DETECTING SIMPLE CLUSTER STRUCTURE OF TRIPLET DISTRIBUTIONS IN GENETIC TEXTS

Alexander N. Gorban[2,3], Andrei Yu. Zinovyev[1], Tatyana G. Popova[2]

[1]*Institut des Hautes Études Scientifiques, Bures-sur-Yvette, France*

[2]*Institute of Computational Modeling, Russian Academy of Science*

[3]*Institute of polymer physics, ETH, Switzerland*

Abstract

Motivation: In several recent papers new algorithms were proposed for detecting coding regions without requiring learning dataset of already known genes. In this paper we interpret some of these results and propose a simpler method.

Results: Several complete genomic sequences were analyzed, using visualization of tables of triplet counts in a sliding window. The distribution of 64-dimensional vectors of triplet frequencies displays a well-detectable cluster structure. The structure was found to consist of seven clusters, corresponding to protein-coding information in three possible phases in one of the two complementary strands and in the non-coding regions. Awareness of the existence of this structure allows development of methods for the segmentation of sequences into regions with the same coding phase and non-coding regions. This method may be completely unsupervised or use some external information. Since the method does not need extraction of ORFs, it can be applied even for unassembled genomes. Accuracy calculated on the base-pair level (both sensitivity and specificity) exceeds 90%. This is not worse as compared to such methods as HMM, however, has the advantage to be much simpler.

Availability: The software and datasets are available at http://www.ihes.fr/~zinovyev/bullet

Contact: zinovyev@ihes.fr

Supplementary information: http://www.ihes.fr/~zinovyev/bullet

Introduction

With few exceptions, almost all commonly used gene-finding programs employ a learning dataset for tuning the parameters of the learning rule. In several recent papers new algorithms were proposed for detecting coding regions without requiring learning dataset of already known genes. In Bernaola (2000) the authors proposed a method developed for unsupervised segmentation of whole DNA texts, which corresponds to the segmentation for coding and non-coding regions. In Audic and Claverie (1998) the authors proposed to use clustering procedure that uses all available annotated genomic data for its calibration and is not based on direct pairwise comparisons. The iterative procedure uses genomic sequences to adjust parameters (that were initialized by randomly partitioning a number of small subsequences) of probabilistic sequence models. The algorithm converges fast and gives accuracy up to 90%. In Baldi (2000) it was explained that this algorithm is

essentially a form of the expectation maximization algorithm applied to the corresponding probabilistic mixture model.

In this paper we introduce a similar but much simpler method, which does not refer to the probabilistic methods of sequence analysis. Instead, we use a method of data visualization to explore the space of frequencies of triplet counts in a sliding window and to demonstrate the structure of a dataset used for learning. We show that in the case of high concentration of coding bases (microbial genomes, yeast genomes), traditional use of a learning dataset (a set of examples of already known coding and non-coding regions) may be replaced by an unsupervised procedure. Then we propose a simple clustering method for detecting coding regions in the whole genomes and test it's performance that turns out to be essentially the same as of the methods mentioned above, as well as of new traditional microbial gene-finders (like GLIMMER [Salzberg et al., 1998, Delcher et al., 1999]).

Let us denote codon frequency distribution by f_{ijk}, where $i,j,k \in \{A,C,G,T\}$, *i.e.*, for example, f_{ACG} is equal to the frequency of the ACG codon in a given coding region. One can introduce such natural operations over the frequency distribution as *phase shifts* $P^{(1)}$, $P^{(2)}$ and *complementary reversion* C^R:

$$P^{(1)} f_{ijk} \equiv \sum_{l,m,n} f_{lij} f_{kmn}, \quad P^{(2)} f_{ijk} \equiv \sum_{l,m,n} f_{lmi} f_{jkn} \quad \hat{f}_{ijk} \equiv C^R f_{ijk} \equiv f_{\hat{k}\hat{j}\hat{i}},$$

where $\hat{i}$ is complementary to *i, i.e.,* $\hat{A} = T$, $\hat{C} = G$, etc.

