Using a Compressibility Measure to Distinguish Coding and Noncoding DNA

Armin Shmilovici[1□], Irad Ben-Gal[^]

[□]*Department of Information Systems Engineering, Ben-Gurion University*
P.O.Box 653, Beer-Sheva, Israel, Fax: +972-3-7399155

[^]*Department of Industrial Engineering, Tel-Aviv University,*
Tel-Aviv, 69978, Israel

armin@bgumail.bgu.ac.il bengal@eng.tau.ac.il

Abstract: DNA sequences consist of protein coding and noncoding regions. Recognition of coding regions is an important phase in gene-finding procedures. This paper presents a new method for distinguishing coding and noncoding DNA regions.

The proposed method implements compressibility measures that results from Variable Order Markov (VOM) models. In contrast to fixed-order Markov models, where the model order is identical for all positions and for all contexts, in VOM models the order may vary – based on a nucleotide position and its contexts. As a result, VOM models are more flexible with respect to model parameterization.

Preliminary experimental results on benchmark datasets demonstrate that the proposed methodology classifies coding and noncoding DNA more accurately than traditional coding measures presented in the literature.

Keywords **:** DNA Compression, Variable Order Markov Model, Coding and Noncoding DNA, Context Tree.

[1]**Corresponding Author**

1. Motivation and Introduction

The identification of protein-coding DNA regions in the genome is a difficult problem. As automated sequencing techniques have began to produce a rapidly growing amount of raw DNA sequences, automating the extraction of information from these sequences becomes a scientific challenge. Computational methods for gene identification - detection of protein coding regions in genomic sequences - typically have two phases: coding region recognition and gene parsing. Currently, it is estimated that less than 2% of the human genome contain protein-coding information, and, hence, the identification of protein-coding regions by purely experimental techniques is expensive and time consuming.

There is not a single standard solution, but rather there is a wealth of different computational approaches. Current gene finding approaches can be roughly divided into three categories (Fickett 1996):

- *Sequence similarity search* between the sequence under study and known proteins libraries.
- *Lexical analysis* involving the identification of special motifs, such as splice donors and acceptor signals (Liu 2001).
- *Methods based on statistical patterns in the coding/noncoding regions*. These methods check for the occurrence of statistical properties that are different in coding and noncoding DNA. The hexanucleotide bias, which was formalized as an inhomogenous 3-periodic fifth-order Markov chain, is probably the most known model for this purpose and was incorporated in genefinders such as GENSCAN (Burge and Karlin 1998). The efficiency of this class depends on the parameterization of the statistical patterns. One particular example of this class is the use of measures of DNA compressibility, which is discussed next.

It is well known that DNA sequences are neither chaotic nor random. For example, it was discovered that DNA sequences, especially in higher eukaryotes, contain copies of essential genes; it is also believed that there are only about a thousand basic protein folding patterns. These phenomena support the conjecture that DNA sequences should be reasonably compressible. However, it is well known that

the compression of DNA sequences is a very difficult task (Grumbach and Tahi, 1994). Paradoxically, it was found that the implementation of standard text-compression algorithms, such as the Unix *compress* procedure, to DNA sequences actually expands the sequence file (Chen et al. 2000).

Specially devised algorithms such as *Biocompress* (Grumbach and Tahi. 1994), *GenCompress* (Chen et al. 2000), and *Cfact* (Rivals et al. 1997) adapted the framework of Lempel-Ziv's *universal coding* (Ziv and Lempel 1997) to DNA sequences. Universal coding methods have been developed to compress data sequences without a prior knowledge on the properties of the generating source. The asymptotic performance of some of these methods is known to converge to that of the optimal non-universal algorithms in terms of compression, prediction, and decision making. However, most of the universal coding methods are less effective with short sequences (Ziv, 2001), such as ESTs.

