BIOINFORMATICS

Vol. 18 no. 7 2002
Pages 908–921

Clustering of proximal sequence space for the identification of protein families

*Federico Abascal and Alfonso Valencia**

Protein Design Group, National Centre for Biotechnology, CNB-CSIC, Cantoblanco, Madrid E-28049, Spain

Received on June 29, 2001; revised on August 22, 2001; accepted on August 22, 2001

ABSTRACT

Motivation: The study of sequence space, and the deciphering of the structure of protein families and subfamilies, has up to now been required for work in comparative genomics and for the prediction of protein function. With the emergence of structural proteomics projects, it is becoming increasingly important to be able to select protein targets for structural studies that will appropriately cover the space of protein sequences, functions and genomic distribution.

These problems are the motivation for the development of methods for clustering protein sequences and building families of potentially orthologous sequences, such as those proposed here.

Results: First we developed a clustering strategy (Ncut algorithm) capable of forming groups of related sequences by assessing their pairwise relationships. The results presented for the ras super-family of proteins are similar to those produced by other clustering methods, but without the need for clustering the full sequence space. The Ncut clusters are then used as the input to a process of reconstruction of groups with equilibrated genomic composition formed by closely-related sequences. The results of applying this technique to the data set used in the construction of the COG database are very similar to those derived by the human experts responsible for this database.

Availability: The analysis of different systems, including the COG equivalent 21 genomes are available at http://www.pdg.cnb.uam.es/GenoClustering.html

Contact: valencia@cnb.uam.es

INTRODUCTION

Deciphering the function of proteins is the main task in molecular biology and biochemistry. The increasing flow of genomic information makes the contribution of bioinformatics very relevant for proposing possible new protein functions based on prior information. Common function prediction methods are based on the assumption of an evolutionary relationship between similar proteins (Zuckerkandl and Pauling, 1965), that will share a common three-dimensional structure (Chothia and Lesk, 1986), and will preserve a certain functional relationship. In this sense it is important to consider protein sequences in the context of the protein families in which they have evolved.

The relationships between protein sequences and protein families are better described with terminology imported from the fields of paleontology and evolutionary biology. The basic, but difficult, concept of homology (http://www.britannica.com/) can be qualified in terms of orthology and paralogy (Fitch, 1970). Orthologues are those genes that have evolved in different species from a common ancestral gene. Paralogues are cases in which the similarity is due to a duplication event inside a given genome, with specialization events having broken the direct relationship.

In terms of function, the implication is that orthologous genes tend to conserve the same function, while paralogous genes acquire specialized functions. Therefore, determination of possible orthologous relationships is a fundamental step in the prediction of protein function (see for example Bork and Koonin, 1998).

Concept of sequence space

Differentiating between paralogous and orthologous genes is not always easy, and different events add additional complications to this situation, e.g. loss of one of the orthologous sequences whose function is replaced by an unrelated sequence (non-orthologous gene displacement; Koonin *et al.*, 1996).

Orthologous relationships cannot be derived from simple observation of pairwise similarities, since it is always possible that other, closer-related sequences are the true orthologues. Therefore, information from related sequences, including the analysis of complete genomes, is required for a complete assessment of orthologous relationships. In the extreme case, accurate assessment of orthologous relationships requires detailed phylogenetic analysis of the precise relationships between corresponding protein families in the presence of complete genome information.

*To whom correspondence should be addressed.

In practice, phylogenetic analysis of complete families is often replaced by double-checking pairwise similarities. The idea is that if sequence A from one organism is that which is most similar to sequence B from a second organism, and sequence B has A as its closest-related sequence in the first organism, then A and B are probably orthologous. Even if this assertion will be true in most cases if all sequences from both organisms are known, this simple analysis cannot replace a complete phylogenetic analysis of the corresponding protein family.

An intermediate approach consists of automatically classifying sequences into groups of related sequences. This approach is more comprehensive than pairwise comparison but less demanding than detailed phylogenetic analysis of each protein family. The methodology presented here belongs to this intermediate type of approach, of which the two closest-related previous incarnations are ProtoMap and GeneRAGE.