The phase-shift operator $P^{(n)}$ calculates the new triplet distribution, but now counted with a frame-shift on *n* positions, in the hypothesis that no correlations exist in codon order. Complementary reversion constructs the distribution of codons from a coding region in the complementary strand, but counted in the forward strand ("shadow" codon usage).

Let us introduce the distance between two distributions as $\left\| f_{ijk} - g_{ijk} \right\| = \sum_{ijk} \left| f_{ijk} - g_{ijk} \right|$.

It is then natural to expect that the problem of gene recognition may be solved if one of the numbers, $\left\| f_{ijk} - P^{(1)} f_{ijk} \right\|, \left\| f_{ijk} - P^{(2)} f_{ijk} \right\|$ is large enough. It follows from that remark that after a large number of insertion and deletion operations of one base-pair at a time, we would have

$$\left\| f_{ijk} - P^{(1)} f_{ijk} \right\| \approx 0, \left\| f_{ijk} - P^{(2)} f_{ijk} \right\| \approx 0.$$

Let us introduce a measure of how far f_{ijk} is from the shifted distributions:

$$CP = \max\left(\left\| f_{ijk} - P^{(1)} f_{ijk} \right\|, \left\| f_{ijk} - P^{(2)} f_{ijk} \right\| \right)$$

Real distributions in the first and second phases (where correlations are taken into account) will be denoted as $f^{(1)}_{ijk}$, $f^{(2)}_{ijk}$, $\hat{f}^{(1)}_{ijk}$, $\hat{f}^{(2)}_{ijk}$. Let us introduce the term "*codon correlation contribution measure"* as the average distance between real and calculated distributions $CC = \frac{1}{2}\left(\left\| P^{(1)} f_{ijk} - f^{(1)}_{ijk} \right\| + \left\| P^{(2)} f_{ijk} - f^{(2)}_{ijk} \right\| \right)$.

Methods

We have constructed datasets of triplet frequencies for several real genomes and for several model genetic sequences, as follows:

1) Only the forward strands of genomes are used for triplet counting;

2) Every p positions in the sequence, we open a window (x-$W/2$,x+$W/2$,) of size W and centered at position x;

3) Every window, starting from the first base-pair, is divided into $W/3$ non-overlapping triplets, and the frequencies of all triplets f_{ijk} are calculated;

4) The dataset consists of $N = [L/p]$ points, where L is the entire length of the sequence. Every data point $X_i = \{x_{is}\}$ corresponds to one window and has 64 coordinates, corresponding to the frequencies of all possible triplets $s = 1,\ldots,64$.

A standard centering and normalization on a unit dispersion procedure is then applied, *i.e.*, $\tilde{x}_{is} = \frac{x_{is} - m_s}{\sigma_s}$, where $\tilde{x}_{is}$ is the value of the sth coordinate of the ith point after normalization, and m_s is the mean value of the sth coordinate, and σ_s is the standard deviation of the sth coordinate.

Then we applied the principal components algorithm in order to visualize a 64-dimension dataset on a 3-dimensional linear manifold spanned by the first three principal vectors of the distribution. It is known that projection onto this manifold is only as informative as the higher value of $\nu^{(3)} = D^{(3)}/D$, where D is the dispersion of the dataset, calculated in 64-dimensional data-space and $D^{(3)}$ is the analogous quantity calculated after projecting the vectors in 3-dimensional space. In practice, even if the value of $v^{(3)}$ is not high enough (say, it equals 0.1-0.3), we may still try to visualize the dataset, in the hope of being able to pick up qualitative "signals" of the presence of hidden patterns in the data distribution, as well as to visually represent the dataset.