The first contribution of this work is the use of Rissanen's context tree (Rissanen 1983) to model and compress DNA sequences. In contrast to other universal modeling procedures that are known to have asymptotic convergence, the context tree also has the best non-asymptotic convergence rate (Ziv 2001). A modified version of this model – which is called *Variable Order Markov* model (VOM), (Buhlmann and Wyner, 1999, Ben-Gal et al. 2000) – is applied here to compress a benchmark set of relatively short sequences from the human genome. The VOM algorithm is efficient in terms of its parameterization. Thus, for cases where only a limited amount of data exists, the VOM model "naturally" generalizes to the simple *Probability Weight Matrix* model (Holste et al. 2000). Similarly, if the ideal memory length (which generates the best bias-variance tradeoff) is identical for all data points, then the VOM generalizes to the fixed-order Markov model. In particular, the VOM model is a special generalization of the well-known hexanucleotide bias fifth-order Markov model. The overall flexibility of the model parameterization implies that when incorporating it in genefinders, it is expected to be effective for species-independent discoveries as well. In the rest of the paper we use the term *VOM model* and *context tree* interchangeably.

The second contribution of this work is the use of a compressibility measure to determine whether a given sequence is coding or noncoding. The underlying idea is to construct two VOM models from two given training sets: one VOM model from a training set of coding sequences, and the other VOM model from a training set of noncoding sequences. Then, in the classification stage, the sequence type is determined by the VOM model that obtains a higher compression rate.

The rest of this paper is as follows: in section 2, we describe the context tree model and its use for compression; in section 3, we present some optimization experiments on a benchmark dataset of short DNA sequences. Section 4 concludes this paper with some discussion.

2. Introduction to Universal Compression and VOM Models

For a stationary ergodic sequence, the expected time until the recurrence of a fixed pattern is the inverse of the pattern's probability (Kac 1947). Universal coding is defined as “any asymptotically optimum method of memoryless coding for sources with unknown parameters”. Universal coding algorithms measure the frequency of recurring patterns to construct a model of the system that generated the sequence. The convergence of a universal algorithm means that for long sequences, the model provided by the universal source behaves like the “true” system for all tasks we wish to use the model for, such as coding, prediction, and decision in general. Restricting this work to coding of finite-alphabet discrete sources, and following Ziv (2001), we will introduce a universal coding with non-asymptotic convergence - the proof of convergence does not require an infinite sequence length. Thus, it can be used when the lengths of the sequences are too “short” for the use of asymptotic methods. This attribute is beneficial for adapting the algorithm to new DNA sequence.

Following the notation in Ziv (2001) and Buhlmann and Wyner (1999), let us consider a discrete sequence with $N+1$ symbols, $X_{-N}^{0} \equiv X_{-N}, X_{-N+1}, \ldots, X_{0}$, where each symbol X_i belongs to an alphabet A of cardinality $|A|$, and where the sequence is emitted by a stationary source.

The estimation problem: given X_{-N}^{0}, estimate $P(X_1 \mid X_{-N}^{0})$, the unknown conditional probability distribution of any X_1 given X_{-N}^{0}. To estimate $P(X_1 \mid X_{-N}^{0})$, one assigns an arbitrary conditional probability measure $Q(X_1 \mid X_{-N}^{0})$ for X_1, hoping that it is "close" in some sense to the true $P(X_1 \mid X_{-N}^{0})$ (Ziv 2001).

Consider the class of universal conditional probability measures that count the recurrence of the longest suffix of X_1 in X_{-N}^{0}. The suffix – $X_{-K_o(X_{-N}^{0})}^{0}$ termed also as the *context* – is a *subsequence* of the past sequence X_{-N}^{0}. $K_0(X_{-N}^{0})$ is an integer function of the training sequence X_{-N}^{0}, which represents the depth of the context for

X_1. Ziv (2001) proved that for a class of stationary Markov sources of order *K*, the suffix depth is bounded as $K_0 \le O\left(K^3\right)$.

A simple universal estimation algorithm is described next. First, compute K_0 as a function of X_{-N}^0. Second, evaluate the empirical probability measure as the relative frequency of the appearance of the sub-sequence $\{X_{-N}^0, X_1\}$ over all sub-sequences $\{X_{-N}^0, X_i\}$, $X_i \in A$. Finally, define K_0 as the smallest value of $K_0(X_{-N}^0)$ for which the sub-sequence $\{X_{-N}^0, X_1\}$ appeared at least once in the sequence X_{-N}^0, and practically $Q(X_1 \mid X_{-K_0}^0) \cong Q(X_1 \mid X_{-N}^0)$.

