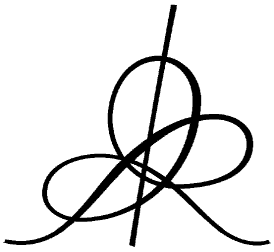
VISUALIZING THE SPATIAL STRUCTURE OF TRIPLET DISTRIBUTIONS IN GENETIC TEXTS

Andreï Yu. ZINOVYEV



Institut des Hautes Études Scientifiques
35, route de Chartres
91440 – Bures-sur-Yvette (France)

Avril 2002

IHES/M/02/28

VISUALIZING THE SPATIAL STRUCTURE OF TRIPLET DISTRIBUTIONS IN GENETIC TEXTS

Andreï Yu. Zinovyev

Institut de Hautes Etudes Scientifiques, Bures-sur-Yvette, France

e-mail: zinovyev@ihes.fr; http://www.ihes.fr/~zinovyev

Abstract

We analyze several genetic texts, using visual representations of triplet count distribution in a sliding window. After appropriate normalization and projection onto a linear manifold spanned by the first three principal components, the distribution of 64-dimensional vectors of triplet frequencies appears as a cloud of points, displaying a well-detectable cluster structure. In several complete bacterial and yeast genomes selected for analysis, as well as some model sequences, 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. Formation of such a seven-cluster structure reflects the unevenness of codon usage, particularly the presence of codon bias. 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 (for example, known codon usage and codon order correlations.) In both cases, final segmentation corresponds with convincing accuracy to the positions of known exons (sensitivity and specificity both higher then 0.9) compared on the base-pair level.

Introduction

One of the most well-stated problems in mathematical molecular biology over the last two decades is that of computational gene recognition; *i.e.*, of predicting the location and content of coding regions in sequenced genomes. Since this problem was of such great practical interest to experimental biologists, it became the main arena for testing and comparing various mathematical methods and ideas. This competition led to the development of a wide variety of both free and commercially available software tools, upon which modern bioinformatics is based.

There are many reviews of the spectrum of approaches used in computational gene recognition. For recent ones see, for example, Fickett (1996,) Claverie (1997,) Burge and Karlin (1998,) and Haussler (1998). Generally, all of them use one of the following information types for recognition: *signals*, *content measures*, and *similarity measures*, or some combination thereof.

Content measures are statistical properties of a DNA region that can aid in distinguishing coding from non-coding regions. A wide variety of such measures is used in modern gene recognition. Several authors (see, for example, Fickett, 1992) have carried out systematic comparative analyses of such measures, several of which have been identified as promising candidates for use in gene recognition. Many of

them implement the concept of codon bias; *i.e.*, that each species employs a bias in its choice of codons such that synonymous codons are not used with the same frequency. Examples of these are codon usage, composition, position asymmetry, and entropy measures (for definitions, see Fickett, 1996.)

Several authors have reported that a measure known as the inphase hexamer count (the number of hexamers in a segment of DNA text offset by 0, 1, and 2 base-pairs from the starting base of the test codons) appears to be the best content measure.

With few exceptions, almost all commonly used gene-finding programs employ a learning dataset for tuning the parameters of the learning rule. A method developed for unsupervised segmentation of whole DNA texts has been described in several recent papers (Bernaola, 2000; Li, 2001). It uses the fact of codon bias by introducing a 12-letter alphabet, corresponding to various positions of base-pairs in test codons. This method is able to detect borders between coding and non-coding regions without a preliminary learning stage.

Thus, many researchers report success in applying new, heuristically discovered measures in this field. Surprisingly, the relevant literature contains a relatively low number of publications in which the space of the variables used is thoroughly explored. As a result, there is still poor theoretical understanding of why some variables are good for gene recognition while others are not. For example, in one of these papers, Fickett (1992,) many measures were compared with the linear discriminative analysis method, which in fact, relies on a rather strong hypothesis concerning the structure of data distribution. M. Zhang (1997) successfully exploited quadratic discriminative surfaces in the spaces of some special variables, but left the underlying reasons for the application of quadrics unclear.

Of course, the success of a given technique in this field may be measured more or less objectively and independently. Much research has been carried out in the aim of comparing the effectiveness of various gene-finders on the base, exon, and gene levels (see, for example, the recent papers of Rogic, 2001 and Guigo, 2000.) The effectiveness of an approach often serves as its own self-justification. However, we believe that the detailed exploration of geometric metaphors for data-sets used in various learning algorithms can greatly improve their effectiveness and provide insight into the underlying mechanisms of the learning process. In this respect, multidimensional data *visualization* methods are invaluable because they can provide a foothold in attempting to understand what happens in the highly multidimensional spaces of learning processes; why some of them are very effective and successful while others are not.

In this paper we will try to explore the space of a very simple measure, which is in fact the oldest measure used for gene recognition: triplet counts in a sliding window. We will visually demonstrate the structure of a dataset used for learning and try to formalize the conditions of learning process effectiveness. We will show that in special cases, traditional use of a learning data-set (a set of examples of already known coding and non-coding regions) may be substituted for *learning without a teacher*; *i.e.*, an unsupervised procedure.

Very generally, the problem may be stated as follows: We have

1) ***codon usage***; *i.e.*, a set of three-letter words (trinucleotides) that may be used for coding biological information. Each trinucleotide has its own frequency, which is more or less maintained during coding. For example, some codons always have an almost-zero frequency (the *codon bias* phenomenon).

2) ***text***, containing two types of sequences: *coding regions*, consisting of codons that succeed each other without delimiters; these regions are strictly conserved during the evolutionary process; and *non-coding regions* (or *junk*) which allow *mutations,* and thus have no specific structure; for example, we can assume that non-coding regions are composed of the same set of codons, but only after a number of random insertions and deletions of various base-pairs.

Here we must underline that to a certain degree, these principles are very naive and oversimplified; we will nevertheless use them to construct several simple models of DNA text.

The following question emerges: Using text alone without knowing the code, is it possible to identify coding regions, and if so, with what accuracy it is possible to detect where the segments of coding information are? To answer this question, we will analyze the structure of triplet distribution in one very simple space of frequencies.

Results

Operations over codon usage

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:

$$P^{(1)} f_{ijk} \equiv \sum_{l,m,n} f_{lij} f_{kmn} ,$$

$$P^{(2)} f_{ijk} \equiv \sum_{l,m,n} f_{lmi} f_{ijn}$$

$\hat{f}_{ijk} \equiv C^R f_{ijk} \equiv f_{\hat{k}\hat{j}\hat{i}}$, $\hat{f}_{ijk} \equiv C^R f_{ijk} \equiv f_{\hat{k}\hat{j}\hat{i}}$, 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.

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 stated at the end of the introduction 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 effective f_{ijk} is in recognizing coding sites:

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

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

$$CC = \frac{1}{2}\left(\left\| P^{(1)} f_{ijk} - f_{ijk}^{(1)} \right\| + \left\| P^{(2)} f_{ijk} - f_{ijk}^{(2)} \right\| \right).$$

Visualization using principal component analysis

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 each frequency of the *s*th possible triplet.

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 *s*th coordinate of the *i*th point after normalization, and

$m_s = \frac{1}{N}\sum_{t=1}^{N} x_{ts}$ is the mean value of the *s*th coordinate, and

$\sigma_s = \sqrt{\frac{1}{N}\sum_{t=1}^{N} (x_{ts} - m_s)^2}$ is the standard deviation of the *s*th coordinate.

We then apply the principal components algorithm (see *Methods* section,) 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:

$$D = \frac{1}{N}\sum_{t=1}^{N}\left\|X_t - \overline{X}\right\|^2 , \ \overline{X} \text{ is the mean point of the data-point distribution}$$

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, 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.

Real texts

Figures 1 through 5 present 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. A rough 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 $f_{ijk}^{(1)}$ (1-shift,) and the last third are close to the $f_{ijk}^{(2)}$ (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, assuming an absence of correlations. Indeed, the calculated values of *CC* are not high (see Table 1, ***CC*** column.)

Model texts

In order to understand which property of codon usage renders the formation of such spatial structures possible, we generated four model distributions of codon usage and constructed four model sequences. They all are 100,000 bp long, with an average coding region length of 500 bp. The exon length distribution was chosen to be Gaussian, with a standard deviation of 100 bp. The minimal length of exons was 50

bp. The junk region length distribution was chosen to be uniform in the 50-1000 bp interval. "Junk" was generated from the same codons as the coding regions, but after $l/4$ random insertion and deletion operations of one base-pair (l is the length of a piece of junk.)

- In the UNIFORM model text, the frequencies of all codons were set to 1/64.
- In the RANDOM model text, the frequencies of all codons were set to be random and normalized on the unit sum.
- In the RANDOM_BIAS model text, the frequencies of all codons were set to be random, after which half of them (randomly selected) were set to zero. The distribution was then normalized on the unit sum.
- In the GC_CORR model text, the frequencies of all codons were set to be proportional to the codon GC-content.

It is clear from the table and from Fig. 6 that the RANDOM_BIAS text leads to efficient detection of coding segments. The worst case is the UNIFORM text, although GC_CORR also does not result in detection with any reasonable accuracy. All generated sequences are available on the accompanying web-page: http://www.ihes.fr/~zinovyev/bullet.

