Frequency
1) Without shift	TATAAA	34
2) Shift 1 base	TATAAA	40
3) Shift 2 bases	TATAAA	40
4) Shift 3 bases	TATAAA	55
5) Shift 4 bases	TATATA	44

16 times. It should be noted that the shifting procedure increases the fraction of k-tuples capable of passing the frequency cutoff . This effect takes place when a very frequent signal is cut, for instance the TATA-box. Table 2 displays the highest frequency hexanucleotides for each run; from a compositional point of view, this set represents the possible combinations of the canonical TATA-box consensus.

$$\mathrm{TATA}\frac{\mathrm{A}}{\mathrm{T}}\mathrm{A}\frac{\mathrm{A}}{\mathrm{T}}$$

Table 3. Each column shows the most common hexamers for each training set: 1 – without shift, 2 – shift of 1 base, 3 – shift of 2 bases, 4 – shift of 3 bases, 5 – shift of 4 bases.

Most common k-tuples					
k-tuple	set 1	set 2	set 3	set 4	set 5
AAAAAA	33	28	25	30	28
ATAAAA	26	–	25	24	34
ATAAAT	25	19	19	–	25
TATAAA	34	40	40	55	23
TATATA	28	28	–	22	44
TTTTTT	24	21	19	17	17

In Table 3, the most common hexamers in all five analyzed sets are shown. It should be noted that all the hexamers in this table, excluding the AAAAAA and TTTTTT, are part of the general TATA-box consensus. In this case the TATA-box consensus frequency is increased due to the sum of each individual component. The distribution obtained by the program is the actual distribution of pattern without *a priori* alignment and constraints. Tables 4 through 8 show the different subsets of hexamers filtered by the net.

Our methodology is capable of recognizing strong signals like the TATA-box. In its present form, the identification of weak signals needs a more careful investigation at a low frequency cutoff level and it requires a parallel analysis of neuron activation frequency and mutual information. In this paper we show only the most frequent patterns. The simultaneous application of different statistical approaches offered by this connectionist method can enhance the recognition capability of weak signals.

Table 4. Hexamers extracted by the program from the first dataset.

k-tuples extracted from EPD dataset without shift			
k-tuple	frequency	k-tuple	Frequency
AAAAAA	33	ATACTT	18
AAACAA	31	CATGTG	19
AAATAG	21	CCAAAT	19
AAGCAA	19	CGCCAC	23
ACTCAT	19	CGCCCT	19
ATAAAA	26	CGCCGC	18
ATAAAT	25	CGCTAA	18
CGTGGC	18	CTATAA	22
GCGGGG	17	GTGCTG	16
TAAATT	16	TATAAA	34
TATATA	28	TCAGTG	21
TCTCTT	22	TGCACT	23
TTCTGA	21	TTTCCC	16
TTTCTC	17	TTTTTT	24

Table 5. Hexamers extracted from the second set.

k-tuples extracted from EPD dataset with shift of 1 base							
k-tuple	frequency	k-tuple	Frequency	k-tuple	Frequency	k-tuple	Frequency
AAAAAA	28	ATATAA	19	AGAAAA	21	CTCTGC	18
AAAGCA	19	ATGATC	16	AGAAGA	19	CTCTTC	32
AAATAT	19	ATGGCC	21	ATAAAT	19	CTTCTC	16
AAATTC	16	CAAAAA	19	GCACTG	24	GCCACC	23
ACTCAT	19	CAGTGA	18	GCCCTC	18	GCCGCC	18
AACAAC	22	CATCAT	18	GCTAAA	21	GGGCGA	17
ATAAAT	25	CGCTAA	18	GTCTGC	17	GTCTGC	16
AAGGGA	17	CCGCCG	16	GTGGCA	19	TAAAAG	16
AAGTCA	18	CGCCGG	18	TAAATA	24	TATAAA	40
AAGTGA	18	CTCATT	23	TATATA	28	TCAATA	16
ACCTTC	16	CTCCTC	17	TCACAA	17	TCTGAA	20
TTTTTT	21						

Table 6. Most frequent hexamers extracted by the program from 2-base shift of EPD training set.

k-tuples extracted from EPD dataset with shift of 2 bases							
k-tuple	frequency	k-tuple	frequency	k-tuple	frequency	k-tuple	frequency
AAAAAA	25	CGGGGC	16	AAGTGA	18	TAAATA	18
AAAACA	21	CTAAAC	20	ACCTTC	16	TATAAA	40
AACCAA	16	CTATAA	21	AGAAAA	21	TCATTC	33
AATTCT	17	CTCATT	19	AGAAGA	19	TCTGCG	28
ACAAAA	18	CTGAAG	18	ATAAAT	19	TCTTCC	16
ACAACA	21	GCCGCC	17	ATATAA	22	TCTTCG	17
AGAAGG	25	GCTGCT	19	CAAAAG	19	TGCGGG	16
AGGAGG	17	GGCGGC	22	CAACAA	17	TGGCAC	16
AAGTCA	18	GTATAA	19	CAAGCA	17	TGGCCG	19
CACTCC	18	TTCTCT	16	CCACCG	23	TTTTTT	19
CCCAAA	16	CCCTCG	17	CCGCCG	18		

Table 7. More frequent hexamers extracted by the program from 3-base shift of EPD training set.