Example: Let, $X_{-6}^0 = ACACGAA$ and suppose we want to estimate the likelihood $P(C \mid X_{-6}^0)$. Note that the sub-sequence *AC* is the longest sub-sequence that start with *A* and it recurs twice in X_{-6}^0. Extending the depth of the context of $X_1 \equiv C$ – the symbol to be predicted – by one more symbol results in the sub-sequence *AAC* that does not appear anywhere within in X_{-6}^0. Thus, for this short sequence, $K_0(X_{-6}^0) = 0$, implying that the context is given by $X_{-K_0\left(X_{-6}^0\right)}^0 = A$. The sub-sequences *AC* and *AA* appear twice and once, respectively, and hence we obtain $Q(C \mid X_{-K_0\left(X_{-6}^0\right)}^0) = Q(C \mid A) = \frac{\#(AC)}{\#(AA)+\#(AC)+\#(AG)+\#(AT)} = \frac{2}{1+2+0+0}$, where $\#(\cdot)$ denotes the frequency of its argument.

For the problem of universal compression, one has to minimize the *Kullback-Leibler* (*KL*) divergence measure $E_P \log \frac{P(X_1 \mid X_{-N}^0)}{Q(X_1 \mid X_{-N}^0)}$, where E_P denotes expectation taken with respect to $P(X_1 \mid X_{-N}^0)$. Ziv (2001) presents non-asymptotic lower bounds to the expected compression rate of *any* universal algorithm that is sequential and has limited training data. Several universal algorithms are proposed in the literature that can achieve these bounds. The advantage of the Context Tree Weighting (CTW) algorithm proposed by Willems et al. (1995) and the similar context-tree algorithm proposed by Ben-Gal et al. (2003) is that they can approach these bounds with *the most efficient learning rate* (defined by Ziv 2001). Practically,

this means that such algorithms can be used for relatively short sequences, as we will show in the next section.

A VOM construction algorithm, is an algorithm that computes from a given *training* sequence the probability estimates for *every context.* The VOM model is used to evaluate the probability of *any testing* sequence to be generated (by the source the generated the training sequence). Figure 1 presents a VOM tree computed from a set of DNA motifs. Branches link two nodes and are labeled by the symbol types. A context is represented by the path of branches starting at the root until it reaches a specific node. The context order is reversed with respect to the order of observance, such that deeper nodes correspond to previously observed symbols. The lengths (depth) of various contexts (branches in the tree) do not need to be equal.

The context-tree algorithm of Ben-Gal et al. (2003) contains two distinct phases: In the tree growing phase, the counts of all the sub-sequences that are shorter than a predefined $K_{\max}$ are used to update the symbol counters in the nodes. In the tree pruning phase, probability estimates are computed for every context and pruning rules keep a descendent node only if the distribution of its symbols is sufficiently different (in *KL* measure) from that of the symbols of its parent node. The distribution of symbols in the nodes of the pruned tree defines the VOM model that is used to estimate $P(X_1 \mid X_{-K_0}^{0})$. The outline of the Context Tree algorithm is as follows:

Tree growing phase:

Step 0. Start with the root as the initial tree, with all symbol counts zero.

Step 1 -counter update: Recursively, having constructed the current tree from the current sequence, read the next symbol X_i in the sequence. Traverse the tree along the path defined by the context X_{-k}^{0} and increment the count of the symbol X_i in that node until its deepest node is reached. *k* is the length of the current context of the current symbol X_i.

Step 2 -tree growth: if the last updated count is at least 1, and the depth $k < K_{\max}$,

create new nodes of depth *k+1* and initialize all symbol counts to zero except for the symbol X_i, whose count is set to 1.

***Tree pruning phase*:**

Keep only the deepest nodes in the tree with depth $K_{\max} \le \log(N+1)/\log(|A|)$ and $\Delta_{leaf}(context) \ge C(|A|+1)\log(N+1)$, where the logarithms are base 2. The driving principle is to prune a descendant node having a distribution of counter values similar to that of the parent node. In particular, calculate $\Delta_{leaf}(context)$ – the (ideal) code-length-difference – as the difference between the entropies of the symbol distributions of parent node and its descendant nodes. *C* is a pruning constant that can be tuned to specific accuracy-size trade-off requirements. A too small *C* may generate a large context-tree that could over-fit the training sequence. The recommended default of $C = 2$ is validated experimentally in the next section. The algorithm is implemented in the MATLAB scripting language setting $K_{\max} = 9$ - the largest value in the growing phase that could fit the context tree in the computer memory (a Pentium II-400 Megahertz computer equipped with a 256 Megabytes memory).