Clusterization, phase graphs

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 clusterization algorithm. After clustering the distribution into seven clusters, 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, which are very commonly used measures 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.,* coding bases predicted to be non-coding, and

FN is the number of false-negatives, *i.e.,* non-coding bases predicted to be coding.

The results are shown in the $\mathbf{Sn_1}$ and $\mathbf{Sp_1}$ columns of Table 1. These values are quite high, especially if we take into account that fact that the method does not really use a learning dataset. The only parameter – window size – may be visually evaluated by comparing pictures of data constructed with various values of *W* (see following sections.) In fact, the dependence of effectiveness on window-size is not strong over a rather long interval of *W*.)

This method of sequence segmentation may also be regarded as detection of the borders between "coding" and "non-coding" regions. It is possible to draw a graph of the "coding phase" for a genome. As one can see from Figure 7, which illustrates several randomly chosen regions of the analyzed genomes, change in coding phase correctly detects the borders between coding and non-coding regions. Of course, this method has the same drawback mentioned in Bernaola, *et al.* (2000,) namely that if two exons are close enough and the number of base-pairs between them is divisible by 3, they will most likely be recognized as a single exon.

Table 1

Summary table of results and parameters used for genome analysis

Sequence	*L*	*W*	*p*	$\nu^{(3)}$	% of coding bases	CP	CC	CD	Sn_1	Sp_1	Sn_2	Sp_2
Helicobacter pylori, complete genome (NC_000921)	1643831	300	120	0.35	90	0.68	0.28	1.21	0.93	0.97	0.93	0.98
Caulobacter crescentus, complete genome (NC_002696)	4016947	300	300	0.21	91	1.07	0.16	0.74	0.93	0.97	0.94	0.98
Prototheca wickerhamii mitochondrion (NC_001613)	55328	120	18	0.17	49	0.83	0.11	1.34	0.82	0.93	0.84	0.95
Saccharomyces cerevisiae chromosome III (NC_001135)	316613	399	99	0.16	69	0.45	0.10	1.77	0.90	0.88	0.90	0.90
Saccharomyces cerevisiae chromosome IV (NC_001136)	1531929	399	120	0.15	73	0.48	0.09	1.69	0.89	0.91	0.92	0.92
Model text RANDOM	100000	500	30	0.13	49	0.46	0.05	1.40	0.90	0.61	0.82	0.77
Model text RANDOM_BIAS	100000	500	30	0.30	45	1.20	0.06	0.53	0.99	0.83	0.94	0.90
Model text UNIFORM	100000	500	30	0.09	50	0.08	0.06	6.90	0.70	0.49	0.78	0.53
Model text GC_CORR	100000	500	30	0.09	49	0.16	0.05	3.60	0.71	0.49	0.78	0.56

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, 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. Once the centers are known, it is possible to visualize the trajectories of shorter DNA segments. If a sliding window is set every three base-pairs (p=3,) the sequence of data points in the data space will form a kind of continuous trajectory (because two neighbors differ only by two codons, one of which is eliminated, and the other added.) The trajectory tends to be located near cluster centers. Two examples of such trajectories for relatively short segments of DNA text are shown on Figs.8 and 9.

Single genes

Detection of the presence of a "coding phase" can be demonstrated not only for complete genomes, but also for short DNA segments containing a single long exon. In some cases it may be possible to determine whether a given DNA segment of, for example, 500-1000 bp, contains information coded by codons. Since in this case we have only one exon and there is no "accidental" phase-shift, we should set sliding windows every *p* positions, where *p* is not divisible by 3; for example, p=1. In the space of triplet frequencies, the succession of sliding windows will form some trajectory. For the coding sequence, this trajectory will form a triangle-like path, such as shown in Fig.10. If the sequence contains both coding and non-coding information, only part of the trajectory will be triangle-like. It would be interesting to detect in a formal way the instant at which the change in trajectory-type occurs.

Dependence on window-size

It is interesting to see how the distribution cluster structure changes with variation in the value of window size *W*. It is evident that in the case of very small window size (30-50 bp,) the calculated statistics of triplet frequencies are not reliable. The large amount of statistical noise does not allow the cluster structure to form. In the case of large windows (>1500 bp,) clusters do not correspond well to coding and non-coding regions. Optimal window-size seems to correspond to average exon length.

An example of evolution in the visual presentation of data distribution is shown in Fig.12. It is not easy to propose a natural measure for the formal calculation of optimal window size. We tested several candidates (such as average in-cluster variation,) but still found the visual evaluation of optimal window-size to be the most useful and practical. Also, if the average length of protein-coding exons is known, it is reasonable to take this value to be the window-size. Table 2 presents the results of testing segmentation for various window sizes. It is clear that the dependence of segmentation accuracy on *W* is not very strong over a rather long interval.

Table 2

Comparison of exon recognition accuracy for several values of *W*

Sequence	*W*	Sn_1	Sp_1
Helicobacter pylori, complete genome (NC_000921)	50	0.799	0.907
	100	0.889	0.921
	150	0.929	0.969
	200	0.930	0.971
	250	0.929	0.974
	300	0.928	0.973
	350	0.919	0.972
	400	0.912	0.971
	450	0.908	0.968
	500	0.911	0.966
	550	0.888	0.962
	600	0.901	0.963
	650	0.896	0.961
	700	0.892	0.959
	750	0.892	0.958
	800	0.893	0.957
	850	0.891	0.952
	900	0.889	0.950
	950	0.882	0.949
	1000	0.878	0.949
	1050	0.881	0.947
	1100	0.876	0.944
	1150	0.869	0.946
	1200	0.864	0.943
	1250	0.862	0.941

Human genes

The methods of visualization proposed are mainly suitable for bacterial and lower eukaryote genomes (such as those of yeast,) due to the comparatively high density of coding information. Therefore, the distribution of triplet frequencies forms a clearly visible seven-cluster structure, after projection onto the linear 3-dimensional principal manifold. Otherwise, large numbers of windows, in the absence of a synchronized coding phase, would make the number of "coding" windows statistically negligible, therefore invisible, after projection onto the principal manifold.

Nevertheless, knowing genome annotation, we can still find triangle-like structures in the distributions of triplets by assigning a weight to every data point that corresponds to the number of coding and non-coding windows. "Coding" data-points are assigned higher weights and "non-coding" ones lower weights, proportional to the relation between the numbers of coding and non-coding windows. Principal component analysis with weights may then be applied (see Methods section). This was done for the HMR195 dataset of human genes used for testing gene-finder programs in (Rogic 2001,) available at http://www.cs.ubc.ca/rogic/evaluation/dataset.html. The results are shown in Fig.11.

Non-homogeneity of codon usage

Another measure is used to compare the maximal distance between the zero and non-zero coding phases, and the standard deviation in the measurement of codon usage.

$$CD = \frac{\sigma f}{CP},$$

where $\sigma f = \sqrt{\frac{1}{N}\sum_{l=1}^{N}\left\|\left(f_{ijk}\right)_l - f_{ijk}\right\|^2}$, $(f_{ijk})_l$ are codons frequencies in the lth window.

The standard deviation of codon usage is the result of two factors: statistical deviation caused by measurements being carried out in a comparatively narrow window, and non-homogeneity of codon usage along the whole genome. The *CD* value is responsible for the distribution "contrast," *i.e.,* how well various phases clusters are separated from each other in data space. *CD* values for the genomes analyzed are shown in the *CD* column of Table 1. A "contrast" picture of distribution corresponds to the $CD \le 1$ values. Satisfactory clusterization corresponds to the case in which $CD \le 1.7$.

Discussion

We can interpret the process of sequence-generating in probabilistic terms in order to make it easier to understand how our approach is related to the standard approaches, such as hidden Markov modeling. Using the notation of Burge and Karlin (1997,) we can consider a *set of states* Q = {N, $E_0^{(+,)}$ $E_1^{(+,)}$ $E_2^{(+,)}$ $E_0^{(-,)}$ $E_1^{(-,)}$ $E_2^{(-)}$}. In every one of these states, triplets are generated according to the corresponding frequency distributions $F = \{junk_{ijk}, f_{ijk}, f_{ijk}^{(1)}, f_{ijk}^{(2)}, \hat{f}_{ijk}, \hat{f}_{ijk}^{(1)}, \hat{f}_{ijk}^{(2)}\}$. The algorithm generates a sequence as follows:

1) One of the states $q_t \in Q$ is taken randomly, with probability A_{qt} ($s = 1 \ldots 7$). Transition into the same state is prohibited.

2) A length (state duration,) d_t, corresponding to the state q_t is generated conditional on the value of q_t from the length distribution l_{qt}.

3) A sequence segment s_t of length d_t is generated, conditional on d_t and q_t, according to an appropriate triplet-generating frequency $f_t \in F$ for state type q_t.

4) This process is repeated until the sum $\sum_{t=1..n} d_t$ of the state durations first equals or exceeds the sequence length L, at which point the last state duration d_n is appropriately truncated, the final stretch of sequence is generated, and the process stops. The sequence generated is simply the concatenation of the sequence segments, $S = s_1 s_2 \ldots s_n$.

This process is similar to the semi-Markov type, but is actually simpler, since it does not use transition probabilities (of course, it is possible to introduce them.) It would then be possible to derive a variant of the Viterbi algorithm, but we could claim that performance of such a probabilistic calculation would not be better than the simpler method proposed.