Results

Figure 1 presents several distributions calculated for real genetic texts. It is clear that the distribution consists of seven clusters. In some cases these clusters are situated quite symmetrically, in others they are not. In addition to the distribution itself, we introduced two triangles, formed by the points $f_{ijk}, P^{(1)}f_{ijk}, P^{(2)}f_{ijk}$ and $\hat{f}_{ijk}, P^{(1)}\hat{f}_{ijk}, P^{(2)}\hat{f}_{ijk}$, into the figures. The large spheres correspond to the points f_{ijk} and $\hat{f}_{ijk}$, where f_{ijk} was calculated from the genome's known annotation. Data-points have different shapes and colors, according to whether they are coding or non-coding in one of the two strands. An explanation of the structure is rather clear: Coding information from windows in the forward strand has one of three possible phase shifts. Since this phase shift is not known in advance, approximately one-third of the windows fall into the vicinity of the point that corresponds to the f_{ijk} (0-shift), one-third are close to the $f^{(1)}_{ijk}$ (1-shift), and the last third are close to the $f^{(2)}_{ijk}$ (2-shift). This is also true for the complementary strand, but with the centers corresponding to complementary distributions.

One can see from the pictures that the centers of phase-shifted distributions are close enough to the calculated points, showing an absence of significant correlations in the order of codons. Indeed, the calculated values of *CC* are not high (see Table 1, ***CC*** column.) This means that in real texts correlation between subsequent codons is much less then the inter-phase difference.

Clusterization

Using visual representation of data-point distribution, it is possible to propose a rather natural way of segmenting sequences into regions that are homogeneous with respect to coding phase. One would expect that regions with the same coding phase correspond to protein-coding regions. This procedure was accomplished using the well-known K-means clustering algorithm. After clustering the distribution into seven clusters (the clustering was accomplished in the 64-dimensional space), triplet distribution may be calculated in the (x-W/2,x+W/2) window for every base-pair in position x, and after appropriate normalization, the closest cluster in the data space may be found. If it is the central cluster, that point is likely to be non-coding; otherwise the presence of coding information should be suspected in one of three possible phases.

To evaluate the ability of this procedure to differentiate between "coding" and "non-coding" base-pairs, we used base-level sensitivity and specificity of exon recognition, the measures which are commonly used in this case:

$$Sn = \frac{TP}{TP + FN}, Sp = \frac{TP}{TP + FP}$$

where TP is the number of true-positives, *i.e.,* coding bases predicted to be coding;

TN is the number of true-negatives, *i.e.,* non-coding bases predicted to be non-coding; FP is the number of false-positives, *i.e.,* non-coding bases predicted to be coding, and FN is the number of false-negatives, *i.e.,* coding bases predicted to be non-coding.

We must underline that the procedure is fully automated and does not require any human intervention. Visualization of datasets can be useful to evaluate how reliable prediction will be (compactness of the clusters, for example) and to compare prediction with known information.

The results are shown in the $\mathbf{Sn_1}$ and $\mathbf{Sp_1}$ columns of Table 1. These values are quite high. The only parameter – window size – may be visually evaluated by comparing pictures of data constructed with various values of W (see full version of the paper on the accompanying web-page.) In fact, the dependence of effectiveness on window-size is not strong over a rather long interval of W.

Using known data

In the previous section the learning process used no information other than the sequence itself; it was completely "unsupervised". Of course, one could try to make use of some previous knowledge, as discussed in the next paragraph.

Studying a set of training examples (for example, following the strategy of GLIMMER, using long ORFs as a training set), it is possible to explicitly calculate the centers of all seven clusters. We have done this, using annotation of the analyzed genomes. First, half of the genes were used to calculate the centers, and the rest for accuracy testing. Using these seven vectors as centroids, we calculated new values for the sensitivity and specificity of gene recognition. They are shown in the $\mathbf{Sn_2}$ and $\mathbf{Sp_2}$ columns of Table 1.

Comparing with GLIMMER gene-finder

To compare the results obtained by our algorithm with some well-established gene-finding program, we introduced new simple rules for deciding if a given ORF is coding or non-coding. For every ORF, we calculate 64-dimensional vector of it's codon frequencies and find the closest centroid in the codon frequencies space (the positions of the centroids are calculated as it was described earlier). If the closest centroid is the one, which corresponds to the correct coding phase (let us denote it by P0), then this ORF is suspected to be coding. Then from all such ORFs in P0 phase we filter out all ORFs that are too distant from the P0-centroid (the threshold is determined by an additional parameter), and all ORFs which are inside other ORFs in the P0 phase (it means that we take the longest ORF in the P0 phase).