Note that for a Markov chain model, the order is fixed in advance: $K_0 = K_{\max}$. The Markov chain model suffers from exponential growth of the number of parameters to be estimated. For small data-sets this results in over-fitting to the training set and poor variance-bias tradeoff (Buhlmann and Wyner 1999). Optimizing the order or interpolation between several model orders is a difficult process (Ohler et al 1999).

Given a context tree obtained as a result of recurring patterns in the data, compression measures can now be calculated. Each node in the tree is related to a specific recurring context (sub-sequence). Hence, the original sequence can be *uniquely* coded by the sub-sequences in the context tree. Using an arithmetic coder, it is guaranteed that the redundancy – the difference between theoretical and practical

coding – does not exceed two bits per sequence (Willems et al. 1995). For example, consider the context tree in Figure 1. Note that the probability of the string "*GCTTA*", can be calculated by applying the multiplication chain rule and then parsing the sequence into identified contexts (see Figure 1): $P(GCTTA) = P(G)\cdot P(C|G)\cdot P(T|GC)\cdot P(T|GCT)\cdot P(A|GCTT) \cong P(G)\cdot P(C|G)\cdot P(T)\cdot P(T|T)\cdot P(A|CTT)$ $= 0.16\cdot 0.22\cdot 0.39\cdot 0.36\cdot 0.54 = 0.002669$ (respectively, by nodes 1, 3, 1, 4, and 6). Using an arithmetic encoder, the number of bits required to represent this sequence is approximately $-\log_2(0.002669) \cong 8.55$ bits. Simple binary coding of 2 bits per symbol would require 10 bits to code this sequence of length five.

A sequence that *does not belong to the same class* of sequences from which the context tree was generated (trained) *is expected to obtain a lower compression rate* when using the context tree probabilities from the training set. The longer the sequence, the stronger the probability of a lower compression. We further use this phenomenon in the next section.

///* insert figure 1 about here ***///**

Figure 1: The VOM generated from a set of E.Coli promoter motifs. The empirical probabilities in each node are ordered with respect to nucleotides $\{A,C,G,T\}$

Note that Ron et al. (1996), Bejerano and Yona (2001), proposed an algorithm for the detection of Probabilistic Suffix Trees. Their algorithm, which consists of *five* arbitrary parameters, was applied to text clustering, classification of protein families, and classification of E-Coli genome. Further modifications by Apostilico and Bejerano (2000), Slonim et al. (2000) speeds up its learning and prediction time so they are linear with the sequence length. Their algorithm is significantly different than ours in the parameterization, growth, and pruning stages. From theoretical considerations (Ziv 2001), we expect that our algorithm - due to its fast convergence rate - will produce more accurate results when small datasets are involved.

3. Experiments: Distinguishing Coding and Noncoding DNA

Our method to distinguish coding from noncoding DNA involves the calculation of a coding measure, acting as a score function, and a classification scheme that infers a "coding" or "noncoding" identification based on the score value. The score is calculated for a given sequence of fixed length. In the first part of this section, we use a benchmark dataset to optimize the parameters of the algorithm. We demonstrate the superiority of the VOM over the de-facto standard Markov models of fixed-order five (denoted hereafter by "Markov5"). In the second part, we construct an up-to-date VOM model and further optimize it by boosting techniques.

A. Optimizing the parameters of the algorithm

In a classical experiment, Fickett and Tung (1992) generated two datasets of representative coding and non-coding sequences from the human genome.

In coding sequences, the nucleotides operate in triplets (called *codons*). Each codon encodes one amino-acid. It is well known that the distribution of the nucleotides depends on their position in the codon (first, second, or third position). Accordingly, Fickett and Tung (1992) extracted a special subset of the coding dataset in which the first nucleotide takes always the first position in the codon triplet. They called it "phase-coding" and used approximately half of the data for model training and the remaining half for model testing.

