BIOINFORMATICS

Vol. 18 no. 11 2002
Pages 1523–1533

Euclidian space and grouping of biological objects

Vyacheslav N. Grishin[1] and Nick V. Grishin[1, 2,*]

[1]Department of Biochemistry and [2]Howard Hughes Medical Institute, University of Texas Southwestern Medical Center, 5323 Harry Hines Blvd, Dallas, TX 75390-9050, USA

Received on February 2, 2002; revised on April 17, 2002; accepted on April 23, 2002

ABSTRACT

Motivation: Biological objects tend to cluster into discrete groups. Objects within a group typically possess similar properties. It is important to have fast and efficient tools for grouping objects that result in biologically meaningful clusters. Protein sequences reflect biological diversity and offer an extraordinary variety of objects for polishing clustering strategies. Grouping of sequences should reflect their evolutionary history and their functional properties. Visualization of relationships between sequences is of no less importance. Tree-building methods are typically used for such visualization. An alternative concept to visualization is a multidimensional sequence space. In this space, proteins are defined as points and distances between the points reflect the relationships between the proteins. Such a space can also be a basis for model-based clustering strategies that typically produce results correlating better with biological properties of proteins.

Results: We developed an approach to classification of biological objects that combines evolutionary measures of their similarity with a model-based clustering procedure. We apply the methodology to amino acid sequences. On the first step, given a multiple sequence alignment, we estimate evolutionary distances between proteins measured in expected numbers of amino acid substitutions per site. These distances are additive and are suitable for evolutionary tree reconstruction. On the second step, we find the best fit approximation of the evolutionary distances by Euclidian distances and thus represent each protein by a point in a multidimensional space. The Euclidian space may be projected in two or three dimensions and the projections can be used to visualize relationships between proteins. On the third step, we find a non-parametric estimate of the probability density of the points and cluster the points that belong to the same local maximum of this density in a group. The number of groups is controlled by a σ-parameter that determines the shape of the density estimate and the number of maxima in it. The grouping procedure outperforms commonly used methods such as UPGMA and single linkage clustering.

Availability: The code of EESG program for Mathematica4 (Wolfram Research) as well as the details of the analysis are freely available at ftp://iole.swmed.edu/pub/EESG/.

Contact: grishin@chop.swmed.edu

INTRODUCTION

Most of similar objects in nature fall into discrete groups. In many cases it is difficult to understand the reasons that cause discreteness in an apparently continuous space. Regardless, it is useful to group objects and thus to reduce the complexity of the system from a large number of objects to a small number of clusters. Many generic methods address the problem of data clustering and grouping (Podani, 2000; Everitt *et al.*, 2001). However, only few of these approaches take into account specifics of biological objects, most importantly, the concept that they are evolutionarily related. Only tree-reconstruction methods fully use the evolutionary information and generate a tree-like structure in which the order of branching is expected to reflect evolutionary events (Felsenstein, 1996). This representation, if possible, would undoubtedly be the best way to view relationships between biological objects and the task of their grouping.

Evolutionary trees are difficult to reconstruct reliably. The reliability drops fast with the degree of divergence between objects and depends drastically on the amount of information used for the tree building (Felsenstein, 1996; Saitou, 1996). For example, tree reconstruction is hardly possible for short (100 amino acids or less) protein sequences sharing 5–15% identity. This has not posed a serious problem in the past, since it was challenging to detect homology and to align sequences with similar to random identity. Recently, with the rapid expansion of protein sequence data and the development of sensitive profile similarity search tools such as PSI-BLAST (Altschul *et al.*, 1997) and HMMer (Eddy, 1996), researchers have been

[*]To whom correspondence should be addressed.

able to extend the limits of sequence-based homology detection (Aravind and Koonin, 1999). Thus the task of understanding relationships in divergent protein families is of particular importance. To complicate the problem even further, alignments of these divergent sequences are often restricted to a few moderately conserved but confidently aligned motifs (Henikoff *et al.*, 2000). This reduces the number of positions for tree reconstruction and thus decreases tree reliability statistics.

The notion of 'sequence space' has been widely used in the literature (Higgins, 1992; Agrafiotis, 1997; Holm and Sander, 1997; Holm, 1998; Forster *et al.*, 1999; Yona and Levitt, 2000). Generally, it is an abstract metric space in which each sequence is represented as a point and distances between points reflect divergence between corresponding sequences. If known, such space can be used to study the relationships between proteins and, in particular, to group them. However, provided a set of biological sequences, it is not straightforward to define the metric and thus to map sequences onto the 'sequence space' (Forster *et al.*, 1999; Yona and Levitt, 2000). To obtain results with evolutionary meaning, it would also be desirable to combine tree-reconstruction procedures with the space mapping.

Here we developed a classification approach that combines evolutionary distance calculation with the Euclidian space mapping. This approach is outlined as follows. (I) Evolutionary distances are estimated from protein sequences (Zuckerkandl and Pauling, 1965; Dayhoff *et al.*, 1978; Grishin, 1995; Felsenstein, 1996; Zhang and Gu, 1998; Grishin *et al.*, 2000) or 3D structures (Grishin, 1997). The distances are defined as expected numbers of amino acid substitutions per site and are calculated from the sequence similarity scores or structure RMSD values using standard correction formulas (Tajima and Takezaki, 1994; Grishin, 1995, 1997; Zhang and Gu, 1998). If the user is studying objects other than proteins, a distance matrix should be provided. (II) Each object (protein) is represented as a point in a multidimensional Euclidian space in such a way that Euclidian distances d_{ij} between the points optimally approximate the estimated distances D_{ij} between the objects: $\sum_{ij}(d_{ij}^2 - D_{ij}^2)^2/D_{ij}^4 = \min$. Notably, such a solution always exists, even if the distance matrix is not metric due to statistical errors in distance estimates or for other reasons. Our approach will find the best approximation of the given distance matrix with Euclidian distances. The user can visualize the results in 2D or 3D by plotting the projections of the multidimensional space. (III) The points corresponding to biological objects are grouped according to the newly designed model-based clustering procedure. This developed methodology is generic and can be applied to any set of objects with a defined distance measure. However, the grouping procedure is statistically meaningful for biological objects with evolutionary connections between them.

ALGORITHM

Similarity measures: from multiple sequence alignment to similarity scores

Standard amino acid similarity matrices of PAM (Dayhoff *et al.*, 1978) or BLOSUM (Henikoff and Henikoff, 1992) series can be used to calculate pairwise scores between the aligned sequences. Let $\boldsymbol{s}$ be an amino acid similarity matrix with elements $s(a, b)$—scores for a match between amino acids a and b, let $\boldsymbol{A}$ be an alignment of n sequences, A_{ik} is a symbol (amino acid or gap: '-') in the site k of the sequence i. For each pair of sequences i and j from $\boldsymbol{A}$ we calculate the following quantities:

$$\text{(I)}\quad S_{ij} = \sum_{k\in K_{ij}} s(A_{ik}, A_{jk})/l(K_{ij}),$$

$$T_{ij} = 0.5 \sum_{k\in K_{ij}} (s(A_{ik}, A_{ik}) + s(A_{jk}, A_{jk}))/l(K_{ij}),$$

where K_{ij} is the set of sites k such that $A_{ik} \neq$'-' and $A_{jk} \neq$'-' (sites in which neither i nor j has a gap) and $l(K_{ij})$ is number of elements in K_{ij}. S_{ij} is called score per site. T_{ij} is the average upper limit of the score per site achieved with identical sequences.

$$\text{(II)}\qquad S_{ij}^{\text{rand}} = \sum_{a=1}^{20}\sum_{b=1}^{20} f_j^i(a) f_i^j(b) s(a, b),$$

where $f_j^i(a)$ is a frequency of amino acid 'a' in ith protein sequence of $\boldsymbol{A}$ over all sites in K_{ij}. S_{ij}^{rand} has a meaning of a score per site expected from random sequences of amino acid composition characteristic of sequences i and j.