To test this procedure, we analyzed output of GLIMMER gene-finder (using default settings), using the list of ORFs, produced by GLIMMER. Thus, we compare only effectiveness of measures used, and not the details of ORF list extraction procedures.

In the table 2 we show the results of this comparison, using existing annotations of the genomes in GenBank. One must understand that the annotations are far from being perfect and some part of the ORFs that we denoted as false positives in the GLIMMER prediction can be unknown putative genes (as it is claimed by the authors of GLIMMER). Nevertheless, we find significantly lower false-positive rate of our method comparing to the GLIMMER prediction. Analyzing this, in some genomes we found that a cluster structure exists in the distribution of false-positive GLIMMER predictions. On fig.2 visualization of GLIMMER predictions on the principal components plane is shown for *Escherichia coli* and *Caulobacter crescentus* for which GLIMMER produces many predictions of "additional genes". For example, our analysis shows that 62% of false-positives predictions for *Escherichia coli* and 80% of false positives for *Caulobacter crescentus* in the 64-dimensional triplet frequencies space are closer to the centroid, which corresponds to the $C^R f_{ijk}$ distribution (C0-centroid), while only 2% of true-positive predictions for *Caulobacter crescentus* are close to the C0-centroid. It seems that such discrepancy cannot be explained simply by the "presence of unknown genes" but it is due to some "overfitting" effect of this HMM-based predictor, which often takes "shadow" genes as positive predictions.

As one can see from table 2, the sensitivity of our method is lower in all cases, comparing to the GLIMMER gene-finder. Using annotation of *E.Coli*, we found that from 228 genes predicted by GLIMMER, and not predicted by our method, 121 are annotated as predicted only by computational methods, 11 ribosomal genes and 12 transposases. From 24 genes predicted by our method and not predicted by

GLIMMER, 17 are annotated as predicted only by computational methods and 3 as ribosomal genes. It is not surprising; since it is known that ribosomal genes, some other highly expressed genes as well as horizontally transferred genes (the percentage of which is estimated as 10% from the overall number, [Medigue, 1991]) can have different (with respect to the average) codon usage, for example, strongly translationally biased codon usage in the case of ribosomes. It is known also, that preliminary clusterization of genes can enhance existing gene-finding procedures [Mathe et al., 1999,2000].

Window-size dependence

Figure 3 presents our study of window-size dependence of the algorithm effectiveness for two genomes. One can see that the minimal window length, which can be used for the algorithm, is about 100 bp. This value is often characterized as a barrier for all gene-prediction methods based on the analysis of compositional differences. Then, the sensitivity of the algorithm drops monotonically, and, after window size of 400-500 bp, becomes poor.

Implementation

All datasets were prepared from sequences in the GenBank flat-file format. The programs used for data analysis, including simple implementation of the K-means clusterization algorithm, were written in Java and are available with instructions at the accompanying web page: http://www.ihes.fr/~zinovyev/bullet/. These programs actively use the BioJava programming package. Technically, the data visualization and all illustrations were produced using the ViDaExpert data visualization tool under Windows, and are available at the supplementary web-page.

Discussion

In prokaryotes (for example, *Helicobacter pylori*) the model has approximately the same performance as GLIMMER gene-finder (Salzberg, *et al.*, 1998, Delcher *et al*., 1999), having slightly worse sensitivity, but significantly better specificity. This means that the essential part of the information needed to discriminate between coding and non-coding regions is already contained in triplet distributions Using hexamer frequencies (that is common practice in modern gene-finders) can be more sensitive, but also can lead to some undesirable effects. One needs more sequence information to evaluate hexamers frequencies, and, as a result, this fact can lead to the "overfitting" effects, leading to worse specificity. We demonstrated this fact, using visual analysis of positive predictions of GLIMMER gene-finder.