In their comparative study of 21 different coding measures, Fickett and Tung (1992) found that homogenous and non-homogenous Markov chains of order 5 yield the highest accuracy in classifying sequences of 54 nucleotides (base pairs). The Markov5 model is exactly the model that would have been obtained had we applied an unpruned (i.e., complete and balanced) VOM model of depth $K_{\max} = 5$. Further details on the use of fixed order Markov models for likelihood estimation can be found in Duda (2001).

In what follows we use the same benchmark data set to compare the accuracy of the VOM classification versus the best of the 21 coding measures evaluated in Fickett and Tung (1992). The first part of Table 1 presents the number of sequences (both coding and noncoding) that were used in Fickett and Tung experiment. For the phased coding sequences, the first nucleotide starts in the first position of the codon. For the non-phased coding sequences, the position of the first nucleotide in the first codon is unknown. Recall that the number of sequences has a potential effect on the size of the truncated context tree, as a result of the pruning algorithm.

Table 1: The number of benchmark sequences of length 54 base pairs (bp)

	#Non-coding Sequences	Coding Sequences		
		#phased	#non-phased	
Fickett and Tung (1992)	125,870	4,199	16,275	Training set
	123,166	4,680	18,238	Testing set
GENIE (1998)	25,333	4,079	4,079	Training set
	25,333	4,079	4,079	Testing set

Following Fickett and Tung (1992), we conducted two experiments. In the first experiment, we constructed two homogenous VOM trees from the training datasets – T_C from coding segments and T_{NC} from noncoding segments. Then, we applied the following rule to classify the unknown sequences:

$$\begin{cases} \text{classify as } \textit{"coding DNA"} & \text{If } \textit{length}(\text{coding by } T_C) < \textit{length}(\text{coding by } T_{NC}) \\ \text{classify as } \textit{"non-coding } \text{DNA}\textit{"} & \text{Otherwise.} \end{cases}$$

In the second experiment, three non-homogeneous VOM trees – one for each position in the codon – were constructed and trained from the phased training set. These VOM models were simultaneously used to score the unknown sequences – the likelihood of each nucleotide was obtained from the respective VOM tree, T_{Ci} $i=1,2,3$ depending on the nucleotide position in the codon. A rule similar to the above was then used to classify unknown sequences, only that now the coding was performed by the combination of VOM models T_{C1}, T_{C2}, T_{C3}.

Note that each non-homogeneous (position-dependent) VOM was constructed from only one third of the coding sequence. The accuracy of the classification was computed as the average of the correct classification ratios on the true coding and the true non-coding testing subsets (i.e., using bioinformatics terminology, as the average of the *true positive* and the *true negative* values). The 95% confidence interval for the accuracy mean of the testing set was estimated to be approximately[2] $\pm 0.7\%$.

Table 2 presents the comparative results of both types of experiments (the accuracy is computed from the testing sets). The first row presents the classification accuracy– based on the Markov5 as the best model found in Fickett and Tung (1992) – for both the non-phased and the phased sequences. The following six rows present the classification accuracy based on different VOM models with various truncation coefficients, in the range $C \in [0.5,...,8]$. The last two columns present the number of parameters of each model (the sum of the number of independent contexts in the non-coding VOM tree and the coding VOM tree(s)). The VOM8 and VOM9 trees are those VOM models generated with $K_{\max} = 8,9$ respectively. Note that the number of parameters for the Markov5 model is smaller than the theoretical number ($4^{5+1} - 1 =$ 4,095 parameter for each Markov5 model) since some contexts with specific biological functions (such as contexts that include the "stop" codons) are excluded from the learning dataset.

From the top part of Table 2, we find that the top four VOM models provide a more accurate classification in comparison to the 21 coding measures evaluated in Fickett and Tung (1992) (for sequences of length 54 bp). In fact, all the VOM models obtain for the non-phased data an accuracy level, which is above the confidence upper-bound 70.5 + 0.7 = 71.2%. For the phased data, the VOM models obtain better accuracy only for $C \le 2$. Note the inverse relation between the truncation coefficient *C* and the number of parameters in the VOM - the accuracy seems to increase with the logarithm of the number of parameters, which is inversely related to *C*. A reasonable