Using a sliding window is somehow "out of fashion" in modern biological sequence analysis. In fact, it explicitly introduces scale into the process of evaluating frequencies. But hidden Markov models with explicit state durations also implicitly introduce scale by length distribution. It is possible to develop a weighted scheme of calculating triplets in a sliding window that takes exon length distributions into account.

As mentioned above, this model is oversimplified, but in prokaryotes (for example, *Helicobacter pylori*) it has essentially the same performance as the GLIMMER gene-finder (Salzberg, *et al.*, 1998,) despite the fact that it does not include *any* parameters other than sliding window size. This means that the model captures the essential part of the information needed to discriminate between coding and non-coding regions, and further complication (such as introducing Markov models with a complicated scheme of interpolating oligomer frequencies, as in Salzberg, *et al.*, 1998) is unnecessary and leads to an excessive number of parameters. In addition, because of unsupervised learning, the method does not depend on the way genes are chosen for the learning process.

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 (frequently 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 worse than others more suitable for a 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 once again 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, codon distribution that is efficient for gene recognition corresponds to a high value of mutual information, *i.e.,*

$$M = \sum_{ijk} f_{ijk} \log_2 \frac{f_{ijk}}{p_i p_j p_k},$$

where p_i is the average frequency of letter $i \in \{A,C,G,T\}$. This value may be zero only in the case $f_{ijk} = p_i p_j p_k$. 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 we have shown that visual analysis of a spatial dataset structure does not require the use of a learning dataset in order to accurately solve gene recognition tasks, at least in DNA segments with high concentrations of coding information. 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. We also showed that even in the case of completely unknown codon usage properties, it is possible to predict how reliable the gene recognition procedure will be. We deliberately did not use additional biological information in detecting gene borders. Of course, if the method were to be proposed for practical use, it would be reasonable to conduct some post-processing steps, *e.g.*, aligning the borders of exons, using known signals (stop-start codons, promoter regions, etc.,) or merging very short regions that have the same phase, thereby enhancing the accuracy of the method. What we tried to show is that in many cases, the structure itself provides good accuracy for DNA segmentation with respect to the coding phase.

Methods

Principal component analysis is an effective method for reducing the dimension of experimental datasets and may be exploited as a data visualization method.

Principal component analysis uses orthogonal projection of data points X_i (i – the number of data point) onto a linear manifold spanned by several first eigen vectors (corresponding to the highest eigen values) of covariation matrix S:

$$S = \frac{1}{N} \sum_{i=1..N} (X_i - \overline{X})(X_i^T - \overline{X}^T),$$

where $\overline{X}$ is the average vector. For purposes of data visualization, the first three principal components (eigen vectors) may be used.

If every point of a dataset has some weight w_i, then the formulas should be slightly corrected as follows:

$$S = \frac{1}{\sum_{i=1..N} w_i} \sum_{i=1..N} w_i (X_i - \overline{X})(X_i^T - \overline{X}^T), \quad \overline{X} = \frac{\sum_{i=1..N} w_i X_i}{\sum_{i=1..N} w_i}.$$

For example, if a dataset is divided into two unequal classes with N_1 and N_2 points, it is sometimes then reasonable to assign to every point a weight that is inversely

proportional to the number of points in the class, *i.e.*, weights $w_1 = \frac{N_1 + N_2}{N_1}$ for points from the first class and $w_1 = \frac{N_1 + N_2}{N_2}$ for the second class.

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 package. Technically, the data visualization and all illustrations were produced using the ViDaExpert data visualization tool under Windows, and are available at the author's home page: http://www.ihes.fr/~zinovyev/vidaexpert/vidaexpert.htm.

Conclusion

In this paper we have reported the results of analysis carried out on several genetic texts, based on visual representations one of the oldest measures used for gene recognition: triplet counts in a sliding window. We showed that after appropriate normalization and projection onto a linear manifold spanned by the first three principal components, the distribution of 64-dimensional vectors of triplet frequencies appears as a cloud of points with a rather well-defined cluster structure. We also showed that, for several complete bacterial and yeast chromosome genomes chosen for study, as well as for some model sequences, the structure consists of seven clusters, corresponding to the presence of protein-coding information in three possible phases in one of the two complementary strands, and in non-coding regions. Formation of such a seven-cluster structure reflects the unevenness of codon usage, particularly the presence of codon bias.

Knowledge of the existence of this structure allows developing methods for segmenting a sequence into regions with the same coding phase and non-coding regions. The method may be either completely unsupervised, or use some "external" information (for example, known codon usage and correlations in codon order.) With convincing accuracy in both cases, the final sequence segmentation corresponds to known exon positions (sensitivity and specificity are both higher then 0.9,) with respect to the base level. In this respect, it would be interesting to explore spaces of other learning processes; for example, those of inphase hexamers, widely used in hidden Markov models, which are very popular in modern gene recognition. We believe that a clear understanding of the spatial structure of data distribution will lead to a higher quality of gene recognition, as well as make it more reliable and controllable. Statistical methods alone are definitely not enough for successful gene recognition, however, they remain one of the main components of modern gene recognition, especially in cases of newly sequenced complete genomes in which many genes have no analogs in databases. The methods we propose are mainly

suitable for bacterial and low-eukaryote genomes (such as yeasts,) because of the comparatively high density of coding information. However the principles considered are rather universal and could certainly be exploited in other studies. For example, we show that the dataset of human gene triplet distributions has the same symmetrical structure as found in lower organisms.

Acknowledgements

The author is thankful to Alexander Gorban for remote consultations while this work was underway, and to Tatyana Popova for several useful ideas. The author also thanks Alessandra Carbone for very fruitful discussions, Misha Gromov for the interest he expressed in this work, and Noah Hardy for editing the manuscript.

References

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.

Burge, C.B. and Karlin S. Prediction of Complete Gene Structures in Human Genomic DNA. J. Mol. Biol. 268, 78–94

Burge, C.B. and Karlin S. 1998. *Curr. Opin. Struct. Biol.* **8:** 346–354.

Claverie, J.-M. 1997. *Hum. Mol. Genet.* **6:** 1735–1744.

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

Fickett, J.W. and A. Hatzigeorgiou. 1997. *Genome Res.* **7:** 861–878.

Fickett, J.W. and C.-S. Tung. 1992. *Nucleic Acids Res.* **20:** 6441–6450.

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

Guigo R., Agarwal P., Abril J.F., Burset M., Fickett J.W. (2000). An assessment of gene prediction accuracy in large DNA sequences. Genome Research, 10(10):1631-1642.

Haussler, D. 1998. *Trends Guide Bioinformatic*s, pp. 12–15.

Li W. (2001) New stopping criteria for segmenting DNA sequences. Physical Review Letters, 86(25):5815-5818.

Rogic S., Mackworth A.K., Ouellette F.B.F. Evaluation of Gene-Finding Programs on Mammalian Sequences. Genome Research, 11(5):817-832.

Salzberg S.L., Delcher A.L., Kasif S., White O. (1998) Microbial gene identification using interpolated Markov Models. Nucleic Acids Research, Vol. 26, No. 2

Zhang M.Q. (1997). Identification of protein coding regions in the human genome based on quadratic discriminant analysis, Proceedings of the National Academy of Sciences (USA), 94: 559-564.