The normalized score (Feng and Doolittle, 1997) per site V_{ij} between sequences i and j is then calculated as follows:

$$V_{ij} = \frac{S_{ij} - S_{ij}^{\text{rand}}}{T_{ij} - S_{ij}^{\text{rand}}} \tag{1}$$

This normalized score range is expected to be from 0 (for random sequences) to 1 (for identical sequences). However, for very divergent sequences, V_{ij} could become negative (score for the two aligned sequences is smaller that the score for the two random sequences) due to statistical errors.

Distance measures: from similarity scores to evolutionary distances

Evolutionary distance between two homologous proteins is defined as an expected number of amino acid substitutions per site on the evolutionary path between them (Felsenstein, 1996). Distance defined this way is a metric, however, what we are able to compute from the aligned sequences is an estimate of this distance. Such estimates contain statistical errors, which may cause violation of triangle inequality.

A considerable amount of work has been done to develop estimates of evolutionary distances from identity fraction or, more precisely, from normalized identity fraction U_{ij} calculated by equations similar to Equation (1), where identity matrix is used as a score matrix $\boldsymbol{s}$ (Zuckerkandl and Pauling, 1965; Dayhoff *et al.*, 1978; Holmquist *et al.*, 1983; Ota and Nei, 1994; Tajima and Takezaki, 1994; Grishin, 1995; Li and Gu, 1996; Grishin, 1997; Zhang and Gu, 1998; Grishin *et al.*, 2000). Since there is no developed theory on how to convert general similarity scores to distances (Feng and Doolittle, 1997), we first convert normalized similarity scores V_{ij} to normalized identity fractions $\overline{U_{ij}}$ and then estimate distances from $\overline{U_{ij}}$. $\overline{U_{ij}}$ is calculated from V_{ij} the following way. (I) We find V_{ij} using the matrix $\boldsymbol{s}$, and U_{ij} using identity matrix for all pairs i and j. (II) We find least-squares best-fit coefficient β in the function $u = (1-\beta)v + \beta v^2$ to approximate the set of pairs (V_{ij}, U_{ij}). (III) We calculate $\overline{U_{ij}} = (1-\beta)V_{ij} + \beta V_{ij}^2$. It is not clear why a second order polynomial gives an excellent fit to the data, but $\overline{U_{ij}}$ obtained with the outlined procedure are expected to be more accurate than U_{ij}. Sites with non-identical amino acids are ignored (scored 0) when U_{ij} is calculated. However, fine differences between mismatches are taken into account in calculation of V_{ij}, thus affecting $\overline{U_{ij}}$ and increasing its accuracy. In the event of $\overline{U_{ij}}$ being non-positive, $\overline{U_{ij}}$ is set to the minimal $\overline{U_{ik}}$ or $\overline{U_{jk}}$ over all k, or 0.05, whichever is greater. The set of $\overline{U_{ij}}$ is used to estimate evolutionary distances D_{ij} with one of the following standard formulas: (I) $D_{ij} = -\ln \overline{U_{ij}}$; (II) $D_{ij} = 1/\overline{U_{ij}} - 1$; (III) $D_{ij} = \vartheta(\overline{U_{ij}})$, where $y = \vartheta(x)$ is the function inverse to $x = \ln(1+2y)/(2y)$. These formulas are derived through consideration of amino acid substitions as a Markov process and differ by underlying assumptions about the variability of substitution rates among sites and amino acids. The formula (I) defines a Poisson distance and is derived under the assumption of equal substitution rates between different sites and different amino acids (Zuckerkandl and Pauling, 1965). This estimate is close to the lower limit for the distance, since any difference in substition rates among sites will result in larger distances (Grishin, 1995, 1997). Poisson distance possesses rather small statistical error. Geometric distance is computed by the formula (II). Geometric distance calculation assumes that substitution rates are distributed exponentially across sites (Uzzell and Corbin, 1971; Holmquist *et al.*, 1983; Grishin, 1995). Since proteins are known to have variation of substitution rates over sites (Uzzell and Corbin, 1971; Feng and Doolittle, 1997), geometric distances are more realistic than Poisson distances, however, their estimates possess larger errors. And, finally, logarithmic distance (III) is derived under the assumption that average rates are distributed exponentially across sites and the rates of different amino acid replacements are approximated by a uniform distribution over some interval (Grishin, 1995; Feng and Doolittle, 1997). This appears to be the most realistic of the distance estimates, but its statistical error is relatively large.

Euclidian space: representation of evolutionary distances

Evolutionary distances D_{ij} between each pair of n proteins were estimated on the previous step. The goal here is to find points $p_i, \ldots, p_n$ in a Euclidian space of some dimensions such that Euclidian distances between the points $d_{ij} = \text{distance}\{p_i, p_j\}$ closely approximate corresponding evolutionary distances D_{ij}. Our approach to this problem differs from a standard multidimensional scaling technique (MDS, see monograph by Borg and Groenen, 1997). We select the following function for minimization:

$$g(\mathbf{p}) = \sum_{i<j}^{n} w_{ij} \left(d_{ij}^2(\mathbf{p}) - D_{ij}^2 \right)^2 \tag{2}$$

where weight coefficients w_{ij} are equal to $1/D_{ij}^4$, instead of the so-called 'stress function' $st(\mathbf{p}) = \sum_{i<j}^{n} w_{ij}(d_{ij}(\mathbf{p}) - D_{ij})^2$ used in MDS. Each term in our sum g is $\left(d_{ij} - D_{ij}\right)^2 \left(d_{ij} + D_{ij}\right)^2 / D_{ij}^4$ thus differing from the corresponding term of stress function st with weight coefficients $w_{ij} = 1/D_{ij}^2$ by a factor $\left(1 + d_{ij}/D_{ij}\right)^2$. If $d_{ij} \approx D_{ij}$ for all i and j, then our function g differs from the stress function st only by a coefficient $\approx$4. If $d_{ij} > D_{ij}$ for some i, j, then the term in g is larger than the corresponding term in st making g more sensitive to large deviations between d_{ij} and D_{ij}. Thus minimization of g is aimed at eliminating large differences first.