It is clear from the constructed representations of datasets that the spatial structure of triplet distributions is almost completely determined by two factors: 1) the frequency distribution of the 64 codons in the coding phase; 2) the dispersion of codon frequency distribution. From the figures, it is evident that the distribution structure renders linear discrimination analysis (sometimes applied in this situation) absolutely inapplicable. Applying linear methods in this case would lead to the incorrect

conclusion that the dataset is not well-separable and that this measure is less effective than others with respect to linear discrimination function. For example, in the case of *Helicobacter pylori*, linear discrimination yields a specificity of ≈0.83 (which means many false positives), while the method we proposed yields ≈0.97. This fact stresses that understanding the spatial structure of a learning dataset is absolutely necessary for the reasonable application of pattern recognition methods.

Frequency normalization plays a key role in cluster structure formation. It indicates the important role in distinguishing coding and non-coding regions played by triplets which may not have high frequency values but that considerably change their frequency after a coding phase-shift (codons that are "prohibited," due to bias.)

From the general point of view, distribution of non-overlapping triplets that is efficient for gene recognition corresponds to a high value of mutual information in three consecutive letters, *i.e.,*

$$M = \sum_{ijk} f_{ijk} \log_2 \frac{f_{ijk}}{p_i^1 p_j^2 p_k^3},$$

where p_i^k is the average frequency of letter $i \in \{A,C,G,T\}$ at the *k*th place in triplet. This value may be zero only in the case $f_{ijk} = p_i^1 p_j^2 p_k^3$. In this case, we would have $P^{(1)} f_{ijk} = P^{(2)} f_{ijk} = f_{ijk}$, *i.e.,* phase-shift does not change the codon distribution. High values of *M* guarantee the presence of a "three-phase triangle" in the data space, as well as the formation of a cluster structure.

In this paper, using visual analysis of a spatial dataset structure and simple clustering technique, we have shown that a learning dataset is not necessary in order to accurately solve gene recognition tasks, at least in DNA segments with high concentrations of coding information (compact genomes). This property of the method we propose seems to be very useful, since the problem of choosing a "good" learning dataset is not very well defined (see, for example, [Mathe, Sagot et al.]).

The method proposed can be applied for the rough annotation of unassembled genomes, since it does not require preliminary extraction of ORFs. This makes it useful for inexpensive genome survey projects.

Acknowledgements

The authors thank Alessandra Carbone (IHES, France) for very fruitful discussions, Misha Gromov for the interest he expressed in this work, Noah Hardy and Arndt Benecke for editing the manuscript.

References

Audic S., Claverie J.-M. (1998) Self-identification of protein-coding regions in microbial genomes. *Proc.Natl.Acad.Sci.USA,* **95**.

Baldi, P. (2000) On the convergence of a clustering algorithm for protein-coding regions in microbial genomes. *Bioinformatics*, **16**. 367-371.

Bernaola-Galvan, P., Grosse, I., Carpena, P., Oliver, J.L., Roman-Roldan, R., Stanley, H.E. (2000). Finding borders between coding and noncoding DNA regions by an entropic segmentation method. *Physical Review Letters* **85**(6): 1342-1345.

Besemer, J., Lomsadze, A., Borodovsky M. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.*, **29**, No. 12. 2607-2618.

Burge, C.B., Karlin, S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.* **8**: 346–354.

Fickett, J.W. (1996) The gene identification problem: an overview for developers. *Computer & Chemistry* **20**: 103-118.

Gorban, A.N., Zinovyev, A.Yu., Popova, T.G. 2001. Statistical approaches to the automated gene identification without teacher. *Institut des Hautes Etudes Scientiques preprint*. *IHES, France.* Web-site link: http://www.ihes.fr/PREPRINTS/M01/M01-34.ps.gz.

Salzberg, S.L., Delcher, A.L., Kasif, S., White, O. (1998) Microbial gene identification using interpolated Markov Models. *Nucleic Acids Research* **26**(2): 544-548.

Delcher A.L, Harmon D., Kasif S., White O., Salzberg, S.L. (1999) Improved microbial gene identification with GLIMMER. *Nucleic Acids Research* **27**(23): 4636-4641.

Mathe C, Peresetsky A, Dehais P, Van Montagu M, Rouze P. (1999) Classification of Arabidopsis thaliana gene sequences: clustering of coding sequences into two groups according to codon usage improves gene prediction. J Mol Biol. **285**(5):1977-1991.