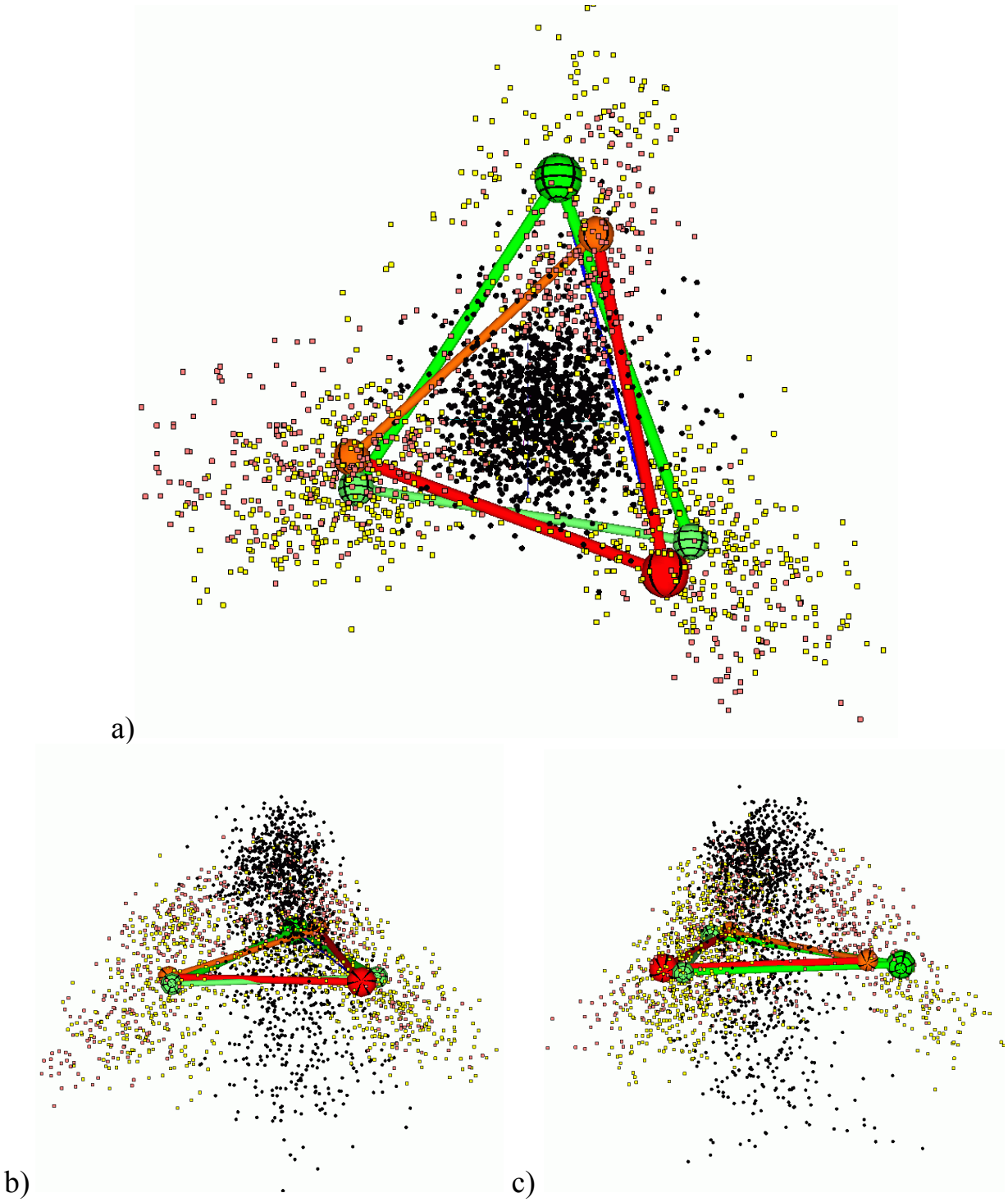


Fig.1. Visualization of *Prototheca wickerhamii* mitochondrion (GenBank NC_001613,)
a) top-view (components 1 and 2,) b) side-view (1 and 3,) c) side-view (2 and 3)

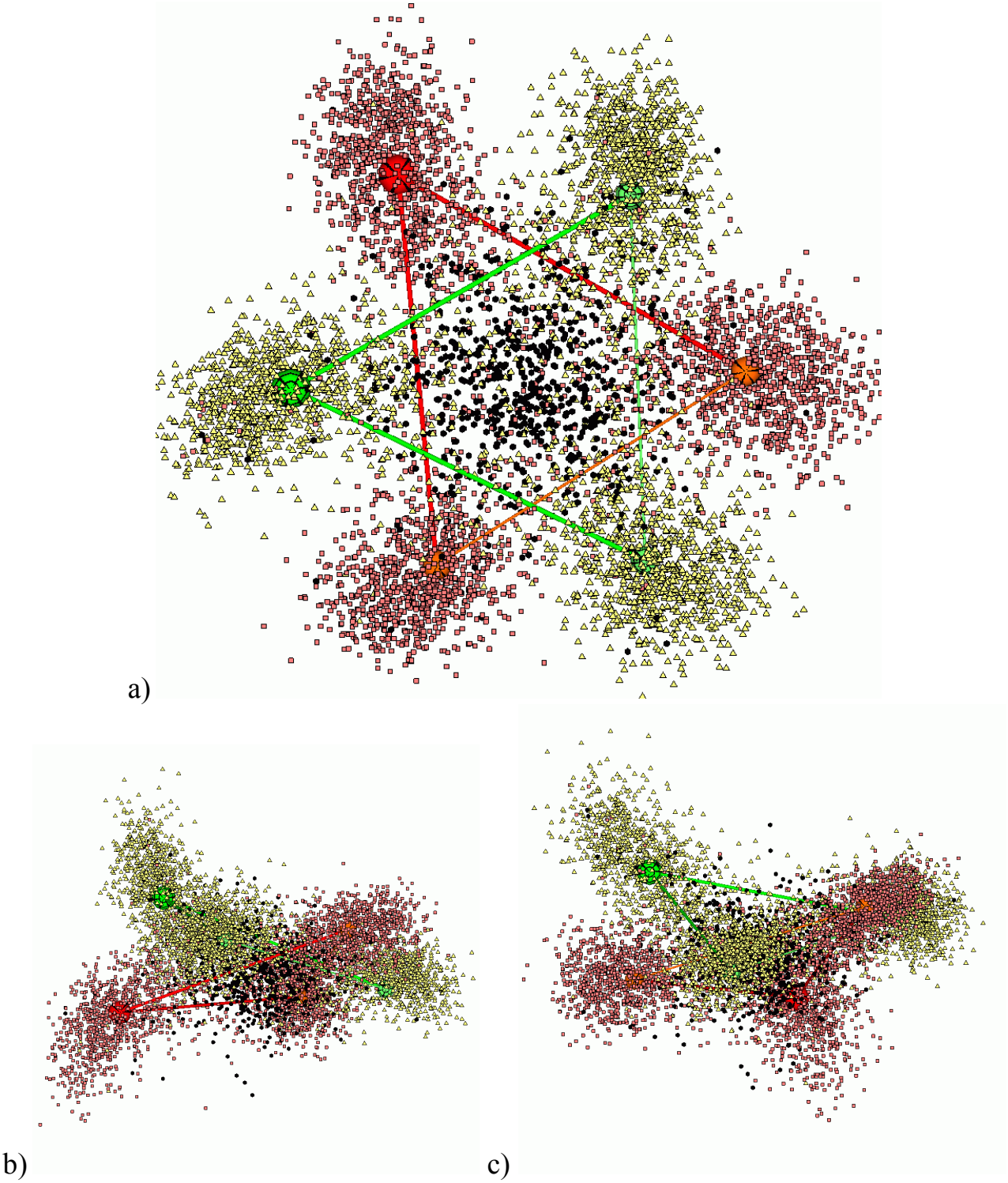


Fig.2. Visualization of *Caulobacter crescentus* (GenBank NC_002696,)
a) top-view (components 1 and 2,) b) side-view (1 and 3,) c) side-view (2 and 3)

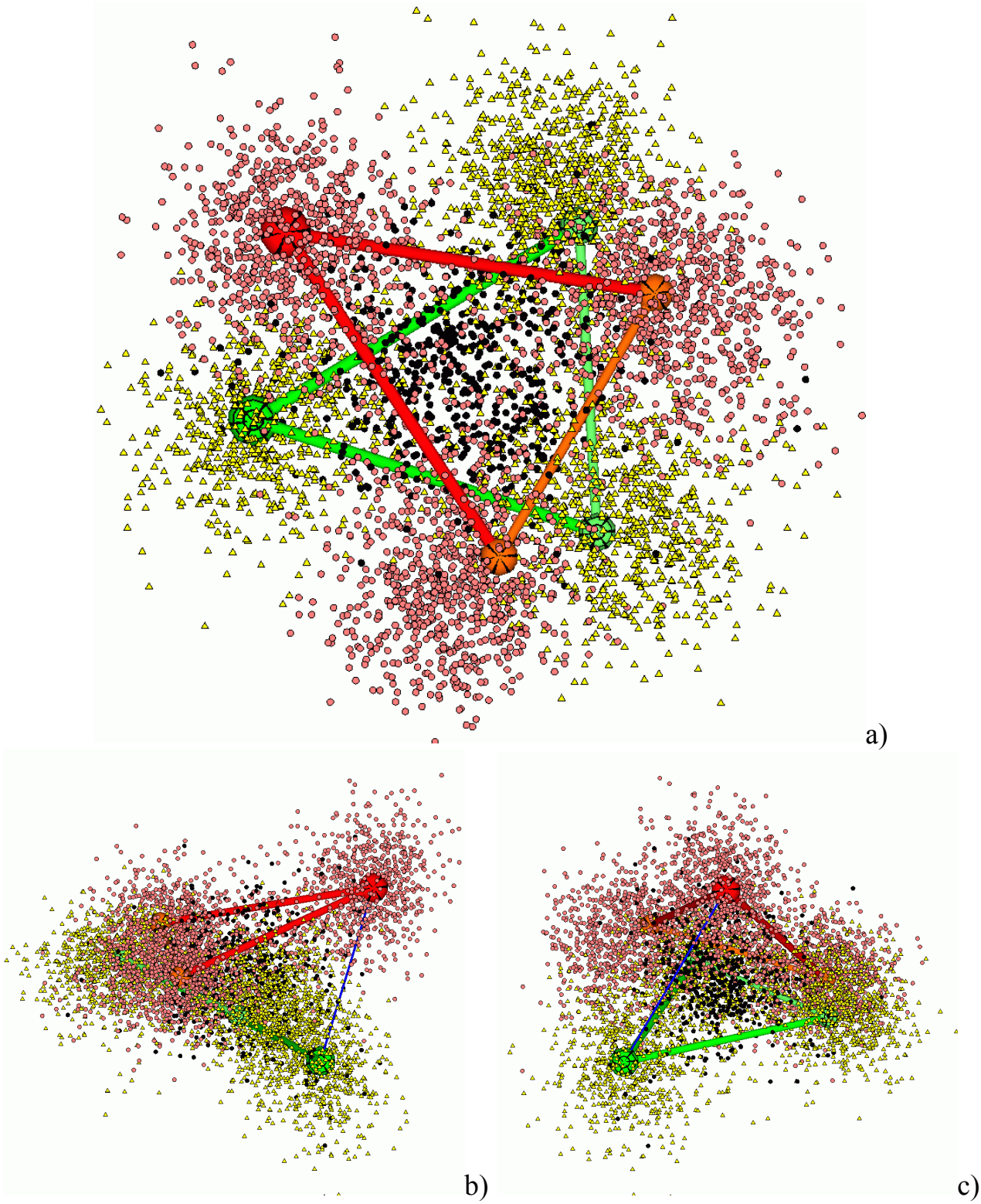


Fig.3. Visualization of *Helicobacter pylori* (GenBank NC_000921,)
a) top-view (components 1 and 2,) b) side-view (1 and 3,) c) side-view (2 and 3)