The function g has smooth derivatives and therefore we can avoid a technique of iterative majorization used for stress minimization (Borg and Groenen, 1997), which involves solution of a large linear system with n^2m^2 unknown variables, where m is the dimensionality of Euclidian space (e.g. if $n \approx 100 \quad m > 10$ then we have $>1\,000\,000$ unknowns). The function g is a polynomial of degree 4 with nm variables and its local minimum is easily computable. Therefore we use the classical gradient method. Theoretically, a local minimum of g can be iteratively found starting from random points $p_i, \ldots, p_n$. This procedure converges slowly and is not practical. To obtain results within acceptable time, the starting points may be chosen not at random, but closer to the ultimate solution. Potentially we could take a point given by classical Torgerson–Gower scaling (Borg and Groenen, 1997) as a starting point. However, truncation of negative eigenvalues, the number of which is large in our case, may produce a distance matrix d very different from D and a lengthy minimization procedure cannot be avoided. Thus we find starting points, i.e. the first approximation to the solution, using the following embedding procedure.

We place points p_i in a Euclidian space one by one. After adding each point, we minimize g allowing only this point to move, but do not change the coordinates of others. It is possible to proceed through an arbitrary order of proteins. Our trials show that RMS error per point of Euclidian representation of a distance matrix $e = \sqrt{2g/(n(n-1))}$ varies little with embedding order (several percent). However, the following ordering gives slightly better results. The first protein in the ordering is the protein for which the sum of evolutionary distances from it to other proteins is minimal. It is the most 'central' protein in our set. Each successive protein is the one with the minimal sum of evolutionary distances up to the already chosen proteins. The technical details of the embedding procedure are given in Appendix 1 (see supplementary data or ftp://iole.swmed.edu/pub/EESG/Appendix1.doc). Using this procedure we find coordinates for all points $p_i, \ldots, p_n$. If the number of dimensions of Euclidian space increases for each application of the above procedure, then we get $(n-1)$-dimensional space for the points and point coordinates form a triangle matrix with zeros on the diagonal and above. In general, the coordinate matrix satisfies equalities $p_{ij} = 0$, for $i \leqslant j \leqslant N$, where N is the number of dimensions in Euclidian space. Our embedding program provides an option to set up the maximal number of dimensions of Euclidian space for embedding. For example, it is possible to search for n points on a line or on a plane.

The coordinate matrix $\boldsymbol{p}$ of points obtained on the embedding step is the first approximation to start minimization of Equation (2) by an iterative gradient method. We monitor the results of this procedure, called optimization, by recording a relative error $e = \sqrt{2g/(n(n-1))}$. Because the optimization is slow, the user specifies the number of iterations. After this number is reached, the program displays relative error e in percent. The user then decides whether to continue optimization. In our tests with protein alignments of 30–60 sequences, the relative error e after the embedding step was 6–16%, which reduces by 1.5–3% after 1000–2000 iterations of the optimization program. The embedding program runs very fast (several seconds for Pentium III 500 MHz). However, the optimization program is slow (several minutes for 2000 steps).

Grouping: from Euclidian coordinates to groups

Let $\boldsymbol{p}$ be an $n \times m$-matrix of coordinates for the n points $p_1, \ldots, p_n$ in an m-dimensional space that was found on the previous step, and σ is a given positive constant. For each point p_i, we define a multinormal probability density function:

$$\varphi_i^{(\sigma)}(x_1, \ldots, x_m) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^m} \times \exp\left(-\frac{1}{2\sigma^2}\left((x_1 - p_{i,1})^2 + (x_m - p_{i,m})^2\right)\right)$$

We consider the following function:

$$\Phi_\sigma(x_1, \ldots, x_m) = \frac{1}{n}\sum_{i=1}^{n} \varphi_i^{(\sigma)}(x_1, \ldots, x_m)$$

It is clear that for large σ ($\sigma > \max|p_i - p_j|$) Φ_σ has a single maximum. If σ is small ($\sigma \ll \min_{i,j}|p_i - p_j|$), then Φ_σ has n maxima. There exist the limits $\sigma_{\max}$ and $\sigma_{\min}$ such that for σ within these limits ($\sigma_{\min} < \sigma < \sigma_{\max}$), the number of maxima of Φ_σ is between 1 and n. The grouping program finds approximate values for these limits and divides the interval $[\sigma_{\min}, \sigma_{\max}]$ into a default of 200 segments: $\sigma_1 = \sigma_{\max}$, $\sigma_{k+1} = q \cdot \sigma_k$, where $q = (\sigma_{\min}/\sigma_{\max})^{1/200}$. Let $\sigma = \sigma_k$. For each point p_i, $(i = 1, 2, \ldots, n)$, the program finds the local maximum of Φ_σ by gradient method starting from this point (See Appendix 2 as supplementary data or at ftp://iole.swmed.edu/pub/EESG/Appendix2.doc) for the procedure description). As a result of this procedure, all points are collected into groups by the property of having the same local maximum. More precisely, two points p_i and p_j belong to one group iff the local maxima for these points are equal. Each σ_k defines a set S_k of groups. The number of groups in the set S_k is equal to the number of maxima of the function Φ_σ for $\sigma = \sigma_k$. The grouping program forms sets S_k for $k = 1, 2, \ldots, N$ and stops when the number of groups in a set S_k reaches a number N specified by the user. Consider the case where a set S arises at σ_{k_1}, does not change while $k < k_2$, and changes at σ_{k_2}. Then we say that lifetime of the set S is $k_2 - k_1$. The grouping program finds the most stable set—the set with the longest lifetime. We treat this set as the most probable grouping. The program also generates a table with groups highlighted in different color for all σ_k (see Figure 4). This table shows relationships between groups and sequences and may be used to generate a grouping tree. The grouping program runs fast. For example, a grouping process for 54 values of σ and 36 points in 14-dimensional space takes about 20 s on Pentium III–500 MHz, i.e. about 100 maxima/s.

RATIONALE

The proposed classification methodology can be applied to any set of objects with distances defined between them. However, we expect that most frequently it will be used to deduce the relationships in homologous families of proteins. Homologs are defined as biological objects with common ancestry. According to the classic definition of Fitch (1970), a relationship in a pair of homologs is described by orthology or paralogy. Orthologs are proteins in different organisms that evolved from the same protein in the last common ancestor of these organisms. Orthologs become different proteins simply because species split in evolution. Thus orthologs represent the same 'species' ('type') of a protein. Paralogs are different proteins that originated through a gene duplication that happened in an organism. Thus paralogs correspond to different 'species' ('types') of proteins. As a rule, orthologs retain the same function and evolve under the constraints imposed by this function. Therefore orthologs tend to be closer to each other in sequence and structure than to any of the paralogs (Tatusov *et al.*, 1997). Paralogs frequently have a different function and may be quite distant from each other in their sequences and structures.