Mathe C, Dehais P, Pavy N, Rombauts S, Van Montagu M, Rouze P. (2000) Gene prediction and gene classes in Arabidopsis thaliana. *J. Biotechnol.* **78**(3):293-299.

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

Medigue C., Rouxel T., Vigier P., Henault A., Danchin A. (1991) Evidence for horizontal gene transfer in Escherichia coli speciation. *J. Mol. Biol.* **222**, 851-856.

Table 1

Summary table of results for assessing the method on the nucleotide level

Sequence	L	W	p	$\nu^{(3)}$	% of coding bases	CP	CC	Sn_1	Sp_1	Sn_2	Sp_2
Helicobacter pylori	1643831	300	120	0.35	90	0.68	0.28	0.93	0.97	0.93	0.98
Caulobacter crescentus	4016947	300	300	0.21	91	1.07	0.16	0.93	0.97	0.94	0.98
Prototheca wickerhamii	55328	120	18	0.17	49	0.83	0.11	0.82	0.93	0.84	0.95
Saccharomyces cerevisiae chromosome III	316613	399	99	0.16	69	0.45	0.10	0.90	0.88	0.90	0.90
Saccharomyces cerevisiae chromosome IV	1531929	399	120	0.15	73	0.48	0.09	0.89	0.91	0.92	0.92

Table 2

Comparing the method with GLIMMER gene-predictor

Sequence	CLUSTER		GLIMMER	
	Sn	Sp	Sn	Sp
Helicobacter pylori	0.94	0.95	0.96	0.78
Haemophilus influenza	0.93	0.88	0.96	0.84
Escherichia coli	0.91	0.87	0.96	0.76
Bacillus subtilis	0.89	0.95	0.97	0.79
Caulobacter crescentus	0.89	0.76	0.94	0.60