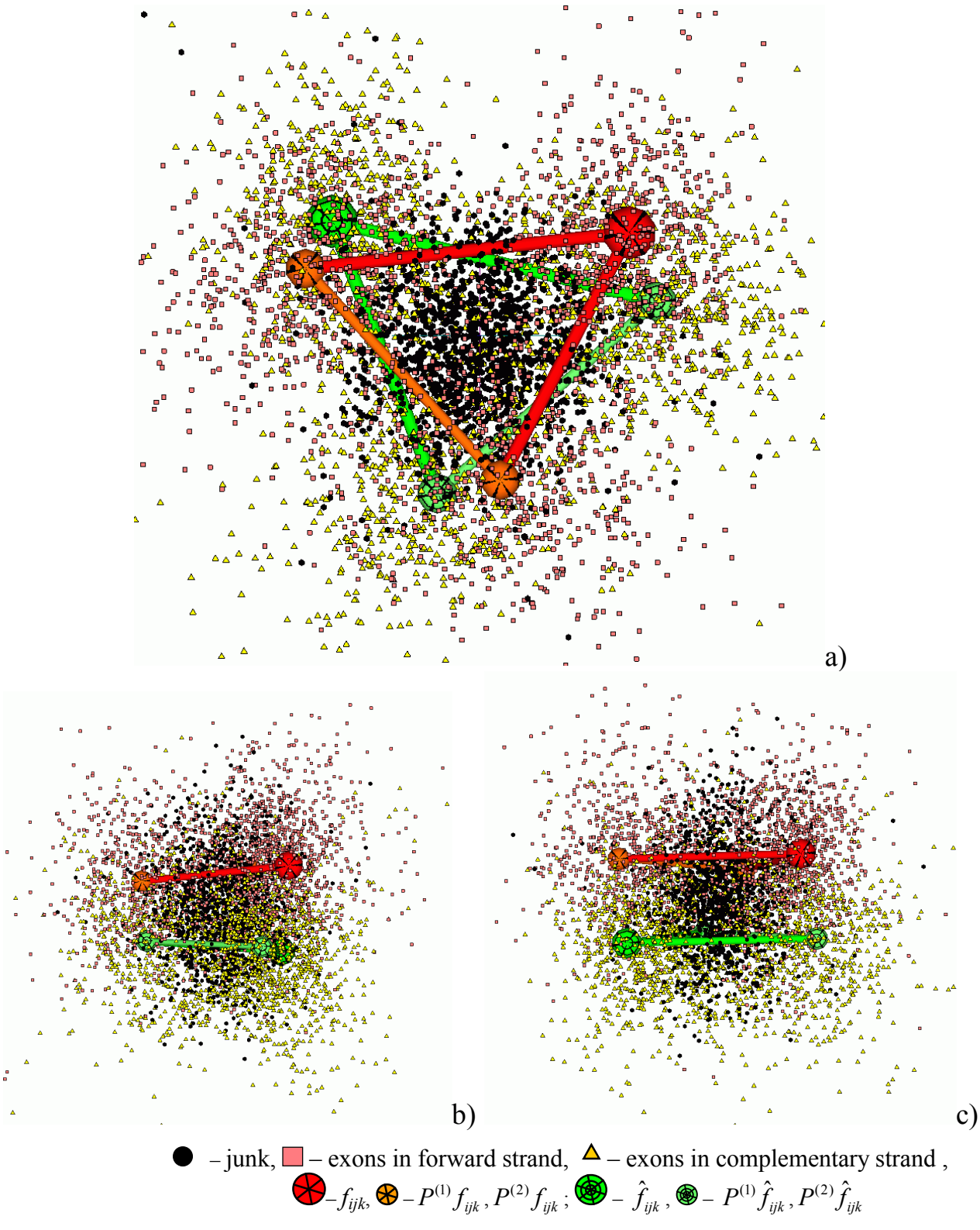


Fig.4. Visualization of *Saccharomyces cerevisiae* chromosome IV (GenBank NC_001136,)
a) top-view (components 1 and 2,) b) side-view (1 and 3,) c) side-view (2 and 3)

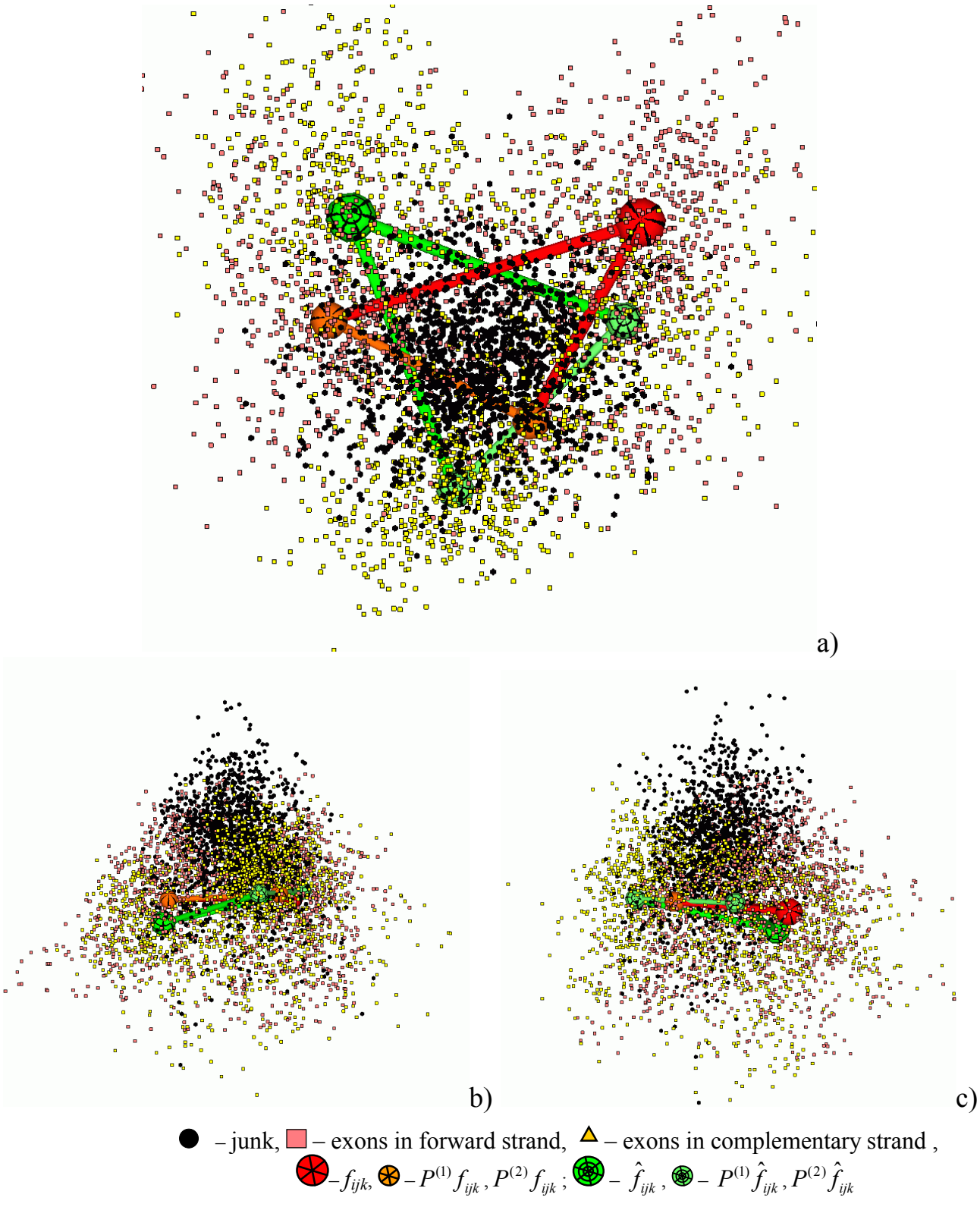


Fig.5. Visualization of *Saccharomyces cerevisiae* chromosome III (GenBank NC_001135,)
a) top-view (components 1 and 2,) b) side-view (1 and 3,) c) side-view (2 and 3)