Typically, a protein family of homologs consists of several groups of orthologs (Tatusov *et al.*, 1997). Consider one group that corresponds to the set of orthologs from different organisms. We represent each organism (i.e. each orthologous protein) by a point in Euclidian space. The distances between the points correspond to evolutionary divergence between sequences. If the sampling of organisms is random and independent, then the distribution of points in space is likely to be well approximated by a Gaussian. The maximum of that distribution corresponds to the most typical sequence of the group. Since the sampling of organisms differs from random, the resulting distribution may deviate from Gaussian. Thus we have chosen a non-parametric approach to estimate the probability density from the points (Simonoff, 1996). Each point generates a Gaussian density in the Euclidian space with the mean being the coordinates of this point and the given variance σ^2, which is set to the same value for all points. The normalized sum of these Gaussians is an estimate of the density of points in a Euclidian space.

Consider several groups of orthologs, which is the typical picture for a protein family. These sequences are expected to fall into several clusters (groups) in Euclidian space. Within each group, the density of sequences may be approximated by a Gaussian. Relationship between groups is not expected to follow any laws, except that the groups are likely to be separated. Thus a non-parametric estimate of the density of points should have several maxima corresponding to the most populated regions. These maxima can be viewed as the centers of groups and objects around the centers should be assigned to the same group. Our grouping procedure attributes each object to its local maximum, which represents one of the groups.

Real cases are likely to deviate from the ideal scenario described above. Samples of orthologs may not be representative or evolutionary rates in some clades may deviate substantially from average. These effects will result in splitting of orthologs into several groups. Alternatively, some paralogs may be very close in their sequences and will be grouped together. Thus caution is advised in interpreting the results. However, despite potential problems with interpretation, our procedure will outline groups of proteins that are closer to each other in terms of evolutionary distances than to other proteins.

The results of grouping depend dramatically on the value of σ used to generate probability density estimates. Indeed, if this value is large, the resulting density will have a single maximum and all points will be grouped together. Alternatively, if σ is very small, each point will be in a separate group. Some of the intermediate σ values should result in a biologically reasonable grouping. In our experiments with artificially generated points or sequences that fall into discrete groups, we found out that expected grouping is stable over longer intervals of σ values. Therefore we hypothesize that if some grouping is preserved for longer σ intervals, it is more likely to be biologically meaningful. Thus we probe a range of σ values. We start from a large σ that leads to grouping of all points together in a single group and perform grouping at smaller σ values calculated in multiplicative increments until the number of groups reaches the specified number. Then the results are analyzed and the grouping that is maintained for the longest interval of σ value is selected. Additionally, we find individual groups that are preserved for longer σ intervals. It is expected that these groups correspond to better-defined and tight clusters.

Generally, we believe that the ultimate decision about the preferred σ value of grouping should be left to the users, since they are expected to know the details about the objects that may go beyond the simple distance measures used in our approach. The combination of computation (objective function) with manual inspection (expert knowledge) typically leads to better results and we offer a versatile tool for it.

RESULTS AND DISCUSSION

Here we illustrate and discuss the performance of the method and compare the results to those produced by the two most popular biological applications: single linkage clustering and UPGMA. Two types of tests are performed. First, we test our grouping strategy on artificially generated points in Euclidian space. Second, we apply the method to several protein families, for

Table 1. Grouping of points from Figure 1 by different methods

	$d = 0.9$			$d = 1.0$			$d = 1.1$			$d = 1.2$		
δ	SG	SLC	UPG	SG	SLC	UPG	SG	SLC	UPG	SG	SLC	UPG
0.05	100	0	1	100	4	25	100	95	49	100	100	64
0.10	98	0	5	100	2	17	100	35	41	100	83	58
0.15	69	1	4	94	5	23	100	20	35	100	60	54
0.20	68	2	13	85	2	19	96	10	31	100	36	49

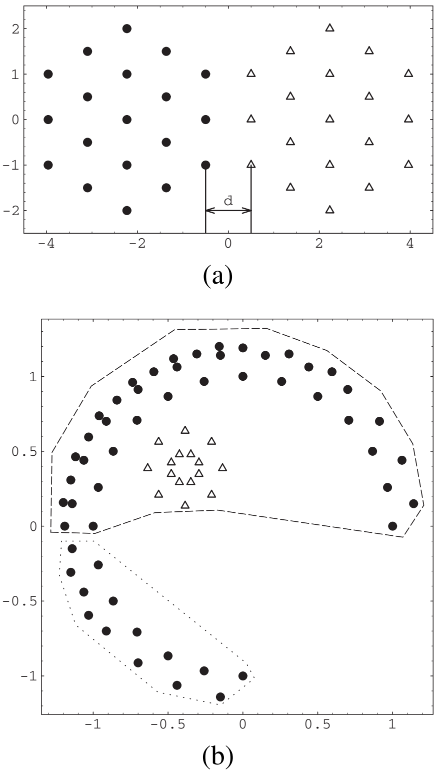
The numbers of correct groupings in 100 tests are shown for different distances d between hexagons and random deviations $-\delta < x_i < \delta$, $-\delta < y_i < \delta$ from the points on Figure 1. Correct grouping corresponds to the segregation of points that belong to two hexagons of Figure 1. SG is our sigma grouping method, SLC is single linkage clustering and UPG is UPGMA method.

which we estimate evolutionary distances from protein sequences.

Grouping of artificially generated points

Cohesion and separation. The first example describes the groups with high cohesion without clear separation (Everitt *et al.*, 2001, Figure 1a). The two groups of 19 points each fill two hexagonal areas. The distances between neighboring points is equal to 1. The separation between the groups is d. We show that our sigma grouping method (SG) outperforms single-linkage clustering (SLC) and UPGMA. Sigma grouping separates the clusters for $d \geqslant 0.75$. Thus even if the distance between the groups is slightly smaller than the distance between the points within the groups, sigma grouping is able to separate these cohesive groups. SLC and UPGMA require $d > 1$ and $d > 1.1$ respectively[†]. Additionally, grouping solutions found by SLC and UPGMA depend strongly upon small deviations in positions of points from the regular lattice shown on Figure 1a. The SG method has proven to be very robust in this case. We generated small uniformly distributed random deviations (x_i, y_i) where $i = 1, 2, \ldots, 38$ and $-\delta < x_i < \delta$, $-\delta < y_i < \delta$, and added vectors (x_i, y_i) to the points from Figure 1a. The procedure was repeated 100 times for different δ and the results are shown in Table 1. It is clear that SG outperforms SLC and UPGMA in all cases.

The second example deals with well-separated groups one of which has low cohesion (Everitt *et al.*, 2001, Figure 1b). Points in the first groups are arranged to fill a circle. The second group has low cohesion and the points are arranged in a crescent semi-enclosing the first group. The groups are well separated and thus SLC recovers the groups. The SG method also outlines the two groups correctly, however, UPGMA clusters the small group together with the part of the large group (Figure 1b).



(a)

(b)

Fig. 1. Cohesion and separation. Configurations of points used to illustrate two groups of high cohesion poor separation (a) and low cohesion (crescent group) clear separation (b). Points in one group are shown as filled circles, points in the other group are shown as triangles. Dashed lines in (b) encircle the two groups as found by the UPGMA method.