[2]Using the normal approximation to the binomial, with the worst case scenario from tables 1,2
$1.96 \cdot 0.5\sqrt{\frac{p(1-p)}{n_1} + \frac{p(1-p)}{n_2}}$, $p = 0.705$, $n_1 = 4079, n_2 = 25333$

compromise between the obtained accuracy and the number of parameters (or, in fact, between the model bias and the model variance) seems to be found at *C=2*. Further decreasing *C* results in a moderate gain in the accuracy for an excessive number of parameters. The smaller numbers of the VOM model parameters indicate that it is less vulnerable to bias errors, and hence, can be better estimated from smaller data sets. Reducing $K_{\max}$ from 9 to 8 had a marginal effect on the obtained accuracy while reducing the number of parameters by approximately 25%. This indicates the potential for further optimization of the number of parameters, yet, further reduction of $K_{\max}$ may result in loosing the capability to identify long contexts of special interest.

Table 2: comparative results with respect to the benchmark sequences of length 54 bp.

Experiment		Accuracy (test set)		# of parameters	
		non-phased	Phased	non-phased	phased
Fickett and Tung (1992)	Markov5	70.5%	80.7%	7,651	15,149
	VOM9, *C=0.5*	72.8%	82.3%	26,467	21,352
	VOM9, *C=1*	72.5%	82.5%	12,561	10,677
	VOM8, *C=2*	71.9%	82.3%	5,153	5,022
	VOM9, *C=2*	72.1%	82.4%	6,894	6,703
	VOM9, *C=4*	71.7%	77.8%	4,417	4,273
	VOM9, *C=8*	71.4%	76.8%	2,314	2,125
GENIE (1998)	Markov5	82.0%	86.3%	7,651	15,149
	VOM8, C=2	81.2%	86.1%	2,542	2,563
	VOM9, C=2	81.2%	86.2%	2,582	2,603
	Boosted VOM8	83.8%	88.9%	2,549	2,706

B. Boosting the VOM model

Boosting (Freund and Schapira 1997) is a general machine-learning technique that mixes predictions from different models, where each model is constructed from a different population of samples. It is often used to improve the accuracy of weak classifiers (having large error probability). In the context of VOM models, weighting the contexts of various VOMs is effectively equivalent to generating a new VOM model with an equivalent distribution of contexts. In the following we used boosting to (indirectly) optimize the structure of the VOM.

Several boosting heuristics are available in the literature. The common theme is that a special population rich with "difficult cases" (the boosted population) is prepared with the implicit assumption that a classifier that is trained on the difficult population will obtain better results on the "normal" population of samples. The size of the VOM model (thus its accuracy) increases non-linearily with the size of the training set. To neutralized this effect, we constrained the size of the boosted training set such that it is approximately equal to the size of the original training set. The following heuristics was used to generate the boosted population:

- Construct the VOM classifier using the original training set.
- Use the VOM classifier to identify the true-classified sequences and the miss-classified sequences.
- Duplicate the miss-classified sequences in the boosted set, while fixing a 30% probability for each true-classified sequence to be excluded from the training set.

Boosting algorithms are sensitive to the effect of outliers in the training set (since their number is also boosted). The data-set of Fickett and Tung (1992) is now days considered outdated and of low quality for the purpose of representing the human genome. We therefore used the more updated GENIE data-set (GENIE, 1998) for the construction of an up-to-date VOM model. The GENIE data-set contains 462 coding sequences and 2381 introns that are representative sequences of the human genome with less than 80% homology between sequences. Following the experiment of Fickett and Tung (1992), the sequences where chopped into segments of size 54 bp. The bottom part of Table 1 details the sizes of the training and the testing sets used in the experiments. A VOM classifier was constructed from the training set and was then used with the above-mentioned algorithm to generate the boosted dataset.

The new VOM classifier was constructed from the boosted dataset and used to re-score the testing dataset. For comparison purpose, we also present the results of the Markov5 model as well as the results obtained from the un-boosted VOM8 C=2 model. The bottom part of Table 2 presents the comparative results of the boosting

experiment. As can be seen, the boosted VOM classifier produced a statistically significant 2.6% improved accuracy over the un-boosted VOM classifier, while retaining a similar number of parameters. Though the accuracy of the Markov5 model is marginally higher than the accuracy produced by the VOM model, it has a significantly higher number of parameters. Thus, we did not pursue with boosting the Markov5 model.