ProtoMap (Yona *et al.*, 1999) describes the relationships between sequences in terms of graph theory, in which the nodes of the graph are the sequences, the arcs connecting the nodes represent the similarity relations, and the weights associated with each arc correspond to the levels of similarity. ProtoMap implements an accretion algorithm, that works by first dividing the entire graph into small sectors containing only very closely related sequences, and then forms larger clusters by merging neighbouring clusters, by decreasing the similarity cutoff required. The different cutoffs provide alternative views of the sequence space at different granularity levels. The biological interpretation is that the different levels of clustering are related to the structure of protein superfamilies and families.

GeneRAGE (Enright and Ouzounis, 2000) is based on a single linkage clustering algorithm, that is, relationships between sequences are represented by a binary yes/no flag, and do not take into account the strength of the relationships. GeneRAGE incorporates a procedure for solving proteins in domains by clustering proteins in all groups to which their domains belong, avoiding the incorrect inclusion of sequences connected by indirect sharing of domains in the same cluster.

Here we propose an alternative approach for clustering sequence space. In contrast to other approaches, our procedure focuses on the space surrounding a given protein by analyzing only local sequence relationships, instead of clustering the full sequence space in the search for distant relationships and a complete overview of sequence space. Consequently this new approach is less computationally expensive than clustering the full sequence space, and is applicable to individual sequences without requiring complete genomes.

After obtaining clusters of related sequences, the second round of algorithms attempts to merge clusters to obtain a representation of species, that can make them equivalent to groups of orthologous sequences, with an aim similar to the COGs approach (Tatusov *et al.*, 1997). For this second round of algorithms complete genomes are required.

COGs are based on the concept of double-checking of pairwise similarities (see above). It involves a first step of collapsing obvious paralogues, followed by identification of sets of at least three related sequences amongst the best hits. These groups of three sequences (triangles) are merged if they have two sequences (vertices) in common. The resulting constructions of connected sequences are inspected visually for the detection of problematic multi-domain proteins and groups of paralogous sequences. Therefore, while COGs are produced by a combination of automatic and manual procedures, the algorithm proposed here is fully automatic and works locally on the sequence space.

In the following text we first discuss the implementation of our clustering procedure, based on the 'Normalized Cut' algorithm, (Shi and Malik, 1997; graph flow theory; Minieka, 1978), based on the pairwise relations obtained through recursive similarity searches (intermediate sequence searches; Park *et al.*, 1997; Gerstein, 1998). As an example we describe the results for the case of the *ras* family, which are then compared with the groups obtained by the ProtoMap clustering system.

In the second part we discuss the performance of a strategy for optimizing the previously-generated clusters in the search for groups of orthologous sequences in the presence of genomic information. To test the quality of these classifications we generated clusters for a set of 486 *Mycoplasma genitalium* sequences with information from twenty-one complete genomes as reference sequence space. This scenario is complex enough to be realistic but sufficiently small to be comparable with the COGs classification.

METHODS

Data set

The first part of the analysis was carried out by searching within a database composed of SwissProt sequences (June 2001, version 39.20; Bairoch and Apweiler, 2000).

The second part was carried out with a set of twenty-one complete genomes analyzed in the October 2000 version of COGs. This includes:

Sixteen bacteria genomes: *Aquifex aeolicus* (letter code: Q, number of proteins: 1526); *Thermotoga maritima* (V, 1852); *Synechocystis* (C, 3168); *Escherichia coli* (E, 4292); *Bacillus subtilis* (B, 4122); *Mycobacterium tuberculosis* (R, 3924); *Haemophilus influenzae* (H, 1694); *Helicobacter pylori* (U, 1577); *H. pylori J99* (J, 1492); *M. genitalium* (MG, 468); *Mycoplasma pneumoniae* (MP, 678); *Borrelia burgdorferi* (O, 1256); *Treponema*

pallidum (L, 1033); *Chlamydia trachomatis* (I, 895); *Chlamydia pneumoniae* (N, 1053); *Rickettsia prowazekii* (X, 834).

Four archea genomes: *Archaeoglobus fulgidus* (A, 2411); *Methanococcus jannaschii* (M, 1747); *Methanobacterium thermoautotrophicum* (T, 1871); *Pyrococcus horikoshii* (K, 2072).