Three groups, equal spread of points. To test our grouping method on a more realistic example, we generated three groups of points g_1, g_2, and g_3 using two-dimensional Gaussian distributions. The means of the three distributions form an equilateral triangle with sides equal to 4.5 and have coordinates (0, 0), (4.5, 0), (2.25, 3.8971). Covariance matrix of each distribution is equal to identity matrix. We randomly generate 30 points in each group using these Gaussians (Figure 2) and applied three

[†] For SLC and UPGMA we selected a configuration that contains 2 groups.

Table 2. Grouping of 90 points generated by three gaussians

	SG	SLC	UPG
Average	2.56	11.83	6.61
Standard deviation	1.9	7.39	6.67

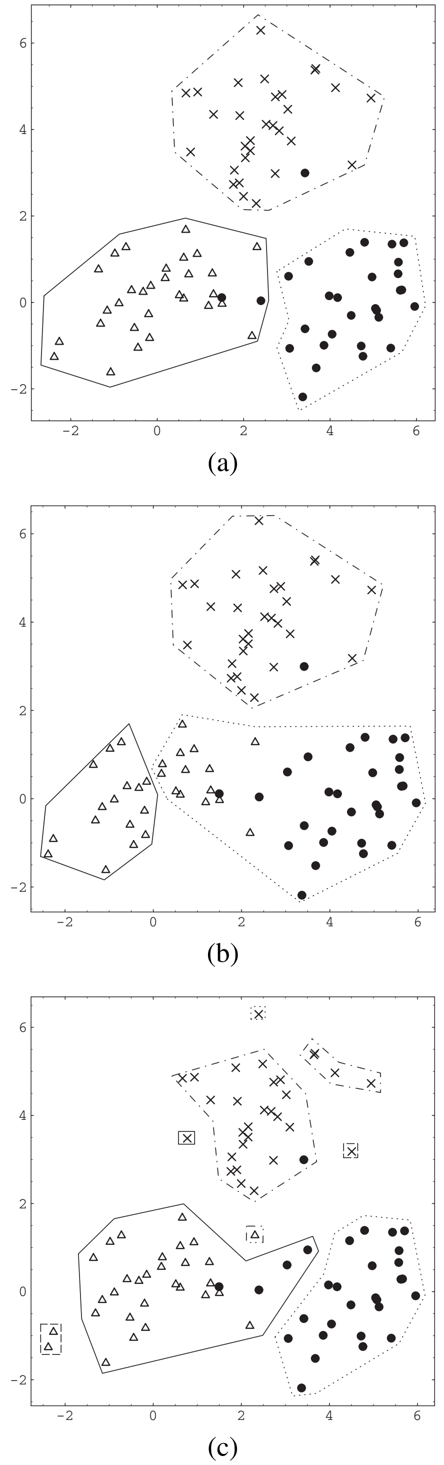
The numbers indicate the number of incorrectly placed points by each grouping method. SG is our sigma grouping method, SLC is single linkage clustering and UPG is UPGMA method.

grouping methods (SG, UPGMA and SL) to the resulting arrangement of points.

The SG method attributes 3 points incorrectly, i.e. not to the group that contains the majority of points generated by a given Gaussian (Figure 2a, black circles grouped with triangles and crosses). Such an error is easily understandable since these points fell closer to other groups. UPGMA splits the group of triangles, erroneously placing 14 of them in the group of black circles (Figure 2b). This example shows that pairing of the closest points in UPGMA method may lead to splitting of a group. Since the three groups are not very well separated, the results of SLC are much worse. The SLC configuration of 3 groups links all but two points into one group. We also found the SLC configuration that contains the smallest number of erroneously placed points (Figure 2c). Such configuration divides the set into 9 groups with 15 incorrectly placed points.

We generated random points by three Gaussians 100 times and repeated the grouping experiment on each configuration. The statistics of the number of incorrectly grouped points are shown in Table 2. For the SLC method we took the configuration with minimal number of incorrectly placed points. Theoretically, it is not possible to get the mean number of incorrectly placed points less than 2.0466 since some points will penetrate into some other group, like the two black circles inside the triangles in Figure 2. This example also illustrates our choice to select the grouping as a non-trivial grouping that stays the same over the largest interval of σ values. We get the three groups (Figure 2a) for a large range of σ values: $0.55224 < \sigma < 1.40381$ (the points were generated with the variance equal to 1). The centers of the groups identified by our program for $\sigma = 1$ are (0.251, 0.322), (4.805, 0.02), (2.33, 3.82), which is close to the means of gaussians used to generate the point.

Two tight groups and a spreadout group, robustness to noise. Groups with a different spread of points are potentially difficult for many clustering methods. To test the SG method in this case we generated three groups of 30 points each by three two-dimensional Gaussians with the means (0, 0), (5, 0), (5, 2.2) and variances 1, 0.09, 0.09, respec-



(a)

(b)

(c)

Fig. 2. Three groups, equal spread of points. One of the random configurations for the three groups generated by the three independent Gaussian distributions with the means (4.5, 0), (2.25, 3.8971) and unity variances. 30 points were generated in each group. The points in different groups are shown by different symbols: black circles, triangles and crosses. The groupings produced by SG (a), UPGMA (b) and SLC (c) are outlined by lines.

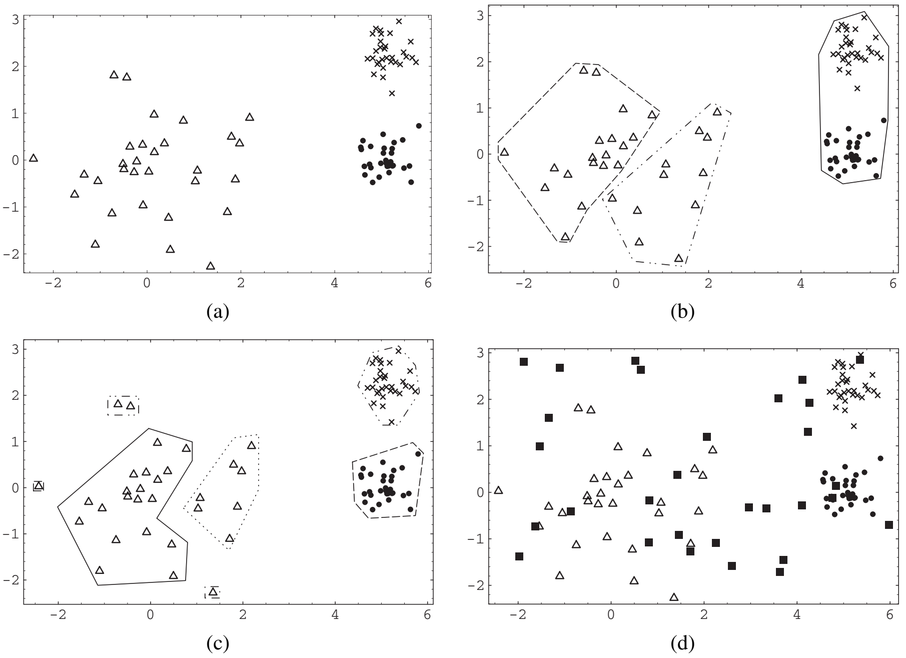


Fig. 3. Two tight groups and a spreadout group, robustness to noise. One of the random configurations generated by the three independent Gaussian distributions with the means (0, 0), (5, 0), (5, 2.2) and variances 1, 0.09, 0.09, respectively. 30 points were generated in each group. The points in different groups are shown by different symbols: black circles, triangles and crosses. SG method identifies the groups correctly (a). The groupings produced by UPGMA (b) and SLC (c) are outlined by lines. 30 random points shown as black squares are added to the configuration of three groups (d).

tively. Correlation coefficients for all distributions were 0 (Figure 3a). SG identifies the grouping correctly as a most stable configuration over large range of σ values. UPGMA cannot handle the differences in spread: the method splits the spreadout group and unifies two tight groups (Figure 3b). The SLC result with three groups makes a group that contains a single point. The SLC configuration that contains the least number of incorrectly placed points is shown in Figure 3c (7 groups, 11 incorrect points).