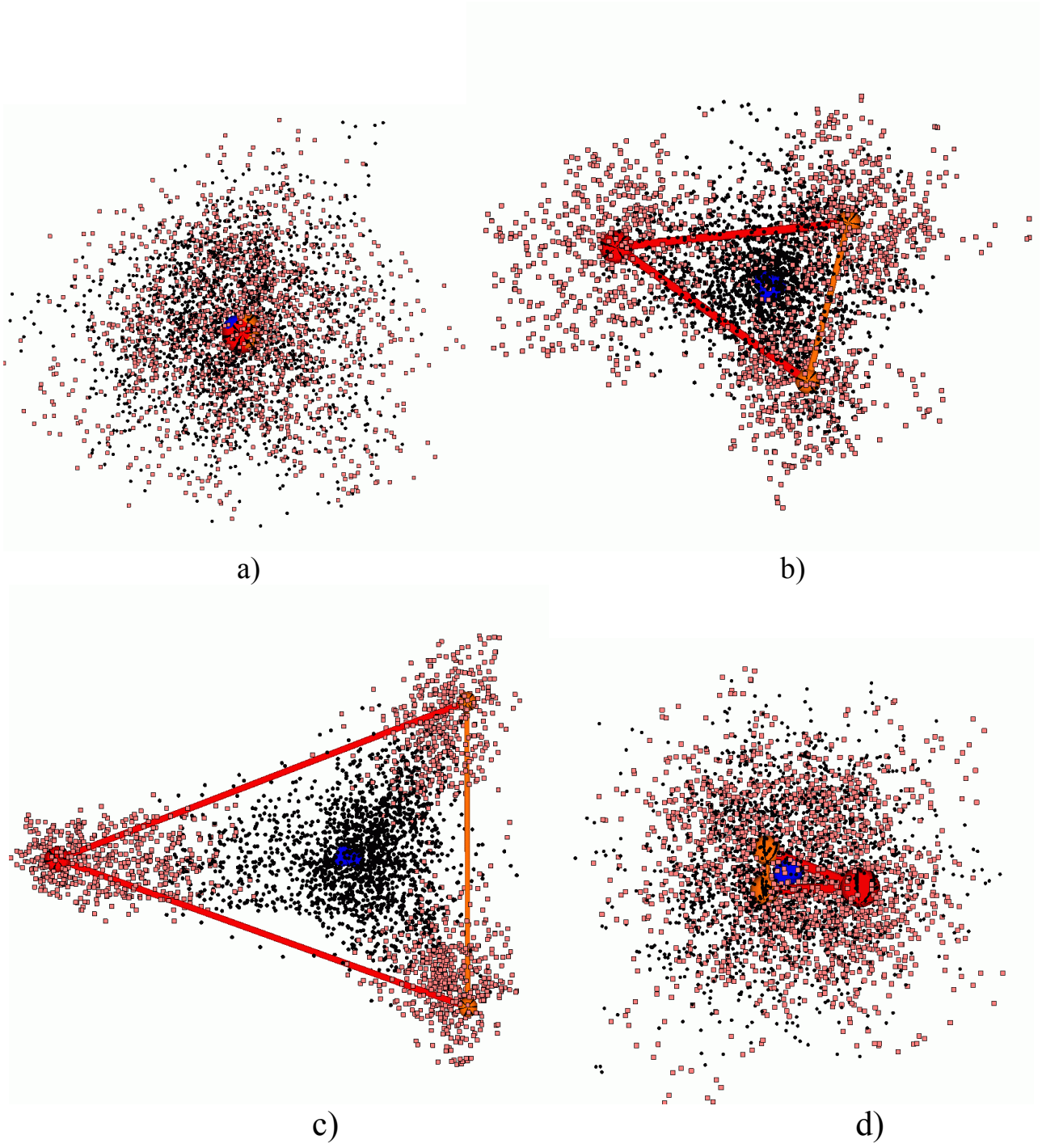
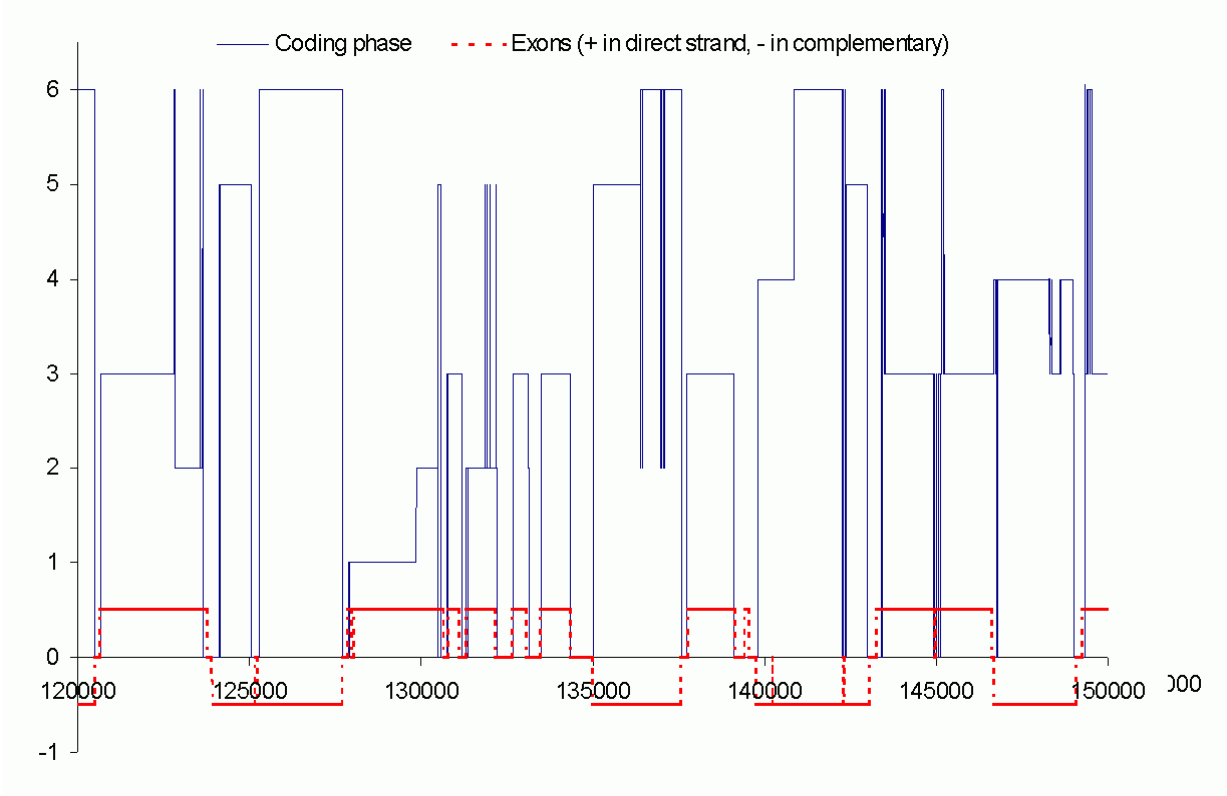
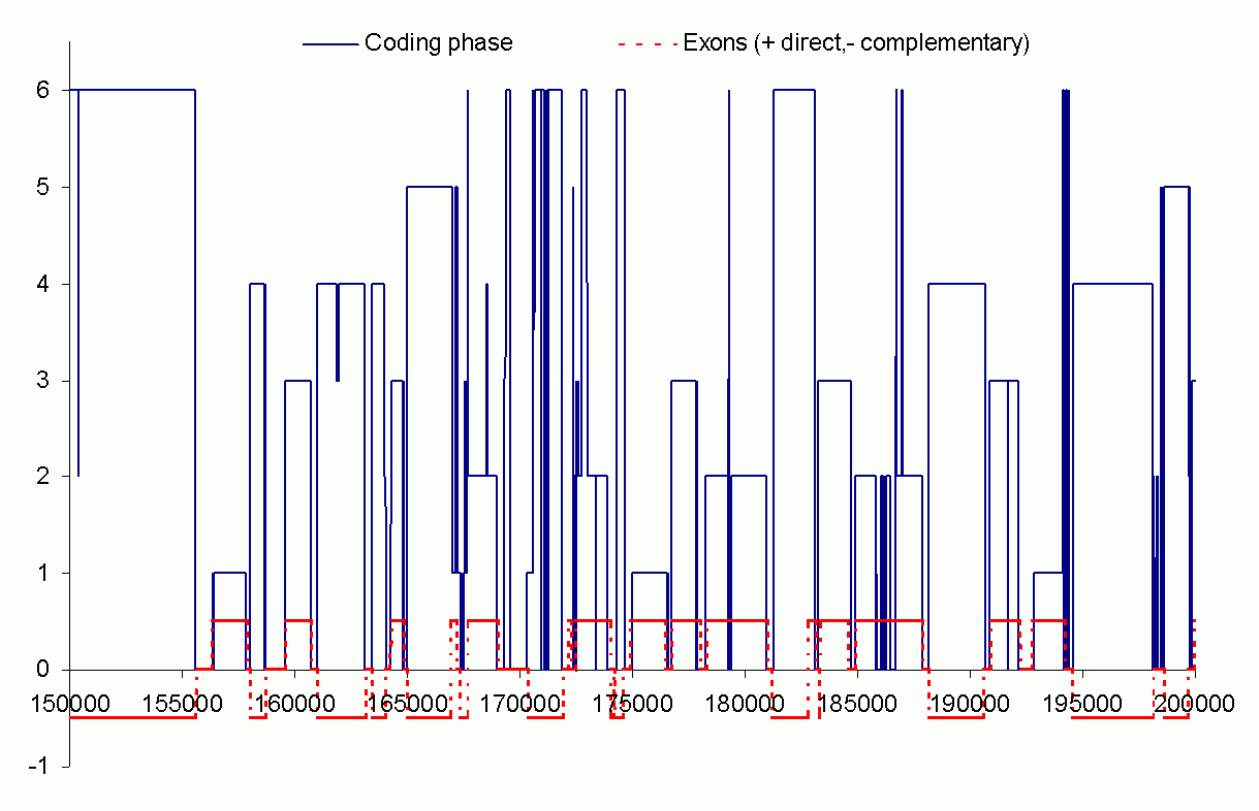