One eukarya genome: *Saccharomyces cerevisiae* (Y, 5932).

These twenty-one genomes are classified into sixteen phylogenetic lineages, as closely-related pairs are considered as a single lineage: E–H, U–J, MG–MP, I–N, and L–O.

Mapping the sequence space with similarity measurements

To obtain a set of similar sequences for each of the analyzed genes we use an intermediate sequence search procedure of concatenated similarity searches (Park *et al.*, 1997; Gerstein, 1998) with BLAST (Altschul *et al.*, 1997). This procedure is based on the transitivity principle: if protein A is related to B, and B is related to C, then protein A will be related to C. This correspondence is often complicated by the presence of multi-domain proteins and partial alignments that can create artificial links between unrelated proteins through intermediates that contain several domains. For each BLAST round we use only the aligned fragments from previous rounds. This process, when applied carefully, reduces the number of cases in which multi-domain proteins create artifacts.

In detail, we start with a BLAST search, for which all the aligned sub-sequences are extracted (and not the complete protein corresponding to the aligned fragments). In a next round BLAST searches are initiated with those sub-sequences. In the following rounds the same procedure is followed, and specific care is taken to avoid particular problems. For example, slightly different fragments of the same protein are extracted from searches carried out with different members of the family. In this case there are two possibilities, if the fragments overlap, we select the larger overlapping fragment as representative of all the hits, but if the fragments do not overlap we choose the set of overlapping fragments with a higher score, this score being the sum of BLAST $-\log 10$ (E-value) for each fragment. Additional problems are encountered in some alignments that tend to include unrelated sequences in the extremes of the alignments. To alleviate this problem we prune the alignments until at least the 20% positions are matched by sequences.

The BLAST version we used was 2.1.2 (November, 2000) with default parameters except that we set -v 2000, -b 2000, -F 'S; C' and the E-value cutoff, that varies in the two experiments.

The BLAST scores were used as the basic measurement of pairwise similarity for the clustering procedure (see below). Other methods such as PSI-BLAST (Altschul *et al.*, 1997) could have been used to find more connections in the sequence space, but they were less adequate for the extraction of pairwise similarities.

For the example of the *ras* family, the BLAST searches were initiated with the RASH_HUMAN protein (H-ras from Human) using SwissProt as the reference database. Four rounds of searches were carried out with an E-value cutoff of 1e−07. Sequences were filtered with SEG and COILS filters (Wootton and Federhen, 1993; Lupas *et al.*, 1991). A minimum of thirty-five amino acids was required in order to accept sequence matches. After the first round of BLAST searches, the new sequences were used to build sub-sequences (the fragments of the sequences aligned during the BLAST searches) with all the different matches.

For the example of the MG sequences, we used the BLAST searches already available at the COG web site (ftp://ncbi.nlm.nih.gov/pub/COG/old99/pa) to facilitate posterior comparison of the results (see below). The strategy for selecting the sequence relations is equivalent to concatenating three rounds of searches with a BLAST E-value cutoff of 1e−05. The process starts by selecting similar sequences for a query protein from the BLAST results. These sequences are used for a second round of BLAST searches, from where additional similar sequences are extracted, sequences that in turn are used in identifying new potential relationships during a third round of inspection of the BLAST runs. The resulting relationships are used as the basis for the clustering procedure, whose results are then compared with the COG groups. It is important to mention that problematic multi-domain proteins are split into their domains by the COG data managers, so it is possible to use the BLAST results directly without selecting sub-sequences from the alignments. To make a fair comparison we only count clusters with more than three lineages, as in the construction of the COG. The general idea of the recursive search process is represented in Figure 1.

Comparison of the clustering results with the COGs data

Here we present the results of applying this procedure to each one of the *M. genitalium* sequences, and subsequent comparison of the resulting clusters with the COGs groups that include each of the MG sequences.

For comparison of the composition of the clusters, we establish two definitions: 'coherent clusters' are those cases where a cluster generated by one of the methods is included in (forms a subset of) a cluster generated by another method; and 'identical clusters' are those coherent clusters that contain the same elements (sequences), allowing a variation of plus or minus one sequence.