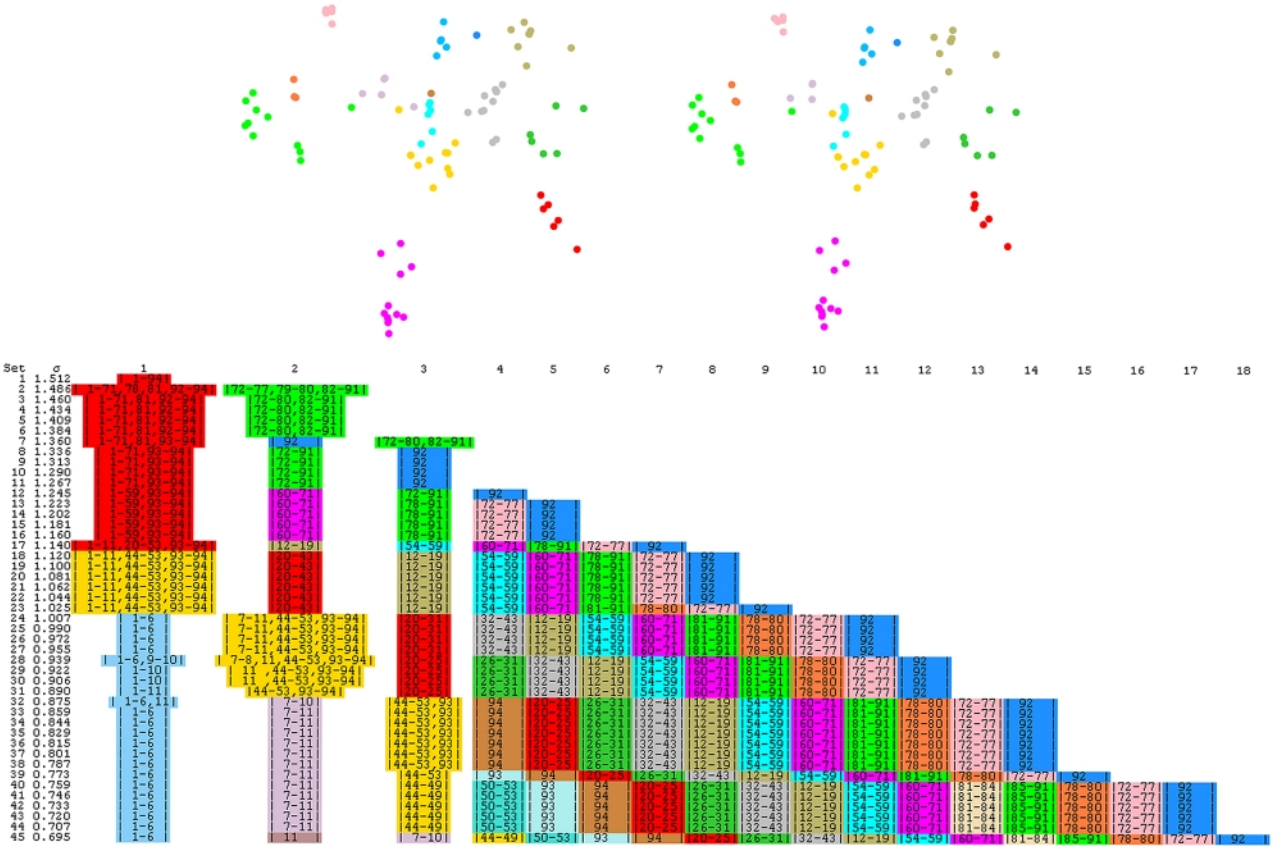
Using the arrangement of points from Figure 3a we test the robustness of SG to noise introduced by points added randomly. We generated 30 points randomly and uniformly distributed in the range $\{(-2, 3), (6, -2)\}$ and added them to the 90 points that were generated with three Gaussians (Figure 3d). The most stable configuration given by SG method splits the set into three groups and attributes all the points generated by the Gaussians to the correct grouping in 82 out of 100 repetitions of the experiment on adding random points. UPGMA and SLC did not give a single correct grouping in these 100 repetitions.

Euclidian space mapping and grouping of protein families

We apply our method to sequence alignments of proteins and illustrate its performance on two examples discussed in the literature.

Sm proteins. Sm proteins participate in pre-mRNA splicing by promoting small nuclear RNA cap modification and targeting small nuclear ribonucleoproteins to specific locations (Seraphin, 1995). Recent analysis of Sm sequences revealed that they can be grouped in at least seven subfamilies (Salgado-Garrido *et al.*, 1999; Wicker *et al.*, 2001). We applied SG method to the alignment from Wicker *et al.* (2001)[‡] and compared the results to those obtained by Wicker *et al.* (2001). The Sm proteins were numbered consecutively and the key to the numbers appears in the legend to Figure 4. The most stable configuration (sets from 33 to 38 on Figure 4, σ values from 0.859 to 0.773) contains 14 groups: {1–6} {7–11} {12–19} {20–25} {26–31} {32–43} {44–53, 93} {54–59} {60–71} {72–77} {78–80} {81–91} {92} {94}. The grouping suggested by Wicker *et al.* (2001) using our sequence numbers is: {1–11} {12–19} {20–31} {32–43} {44–59} {60–71} {72–80} {81–94}. The group {1–11} of Wicker *et al.* is a sum of the first two groups in our grouping. The group {1–11} is found by our approach as well, but at a larger σ value (set 31, Figure 4). Wicker *et al.* group {20–31} is the sum of our groups

[‡] Alignment from Wicker *et al.* (2001) contains several proteins with identical amino acid sequences. We removed these identical sequences prior to application of the SG method reducing the number of sequences from 101 to 94.