The new and reliable dataset improved the overall classification performance of the tested models by 4% – 10%. Since a single iteration of the boosting algorithm was sufficient to produce a significant improvement in the classification accuracy, we believe that implementing the complete boosting algorithm can further improve the result. We did not pursue further this issue because the results were sufficient to prove the superiority of the VOM classifier, and sufficient accuracy was obtained for the construction of the sequence annotation device presented next.

Finally, let us note that typically the updated "clean" human datasets have huge sizes (containing more than 50,000 sequences). This situation is a-typical to other organisms that are of interest in Bioinformatics. The VOM models – due to their efficient parameterization – are expected to outperform other algorithms when data is scarce, or when the quality of the sequence annotation is poor.

4. Discussion

Current gene-recognition approaches are exceedingly multifaceted, implementing a variety of well-established algorithms. Recognition of coding DNA regions is an important phase of any gene-finder procedure. We believe that there exist niche datasets with specific characteristics, which are not entirely addressed by conventionally used algorithms. For example, datasets from newly sequenced genomes that share little homology with known datasets. Often, in such cases, it is difficult to tune properly the gene-finders due to over-parameterization.

In this paper, we introduce the VOM-based compression method to coding recognition. The VOM model was originally introduced in the field of information theory for compression purposes and later has been implemented successfully in various research areas, such as statistical process control (Ben-Gal et al. 2003) and analysis of financial series (Shmilovici et al. 2003). The unique feature of the proposed method (compared to closely related ones, such as the Probabilistic Suffix Tree by Bejerano and Yona, 2001) is that its convergence to the optimal parameter set was proven to be optimally fast even for *finite* sequences (Ziv 2001). Practically, such convergence ensures an efficient parameterization of the VOM model, even when it is trained on small and relatively unreliable datasets.

In our experimentations for distinguishing coding and noncoding DNA, the proposed VOM-based approach outperforms any of the 21 traditional methods evaluated in Fickett and Tung (1992).

The initial encouraging results makes it tempting to conjecture that elements of the proposed method could be integrated into other gene-finding procedures. Such integration might enhance the procedure performance on short coding fragments from relatively low-quality sequenced data.

References

Apostolico, A., G. Bejerano. 2000. Optimal amnesic probabilistic automata or how to learn and classify proteins in linear time and space. J. Comp. Biology, 7(3/4) 381-343.

Bejerano, G., G. Yona. 1999. Variations on Probabilistic Suffix Trees: statistical modeling and prediction of protein families. *Bioinformatics*. 17(1) 23-43.

Ben-Gal, I., A. Shmilovici G. Morag. 2000. Design of control and monitoring rules for state dependent processes. *The International Journal for Manufacturing Science and Production*, 3(2-4) 85-93.

Ben-Gal, I., A. Shmilovici G. Morag. 2003. CSPC: A Monitoring Procedure for State Dependent Processes. *Technometrics*, 45(4) 293-311.

Brown, N.P., C. Sander et al. 1998. Frame: detection of genomic sequencing errors. *Bioinformatics* 14(4) 367-71.

Buhlmann, P., A. J. Wyner. 1999. Variable Length Markov Chains. *Ann. Statist.* 27(2) 480-513.

Burge, C., S. Karlin. 1998. Finding the genes in genomic DNA. *Current Opinion in Structural Biology*. 8(3) 346-54.

Chen, X., S. Kwong, M. Li. 2000. A compression algorithm for DNA sequences and its application in genome comparison. Proceedings of the fourth annual international Conference on Computational Molecular Biology - RECOMB 2000. ACM Press. 107-117.

Duda, R.O., P.E. Hart D.G. Stork. 2001. *Pattern Classification*. John Wiley & sons.

Fickett, J.W., C.S. Tung. 1992. Assessment of protein coding measures. *Nucleic Acids Research,* 20(24) 6441-50.

Fickett, J.W., 1996. Finding genes by computer: the state of the art. *Trends in Genetics*. 12(8) 316-20.

Freund, Y., R.E. Schapira, 1997. A Decision Theoretic Generalization of on-line learning and an application to Boosting, *Journal of Computer and Systems Sciences*, 55(1), 119-139.