k-tuples extracted by EPD dataset with shift of 3 bases							
k-tuple	frequency	k-tuple	frequency	k-tuple	frequency	k-tuple	Frequency
AAAAAA	30	GATTTT	19	CAAAAT	19	TAAACT	22
AAAAGA	17	GCCGCC	16	CAACAT	30	TAAATA	18
AAAAGG	22	GCGGCG	18	CACCGC	19	TATAAA	55
AAAAGT	17	GCGGGA	18	CATTCT	20	TATATA	22
AAACAA	19	GGCCGC	18	CCAAAT	18	TCATTC	19
AAAGAA	16	GGCTCC	16	CCTCGT	19	TCTCTT	16
ACTCCG	16	GGGAGC	16	CGCCGC	17	TGAAGT	22
ATAAAA	24	GGGGCG	17	CTATAT	21	TGGCAG	16
ATATAT	19	GTGAAA	18	CTGCGC	24	TTTCCT	17
CAAAAA	19	TAAAAG	16	CTTCGT	16	TTTTCT	17
GAAGTA	20	GAAGTT	18				

Table 8. More frequent hexamers extracted by the program from 4-base shift of EPD training set.

k-tuples extracted by EPD dataset with shift of 4 bases							
k-tuple	frequency	k-tuple	frequency	k-tuple	frequency	k-tuple	frequency
AAAAAA	28	ATAAAT	25	GCCGCC	16	TGCGCC	18
AAAAAG	24	ATATAA	16	GCCGCT	20	TGGAAA	20
AAACTC	19	ATCCAA	20	AGAAAA	21	TTACCT	17
AAAGGA	19	ATTTTT	25	GTATAA	22	TTCGCT	17
AACAGA	16	CAAATG	16	GTTTTC	16	TTCGTC	16
AACATG	27	CATTCT	28	TATAAA	23	TTCTCT	16
ACCATG	24	AAGAAG	20	CCAGCC	16	TATATA	44
AAGGTG	19	CTCCGG	18	TCAGTT	19	TGGCAC	16
AAGTTT	19	CTCCTC	20	TGAAAC	16	TGGCCG	19
ACCGCC	17	GAGAAA	18	TGACCC	17	TTTTTT	19
ATAAAA	34	GCAGAG	16	TGCCTG	17		

Bibliography

[1] Hopcroft, J.E., J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading (1979).

[2] Collado-Vides, J., "Grammatical model of the regulation of gene expression", *Proc. Natl. Acad. Sci.*, **89**, 9405–9409 (1992).

[3] Waterman, M.S., *Mathematical methods for DNA sequences*, CRC Press, Boca Raton (1989).

[4] Bucher, P., "The eukaryote promoter database of Weizmann Institute of Science", *EMBL Nucleotide Sequence Data Library Release 17*, Heidelberg, Germany (1988).

[5] Hertz, J., A. Krogh, R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley, Redwood City, CA (1991).

[6] Kaufman, L., P.J. Rousseuw, *Finding groups in data*, J. Wiley (1990).

[7] Press, W.H., B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, MA (1989).

[8] Giuliano, F., P. Arrigo, F. Scalia, P.P. Cardo, G. Damiani, "Potentially functional regions of nucleic acids recognized by a Kohonen's self-organizing map", *CABIOS*, **9**, 687 (1993).

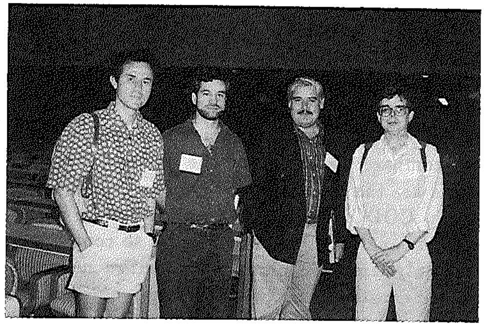
[9] Searls, D.B., "The Linguistic of DNA", *American Scientist*, **80**, 579–591 (1992).

A PRINCIPAL COMPONENT ANALYSIS OF CODON USAGE AMONG LAMBDOID BACTERIOPHAGE

Heinz Hemken*, Gabriel Guarneros†, Francisco M. De La Vega†

*Department of Cell Biology, Center for Research and Advanced Studies of the National Polytechnic Institute, P.O. Box 14–740, Mexico D.F. 07000, Mexico

† Center for Research and Advanced Studies of the National Polytechnic Institute, P.O. Box 14–740, Mexico D.F. 07000, Mexico



The regulation of gene expression is a phenomenon exerted at every step in the cascade of events that leads to the synthesis of a given protein. There are signals at the level of the DNA source code (promoters, operators, etc.), the RNA transcript (splicing and processing sites), and the final protein itself (phosphorylation and processing sites) that must be recognized and acted upon by appropriate regulatory molecules. Most of the detail in current theoretical frameworks of the gene regulation system is at the transcriptional level [1–4], whereas regulatory events in the latter stages of gene expression, namely mRNA translation, are understood in much broader terms. Transcriptional regulation is associated with relatively easily measured and recognized components, and a large body of data and theory has been accumulated. Events occurring at the translational level are less spectacular, dependent on more subtle features of the genetic code, yet no less important.

The non-randomness in the use of synonymous codons in natural coding sequences has been a source of interest as a feature that might be related to levels of gene expression. In prokaryotes, the preferential use of certain synonymous codons over others has been regarded by some investigators as a regulatory strategy aimed at maximizing the translation rate of highly expressed proteins. It has been shown in bacteria that the relative abundance of tRNA species changes notably between different growth rates, and during rapid growth the available species are reduced to a streamlined set of isoacceptors [5], which appear to reflect