Set	σ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.512	1-94																	
2	1.486	1-71,78,81,92-94	72-77,79-80,82-91																
3	1.460	1-71,81,92-94	72-80,82-91																
4	1.434	1-71,81,92-94	72-80,82-91																
5	1.409	1-71,81,92-94	72-80,82-91																
6	1.384	1-71,81,92-94	72-80,82-91																
7	1.360	1-71,81,93-94	92	72-80,82-91															
8	1.336	1-71,93-94	72-91	92															
9	1.313	1-71,93-94	72-91	92															
10	1.290	1-71,93-94	72-91	92															
11	1.267	1-71,93-94	72-91	92															
12	1.245	1-59,93-94	60-71	72-91	92														
13	1.223	1-59,93-94	60-71	78-91	72-77	92													
14	1.202	1-59,93-94	60-71	78-91	72-77	92													
15	1.181	1-59,93-94	60-71	78-91	72-77	92													
16	1.160	1-59,93-94	60-71	78-91	72-77	92													
17	1.140	1-11,20-53,93-94	12-19	54-59	60-71	78-91	72-77	92											
18	1.120	1-11,44-53,93-94	20-43	12-19	54-59	60-71	78-91	72-77	92										
19	1.100	1-11,44-53,93-94	20-43	12-19	54-59	60-71	78-91	72-77	92										
20	1.081	1-11,44-53,93-94	20-43	12-19	54-59	60-71	78-91	72-77	92										
21	1.062	1-11,44-53,93-94	20-43	12-19	54-59	60-71	78-91	72-77	92										
22	1.044	1-11,44-53,93-94	20-43	12-19	54-59	60-71	78-91	72-77	92										
23	1.025	1-11,44-53,93-94	20-43	12-19	54-59	60-71	81-91	78-80	72-77	92									
24	1.007	1-6	7-11,44-53,93-94	20-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92							
25	0.990	1-6	7-11,44-53,93-94	20-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92							
26	0.972	1-6	7-11,44-53,93-94	20-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92							
27	0.955	1-6	7-11,44-53,93-94	20-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92							
28	0.939	1-6,9-10	7-8,11,44-53,93-94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92						
29	0.922	1-10	11 ,44-53,93-94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92						
30	0.906	1-10	11 ,44-53,93-94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92						
31	0.890	1-11	44-53,93-94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92						
32	0.875	1-6,11	7-10	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
33	0.859	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
34	0.844	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
35	0.829	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
36	0.815	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
37	0.801	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
38	0.787	1-6	7-11	44-53,93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92				
39	0.773	1-6	7-11	44-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-91	78-80	72-77	92			
40	0.759	1-6	7-11	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92	
41	0.746	1-6	7-11	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92	
42	0.733	1-6	7-11	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92	
43	0.720	1-6	7-11	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92	
44	0.707	1-6	7-11	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92	
45	0.695	1-6	11	7-10	44-49	50-53	93	94	20-25	26-31	32-43	12-19	54-59	60-71	81-84	85-91	78-80	72-77	92

Fig. 4. Euclidian space mapping and grouping of protein families exemplified by Sm proteins. The multiple sequence alignment was taken from Wicker *et al.* (2001). The sequences are numbered as follows: 1. ySmE; 2. riSmEa; 3. riSmEb; 4. arSmE; 5. huSmE; 6. caSmE; 7. huLsm5; 8. q9vrt7; 9. yLsm5; 10. o42978; 11. globu2; 12. huSmN; 13. oSmB; 14. chSmB; 15. dSmB; 16. caSmB; 17. arSmB; 18. ySmB; 19. sSmB; 20. riLsm1; 21. huLsm1; 22. yLsm1; 23. q20229; 24. yb18-schpo; 25. aaf46688; 26. aad56232; 27. aaf47567; 28. riSmx9; 29. aaf23841; 30. o74483; 31. ySmx13; 32. riSmGa; 33. riSmGb; 34. arSmG; 35. alSmG; 36. huSmG; 37. caSmG; 38. ySmG; 39. sSmG; 40. bLsm7; 41. riLsm7; 42. huLsm7; 43. yLsm7; 44. sulfo; 45. globu1; 46. pyroc1; 47. p-abys; 48. metha1; 49. aero-pern1; 50. riLsm3; 51. huLsm3; 52. q9y7m4; 53. yLsm3; 54. huSmD2; 55. caSmD2; 56. arSmD2; 57. sSmD2; 58. ySmD2; 59. pfalSmD; 60. huSmF; 61. dSmF; 62. riSmF; 63. bSmF; 64. caSmF; 65. ySmF; 66. arSmF; 67. sSmF; 68. nLsm6; 69. huLsm6; 70. yLsm6; 71. cab54975; 72. ySmD1; 73. sSmD1; 74. huSmD1; 75. riSmD1a; 76. arSmD1; 77. caSmD1; 78. yLsm2; 79. mLsm2; 80. amphSm; 81. yLsm4; 82. caLsm4; 83. huLsm4; 84. arLsm4; 85. ySmD3; 86. sSmD3; 87. arSmD3; 88. riSmD3; 89. huSmD3; 90. dSmD3; 91. caSmD3; 92. yLsm9; 93. aero-pern2; 94. m-therm2. The three dimensional projection of the multidimensional Euclidian space is shown on top. Proteins are shown as circles. Groupings at different steps defined by different σ-values are shown at the bottom. The grouping on the projections corresponds to the most stable configuration: sets 33–38. Colors of groups are the same in the projection and in the table.

{20–25} and {26–31}, which appeared after the group {20–31} split at $\sigma = 0.995$ (set 27, Figure 4). The group {20–31} splits at a larger σ value than the group {1–11} (0.995 versus 0.890). Thus the two parts {20–25} and {26–31} of the group {20–31} are further from each other in Euclidian space than the two parts {1–6} and {7–11} of the group {1–11}. Wicker *et al.* group {44–59} is the sum of our groups {54–59} and {44–53,93} after removing the protein 93. The protein 93 splits from our group on the step 39 (Figure 4). From the projection of the Euclidian space into three dimensions (Figure 4) it is clear that the group of yellow points contains one point that is further from the rest. This point stands for the protein 93. Wicker *et al.* group {72–80} is the sum of our groups {72–77} and {78–80}. Wicker *et al.* group {81–94} is the sum of our groups {81–91}, {92}, {94} and a protein 93. Our method stresses the isolation of proteins 92 and 94, 92 in particular, which forms a separate group starting from the set 7 (Figure 4). The remaining three Wicker *et al.* groups are exactly the same as in our most stable configuration. Thus there is a good correspondence between the grouping obtained by Wicker *et al.* and SG results. A general trend is that Wicker *et al.* groups are larger than SG groups and combine several groups from the most stable configuration found by our method.