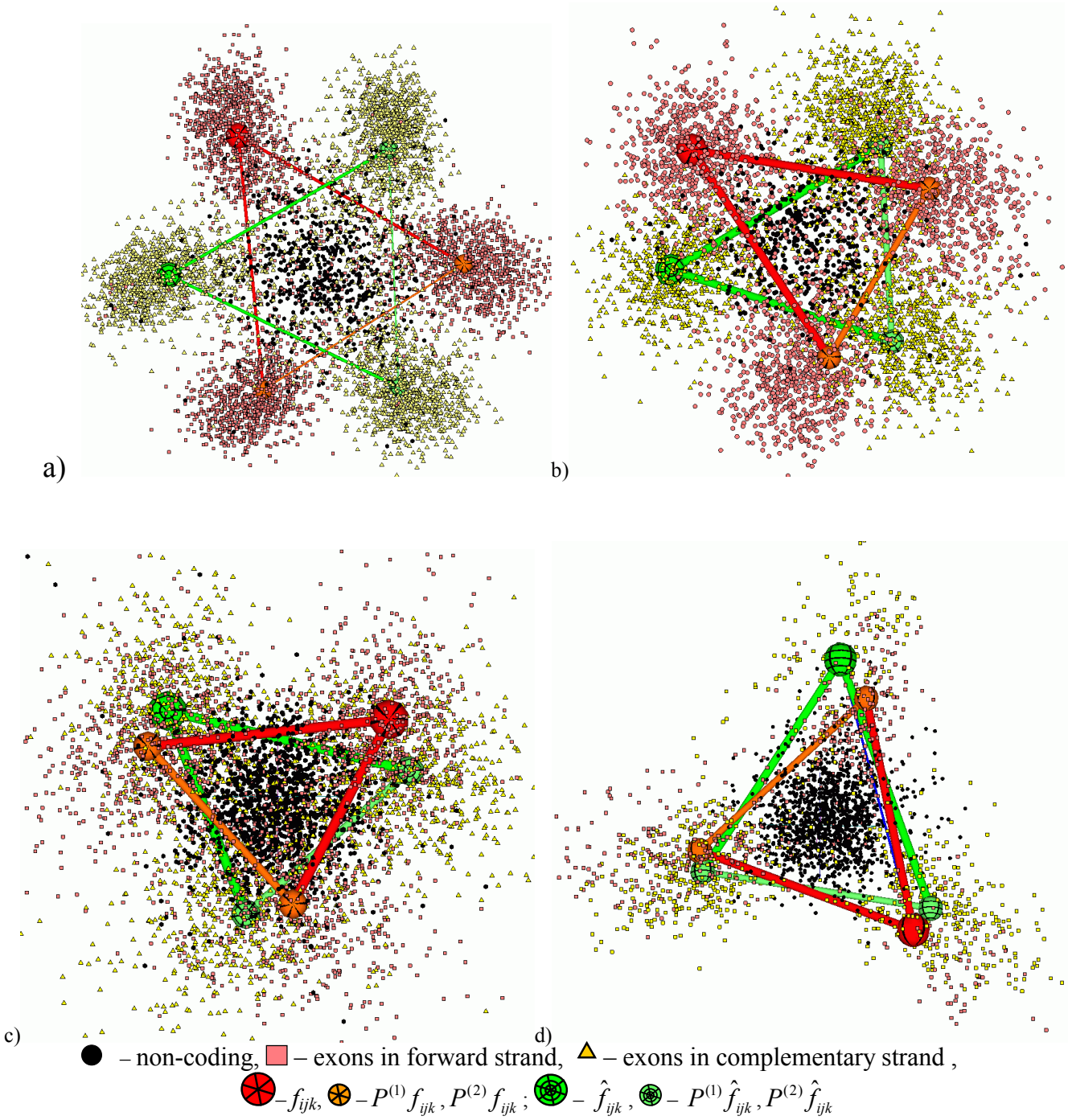

Fig.1. Visualization of genetic sequences in the space of triplet frequencies
a) *Caulobacter crescentus* (GenBank NC_002696);
b) *Helicobacter pylori* (GenBank NC_000921);
c) *Saccharomyces cerevisiae* chromosome IV (GenBank NC_001136);
d) *Prototheca wickerhamii* mitochondrion (GenBank NC_001613).