Fig.6. Visualization of model texts.
a) UNIFORM: uniform codon frequencies;
b) RANDOM: random codon frequencies;
c) RANDOM_BIAS: random codon frequencies, half of which are set to zero;
d) GC_CORR: codon frequency is proportional to codon GC-content.



a)



b)

Figure 7. Correspondence between predicted coding phase and positions of coding regions (exons).
a) *Caulobacter crescentus* region (120000..150000)
b) *Saccharomyces cerevisiae* chromosome IV region (150000..200000)

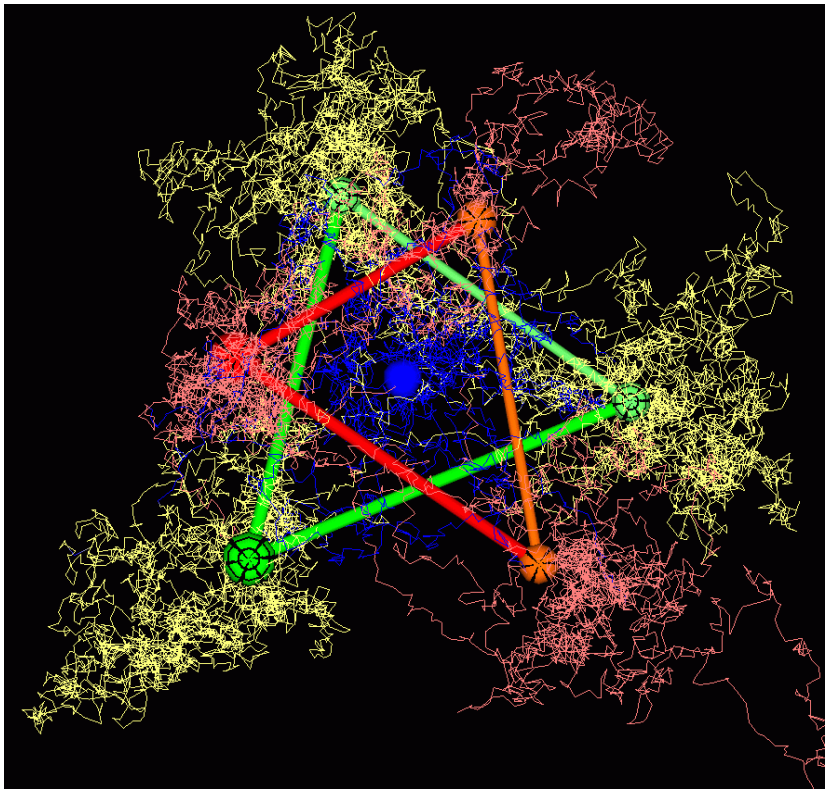


Fig.8. Trajectory of triplet usage (p=3) in dataspace for 50000..100000 region of *Helicobacter pylori*

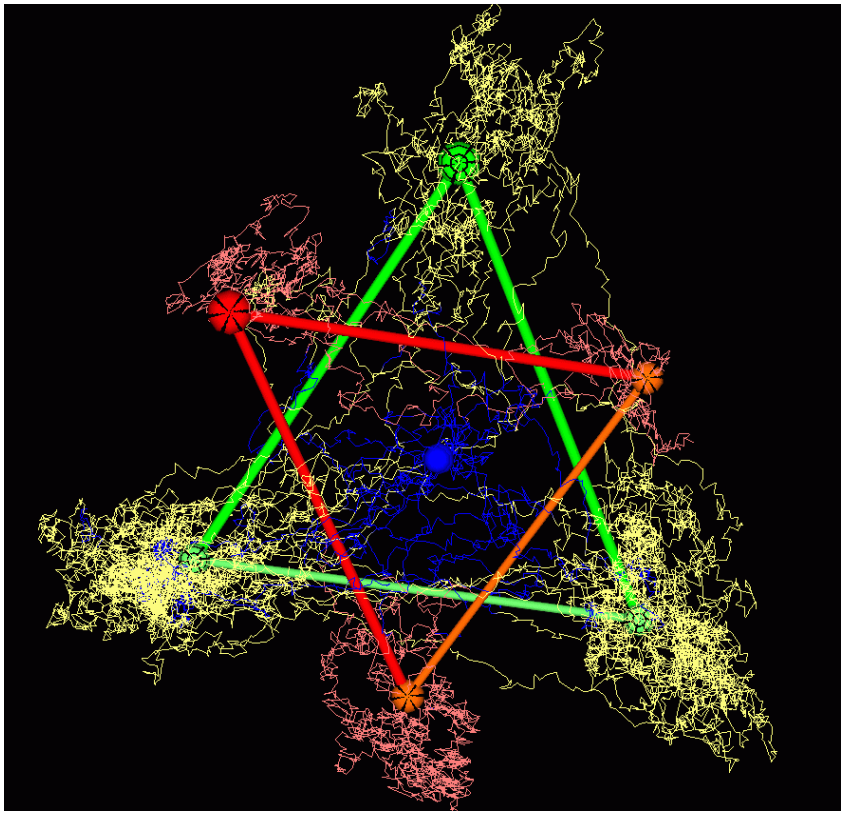
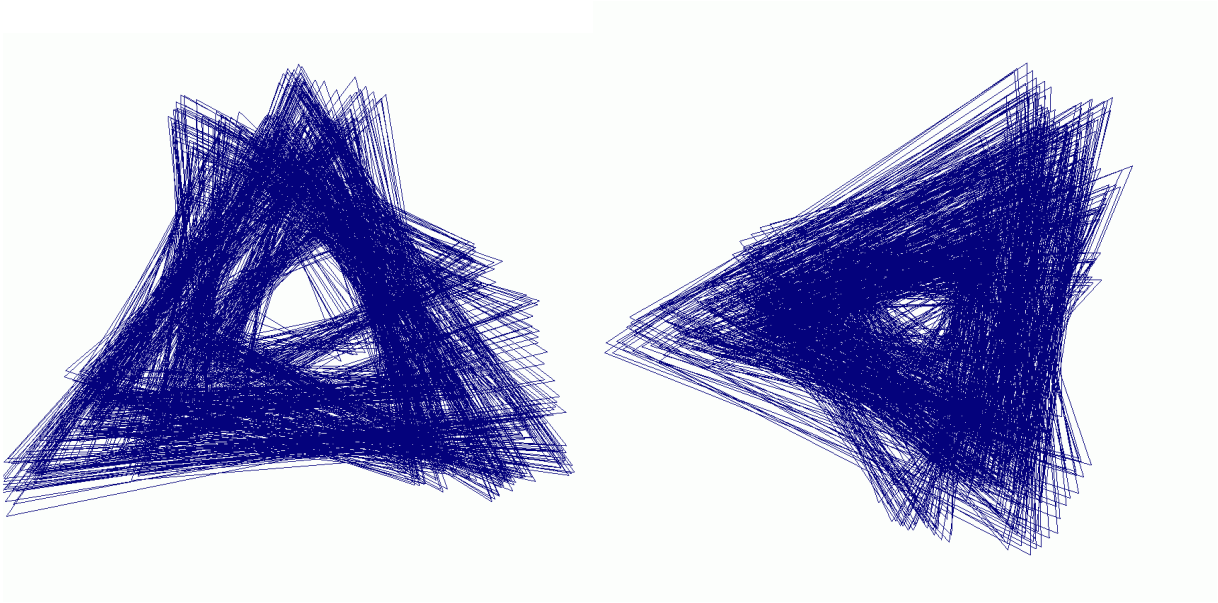
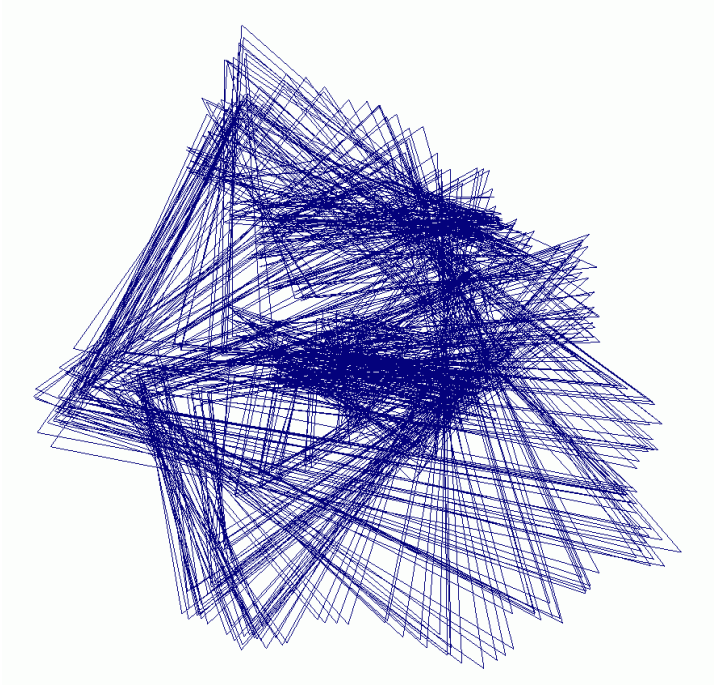