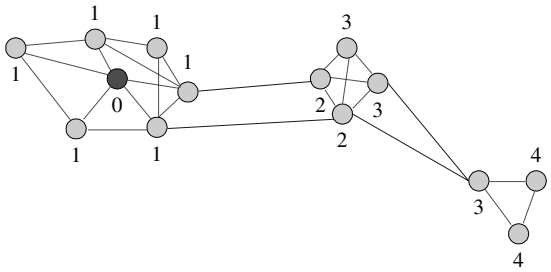


Fig. 1. The recursive similarity searches and graphical representation of the results. Sequences and their similarities (above a given cutoff value) are represented as circles and lines connecting them. The numbers indicate the round of BLAST searches in which the sequence was identified for the first time. Following the numbers and connections it is possible to reproduce the process of sequence identification through an intermediate sequence search process.

Graph based representation of the sequence space

We represent the relationship between sequences as a graph G(V, E), where the nodes (V) correspond to sequences and the arcs (E) to the similarity relationships between them. Arcs have an associated label or weight (w) that corresponds to the pairwise sequence similarities, expressed as $-\log 10$ (E-value). E-values larger than 1e−05 were disregarded in the case of the MG sequences, and 1e−07 in the case of the *ras* proteins.

The clustering algorithm implemented here works with undirected graphs, with no directionality in the arcs. To adapt the sequence relationships to this asymmetry, the weights of the arcs are made equal to the mean of the two directed arc weights (A compared to B, and B compared to A). In the cases in which only one of the two relationships exists, this is considered as the weight of the undirected arc.

Ncut clustering strategy

The clustering method uses the normalized cut algorithm, which is derived from the more general minimum cut algorithm. The minimum cut algorithm (Wu and Leahy, 1993) is based on the measures of capacity and flow of a graph. A cut (A, B) in a graph G(V, E) is a partition of V into two distinct sets of nodes A and B. The capacity of a cut is the sum of all weights associated with the arcs that cross the cut, that is:

$$\text{cut(A, B)} = \text{Sum } w(i, j);\ i \text{ in A},\ j \text{ in B}$$

the minimum cut being that with the minimum associated capacity.

This algorithm has a clear preference for cutting small groups of nodes as previously observed by Shi and Malik (1997), who then propose a version of the algorithm called the normalized cut algorithm (Ncut), where:

$$\text{Ncut(A, B)} = \text{cut(A, B)}/\text{asso(A, } V) + \text{cut(A, B)}/\text{asso(B, } V)$$

where asso(A, V) is the sum of the weights of the arcs from all nodes in A to all nodes in V (including those in A), thereby weighting the capacity of the cut by the level of disconnection it induces in the graph, so avoiding the preference for small groups. To find the minimum Ncut, the Ncut (A, B) equation is treated in a standard eigenvector system (Press *et al.*, 1992; and public MESCHACH numerical library, http://www.math.uiowa.edu/∼dstewart/meschach/).

Recursivity and stop conditions

The clustering algorithm works recursively: once the cut is found its convenience is evaluated, and if it is appropriate then a new cut is sought for each of the resulting subgraphs.

The clustering continues until neither of the following two conditions holds:

(1) the arithmetic mean of the capacity of the relationships inside the children clusters exceeds (according to a relative threshold) the value of the same measure between the children clusters. We set the relative threshold to two-fold (×2) and four-fold (×4);

(2) the number of relationships existing inside any of the two children clusters divided by the number of possible relationships is higher than the same measure in the parent cluster.

Mean capacity and connectivity

We define two parameters to evaluate the clustering process:

Mean capacity: the arithmetic mean of the similarity relationhips between two clusters. For example, a mean capacity of 10 corresponds to E-values between the sequences in the two clusters of around 1e−10.

Connectivity: the number of connections between the two clusters divided by the number of possible connections that could exist if all nodes were connected. (Note that 'connectivity' is used here in a different sense to its common usage in graph theory.)

Further merges of the sequence clusters

We tested two procedures for merging the clusters generated by the ×2 and ×4 clustering procedures. The purpose of merging the clusters is to find an optimal distribution of sequences that reflects the separation of orthologous sequences.