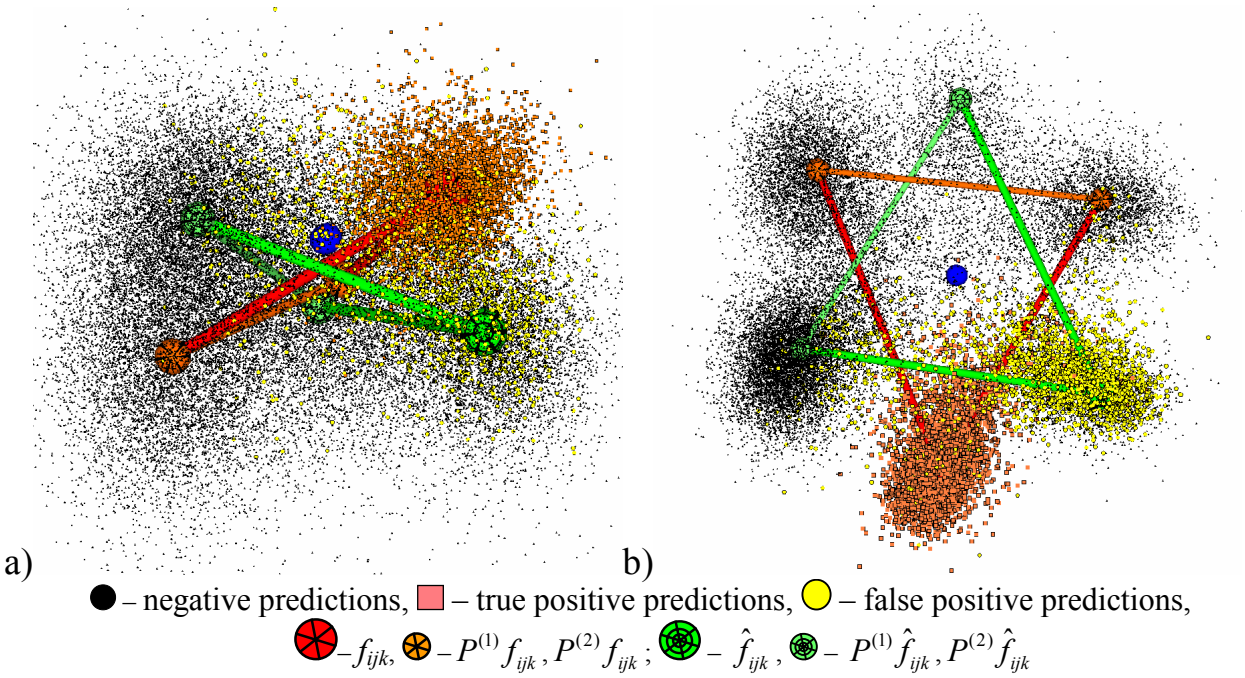

Figure 2. Visualization of the distribution of predictions of GLIMMER gene-finder in 64-dimensional space of codon frequencies.
Every point corresponds to one ORF. Red and green triangles denote the same structures as described at the figure 1.

a) *Escherichia coli*. Projection on the 1st and 3d principal components.

b) *Caulobacter crescentus.* Projection on the 1st and 2d principal components.

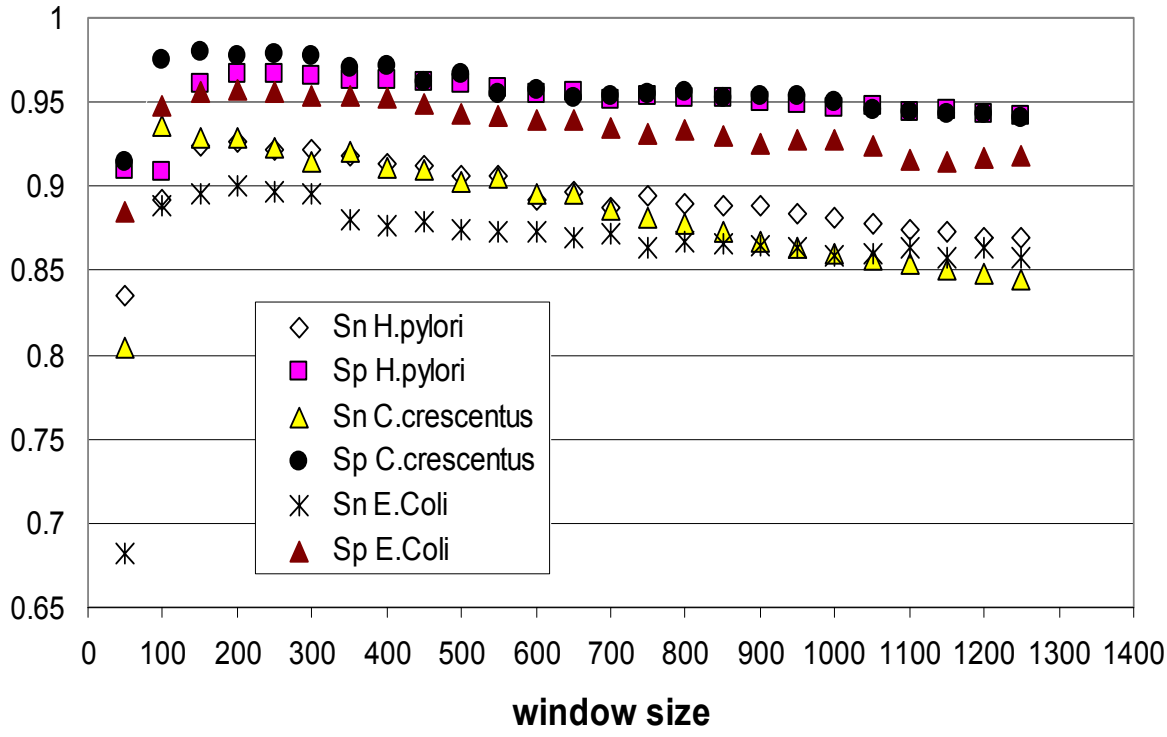

Figure 3. Window-size dependence of the algorithm