GENIE data-sets, from Genbank version 105, 1998. Available: www.fruitfly.org/seq_tools/datasets/Human/CDS_v105/ ; www.fruitfly.org/seq_tools/datasets/Human/intron_v105/

Grumbach, S., F. Tahi. 1994. A new challenge for compression algorithms: genetic sequences. *J. of Information Processing and Management*, 30(6) 866-875.

Holste D., I. Grosse, S. V. Buldyrev, H. E. Stanley, H. Herzel. 2000. Optimization of Protein Coding Measures Using Positional Dependence of Nucleotide Frequencies. *J. of Theoretical Biology*. 206, 525-537.

Iseli, C., C. V. Jongeneel, P. Bucher 1999. ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. Proceedings of Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA.

Kac K., 1947. On the notion of recurrence in discrete stochastic processes. *Bulletine of the* American Mathematical Society. 53 1002-1010.

Liu, X., D. L., Brutlag, J. S., Liu. 2001. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Pac. Symp. Biocomput., 6, 127-138.

Ohler U., S. Harbeck, H. Niemann, E. Noth, M. Reese. 1999. Interpolated Markov chains for eukaryotic promoter recognition. Bioinformatics. 15(5) 362-369.

Rissanen, J., 1983. A universal data compression system. IEEE Transactions on Information Theory, 29(5) 656- 664.

Rivals, E., O. Delgrange, J.P. Delahaye, M. Dauchet, M.O. Delorme, et al. 1997. Detection of significant patterns by compression algorithms: the case of approximate tandem repeats in DNA sequences. CABIOS, 13(2) 1313-136.

Ron, D., Y. Singer, N. Tishby. 1996. The power of amnesia: learning probabilistic automata with variable memory length. *Machine Learning*. 25:117-149.

Shmilovici, A., Y. Alon-Brimer, S. Hauser. 2003. Using a Stochastic Complexity Measure to Check the Efficient Market Hypothesis. *Computational Economics*, 22(3) 273-284.

Slonim, N., S. Fine N. Tishby. 2000. Discriminative Variable Memory Markov Model for Feature Selection. Available online: citeseer.nj.nec.com/484303.html

Willems, F.M.J., Y.M. Shtarkov T.J.Tjalkens. 1995. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*. 41(3) 653-664.

Ziv, J., 2001. A universal prediction lemma and applications to universal data compression and prediction. *IEEE Transactions on Information Theory* 47(4) 1528-1532.

Ziv, J., A. Lempel. 1997. A universal algorithm for sequencial data compression. *IEEE Transactions on Information Theory*, 23(3) 337-343.

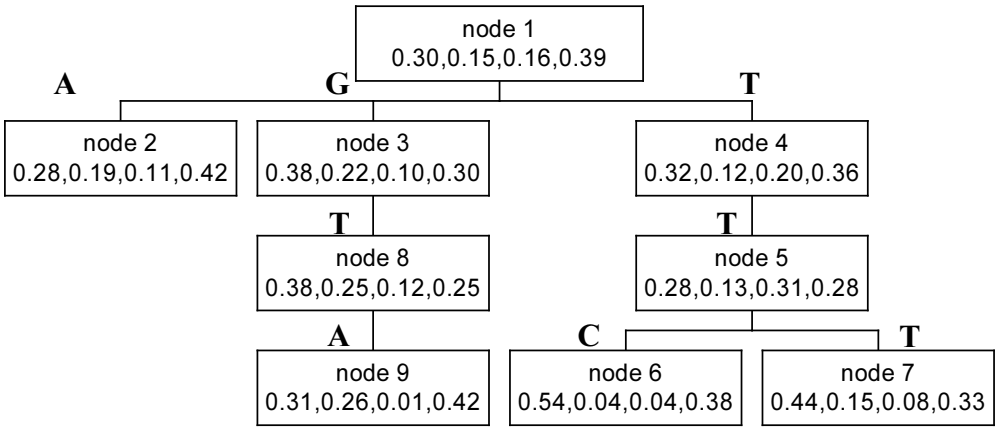

Figure 1: The VOM generated from a set of E.Coli promoter motifs. The empirical probabilities in each node are ordered with respect to nucleotides $\{A,C,G,T\}$