Fig.9. Triplet usage trajectory (p=3) in dataspace for 50000..100000 region of *Caulobacter crescentus*

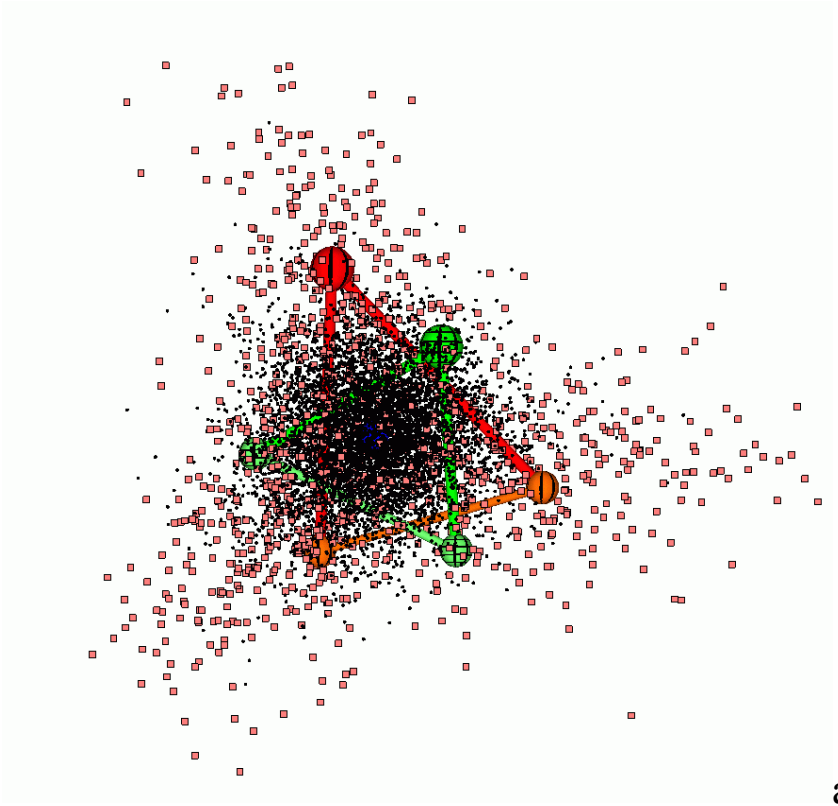


a) b)

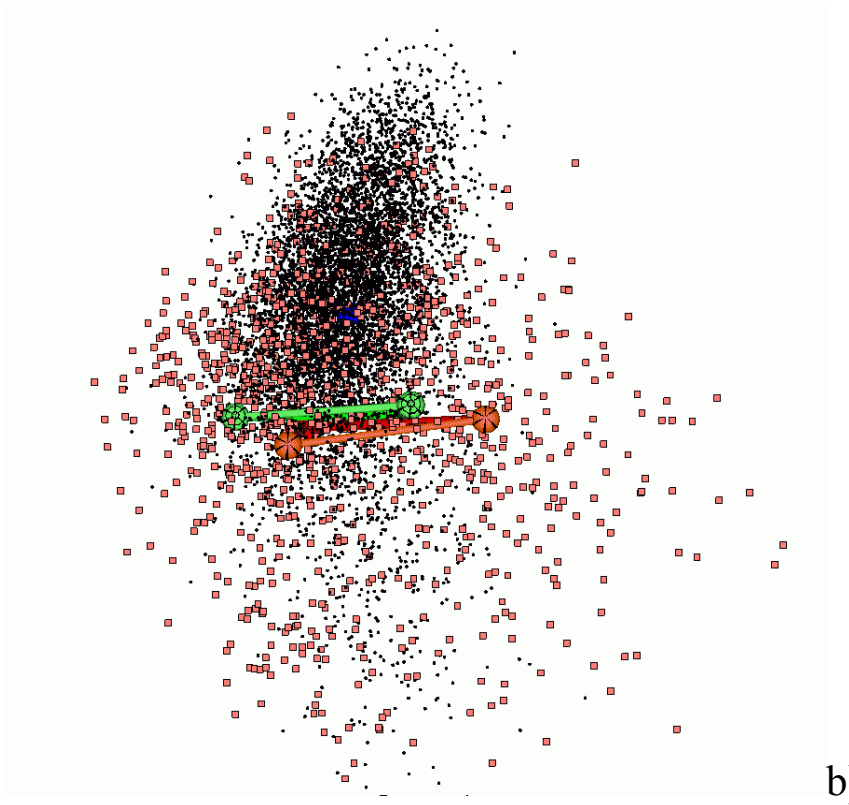


c)

Fig.10. Visualization of triplet usage trajectory for short DNA segments (p=1)
a) *E.coli trpC* gene; b) *E.coli trpE* gene;
c) intron 1 of *Prototheca wickerhamii* mitochondrion *cox1* gene.



a)



b)

● – junk, ■ – exons in forward strand, ▲ – exons in complementary strand ,
– f_{ijk}, – $P^{(1)} f_{ijk}, P^{(2)} f_{ijk}$; – $\hat{f}_{ijk}$, – $P^{(1)} \hat{f}_{ijk}, P^{(2)} \hat{f}_{ijk}$

Fig.11. Visualization of HMR195 human gene dataset.
Projection onto the weighted principal components: a) top-view; b) side-view.

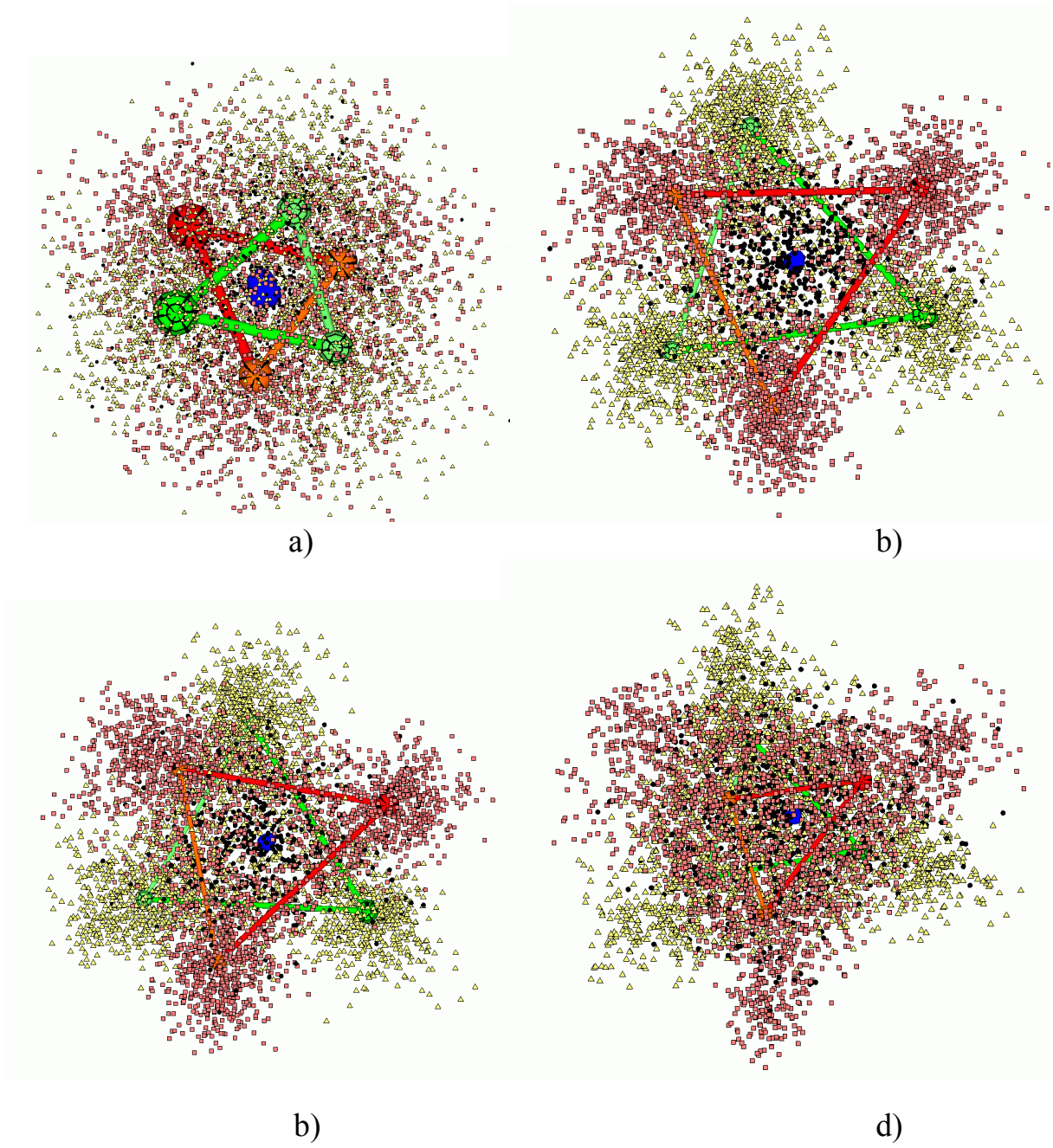


Fig.12. Visualization of triplet distributions for *Helicobacter pylori*, calculated with different window size *W*.
a) *W* = 51; b) *W*=600; c) *W*=900; d) *W*=2000