γGCS/GS proteins. γ-Glultamylcysteine synthetase (γGCS) and glutamine synthase (GS) have recently been shown to be homologous. These enzymes perform ATP-dependent ligation of amine/ammonia to γ-carboxyl of glutamate. SG method has been applied to the alignment of 39 diverse sequences of this family taken from Abbott *et al.* (2001, see supplementary data or ftp://iole.swmed.edu/pub/EESG/GCS.pdf for the results). The most stable configuration persists over a wide range of σ-values ($0.671 \leqslant \sigma \leqslant 0.968$, steps 34–56) and contains 8 groups: {1–5} {6–12} {13} {14–18} {19–28} {29–32} {33–36} {37–39} (see ftp://iole.swmed.edu/pub/EESG/GCS.pdf). Multiple lines of evidence support this grouping as biologically reasonable with the exception that the protein 13 should be placed in the same group with the proteins {6–12} (Abbott *et al.*, 2001). Such a configuration is found by SG method but not as the most stable one (steps 28–33). Protein 13 is the most distant protein in this group, which is reflected in a three-dimensional projection of the Euclidian space (yellow protein in ftp://iole.swmed.edu/pub/EESG/GCS.pdf). The projection also reveals a correlation between the life span of the group and closeness of the points in space. For instance, the group of proteins {19–28} represents a visually tight cluster and is the most long-lived group (63 steps out of 65). Abbott *et al.* (2001) predicted the structure of γGCS by homology with GS. The link between the two families of proteins was found by sequence analysis using PSI-BLAST (Altschul *et al.*, 1997; Aravind and Koonin, 1999). The three dimensional projection of the Euclidian space rationalizes the link between the two families. γGCS family protein sequences (pink, red, rosy and yellow, proteins 1–18) were used to find GS family proteins (green, olive, cyan, proteins 29–36) (Abbott *et al.*, 2001). The first GS protein that was found in the searches was the protein 32, which is indeed the closest one to the γGCS family sequences (proteins 1–18) in the 3D projection (ftp://iole.swmed.edu/pub/EESG/GCS.pdf).

CONCLUSIONS

We developed a novel approach to visualization of relationships between biological objects and to their clustering that is based on the Euclidian space mapping. We have shown that our model-based grouping approach outperforms UPGMA and single linkage clustering, algorithms commonly used in biology, for the cases when the groups possess unusual cohesion and separation or display different spreads of points. Our method is robust to noise caused by adding random points or by random deviations in positions of points. The projections of the Euclidian space onto 2 or 3 dimensions can be used to visualize relationships between sequences and to rationalize transitive sequence search strategies for remote homolog detection.

ACKNOWLEDGEMENTS

We are grateful to James Wrabl for critical reading of the manuscript and helpful comments and to Lisa Kinch, Hong Zhang and Jimin Pei for evaluating the strategy and applying it to protein families they have studied.

REFERENCES

Abbott,J.J., Pei,J., Ford,J.L., Qi,Y., Grishin,V.N., Pitcher,L.A., Phillips,M.A. and Grishin,N.V. (2001) Structure prediction and active site analysis of the metal binding determinants in gamma-glutamylcysteine synthetase. *J. Biol. Chem.*, **276**, 42099–42107.

Agrafiotis,D.K. (1997) A new method for analyzing protein sequence relationships based on Sammon maps. *Protein Sci.*, **6**, 287–293.

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Aravind,L. and Koonin,E.V. (1999) Gleaning non-trivial structural, functional and evolutionary information about proteins by iterative database searches. *J. Mol. Biol.*, **287**, 1023–1040.

Borg,I. and Groenen,P. (1997) *Modern Multidimensional Scaling*, Springer Series in Statistics, Springer, New York.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1978) A model of evolutionary change in proteins. In Dayhoff,M.O (ed.), *Atlas of Protein Sequences and Structures*, Vol. 5, Suppl. 3, National Biomedical Research Foundation, Washington, DC, pp. 345–352.

Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.

Everitt,B.S., Landau,S. and Leese,M. (2001) *Cluster Analysis*. Arnold, London.

Felsenstein,J. (1996) Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Methods Enzymol.*, **266**, 418–427.

Feng,D.F. and Doolittle,R.F. (1997) Converting amino acid alignment scores into measures of evolutionary time: a simulation study of various relationships. *J. Mol. Evol.*, **44**, 361–370.

Fitch,W. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–106.

Forster,M., Heath,A. and Afzal,M. (1999) Application of distance geometry to 3D visualization of sequence relationships. *Bioinformatics*, **15**, 89–90.

Grishin,N.V. (1995) Estimation of the number of amino acids substitutions per site when the substitution rate varies among sites. *J. Mol. Evol.*, **41**, 675–679.

Grishin,N.V. (1997) Estimation of evolutionary distances from protein spatial structures. *J. Mol. Evol.*, **45**, 359–369.

Grishin,N.V., Wolf,Y.I. and Koonin,E.V. (2000) From complete genomes to measures of substitution rate variability within and between proteins. *Genome Res.*, **10**, 991–1000.

Henikoff,J.G., Pietrokovski,S., McCallum,C.M. and Henikoff,S.

(2000) Blocks-based methods for detecting protein homology. *Electrophoresis*, **21**, 1700–1706.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Higgins,D.G. (1992) Sequence ordinations: a multivariate analysis approach to analysing large sequence data sets. *Comput. Appl. Biosci.*, **8**, 15–22.

Holm,L. (1998) Unification of protein families. *Curr. Opin. Struct. Biol.*, **8**, 372–379.

Holm,L. and Sander,C. (1997) New structure–novel fold? *Structure*, **5**, 165–171.

Holmquist,R., Goodman,M., Conroy,T. and Czelusniak,J. (1983) The spatial distribution of fixed mutations within genes coding for proteins. *J. Mol. Evol.*, **19**, 437–448.

Li,W.H. and Gu,X. (1996) Estimating evolutionary distances between DNA sequences. *Methods Enzymol.*, **266**, 449–459.

Ota,T. and Nei,M. (1994) Estimation of the number of amino acid substitutions per site when the substitution rate varies among sites. *J. Mol. Evol.*, **38**, 642–643.

Podani,J. (2000) *Inroduction to the Exploration of Multivariate Biological Data*. Backhuys, Leiden.

Saitou,N. (1996) Reconstruction of gene trees from sequence data. *Methods Enzymol.*, **266**, 427–449.

Salgado-Garrido,J., Bragado-Nilsson,E., Kandels-Lewis,S. and Seraphin,B. (1999) Sm and Sm-like proteins assemble in two related complexes of deep evolutionary origin. *Embo J.*, **18**, 3451–3462.

Seraphin,B. (1995) Sm and Sm-like proteins belong to a large family: identification of proteins of the U6 as well as the U1, U2, U4 and U5 snRNPs. *Embo J.*, **14**, 2089–2098.

Simonoff,J.S. (1996) *Smoothing Methods in Statistics*, Springer Series in Statistics, Springer, New York.

Tajima,F. and Takezaki,N. (1994) Estimation of evolutionary distance for reconstructing molecular phylogenetic trees. *Mol. Biol. Evol.*, **11**, 278–286.

Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–637.

Uzzell,T. and Corbin,K.W. (1971) Fitting discrete probability distributions to evolutionary events. *Science*, **172**, 1089–1096.

Wicker,N., Perrin,G.R., Thierry,J.C. and Poch,O. (2001) Secator: a program for inferring protein subfamilies from phylogenetic trees. *Mol. Biol. Evol.*, **18**, 1435–1441.

Yona,G. and Levitt,M. (2000) Towards a complete map of the protein space based on a unified sequence and structure analysis of all known proteins. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **8**, 395–406.

Zhang,J. and Gu,X. (1998) Correlation between the substitution rate and rate variation among sites in protein evolution. *Genetics*, **149**, 1615–1625.

Zuckerkandl,E. and Pauling,L. (1965) Molecules as documents of evolutionary history. *J. Theor. Biol.*, **8**, 357–366.