Under the join6 procedure, the nearest neighbouring cluster will be merged if the mean capacity is larger than 6.

The results are called $\times 2$ join6 and $\times 4$ join6, depending on the origin of the clusters ($\times 2$ and $\times 4$ clustering above).

Relative entropy-based reconstruction uses the information available on neighbouring clusters to evaluate the possibility of merging these depending on their potential content in paralogous and orthologous sequences.

The parameter used to avoid inclusion of paralogous sequences is the relative entropy value, defined as:

$$\text{Srel} = \text{H}(P \| Q) = \text{Sum}\, i(P(xi) \log(P(xi)/Q(xi)))$$

where $P(xi)$ is derived from the observed frequency of genome i in the set of accepted clusters (see below), and $Q(xi)$ is the frequency in the expected distribution (see below). Srel describes the similarity of the two distributions, in this case the observed distribution and an expected reference distribution. In the present analysis, two types of expected distribution wer e used:

S_var expected distribution. Here the number of genes contributed by each genome to the groups is expected to be proportional to the number of genes existing in that genome, so $Q(xi)$ will be ni/nt, where ni is the number of genes in genome i, and nt is the total number of genes in all genomes.

S_one expected distribution. This corresponds to groups formed with one gene from each genome. $Q(xi) = 1/N$, with N being the number of genomes (i.e. $1/21$ here).

The merging algorithm

We have used a recursive algorithm for merging clusters progressively, from the closest-related to those that are more distant, with the procedure being as follows:

(0) Initialize the set of accepted clusters to the cluster containing the seed gene.

(1) Calculate the relative entropy of the set of accepted clusters.

(2) Find the cluster that has the highest connectivity with those in the set of accepted clusters. If two clusters have the same connectivity, the one with the higher mean capacity is selected.

(3) Calculate the relative entropy that would be obtained if the two clusters were merged.

(4) If the relative entropy decreases with the addition of the cluster, the clusters are merged and we return to step 1. If the entropy value increases, the process stops.

RESULTS

This results section is divided into two parts. In the first part we analyze the results of searching with a seed gene in a complete sequence database and then clustering the resulting sequences. These results are compared with the ProtoMap results, which are based on classification of the full sequence space.

In the second part we analyze the clustering of a set of twenty-one genomes around MG sequences. The results of various different procedures are presented: initial clustering using the Ncut procedure with $\times 2$ and $\times 4$ cutoffs, and subsequent merging of these initial clusters using the join6 and relative entropy algorithms. The results are then compared with the corresponding COGs groups of orthologous sequences.

Clustering around RasH in a populated sequence space background

After building the local map to find sequences around RASH_HUMAN (see Section **Methods**), Ncut produces clusters corresponding to: ras/ral, ran, gem/rad, rab, rab7, rac/rho, ran, arf, sar and G-alpha sequences (Figure 2). The relationship of the Ras subfamily with the rab, rab7, and ran groups is strong, while the relationship with rho is more distant. The groups and the relationships correspond well with what is known about the family (Valencia *et al.*, 1991; Garcia-Ranea and Valencia, 1998), and with the two main ProtoMap groups at a cutoff level of 1e−00, one with Ras and the other with arf, sar and G-alpha sequences.

Some of the differences are interesting, for example the rab7 group, which is clearly different from the other rabs, forms a different cluster but does not form a different ProtoMap group.

It is also interesting to note that in this case the groups obtained by Ncut clustering give an adequate view not only of the superfamily but also of the relationships between the different families.

Distribution of cluster sizes in the twenty-one genome set

The distribution of the cluster sizes obtained via independent searches using each one of the MG sequences provides a useful first overview of the efficiency of the method. In the space formed by the sequences of the twenty-one genomes, the best-represented families are expected to correspond more or less to one sequence from each one of the genomes (families of orthologues). Therefore, the presence of clusters with twenty-one sequences probably indicates the presence of complete families, while smaller clusters correspond to families that are not represented in all the genomes analyzed (e.g. bacteria-only sequences), or incorrect identifications of relationships between groups of sequences that are distant members of the same family. Larger groups correspond to families in which the separation of orthologous and paralogous sequences was not achieved.

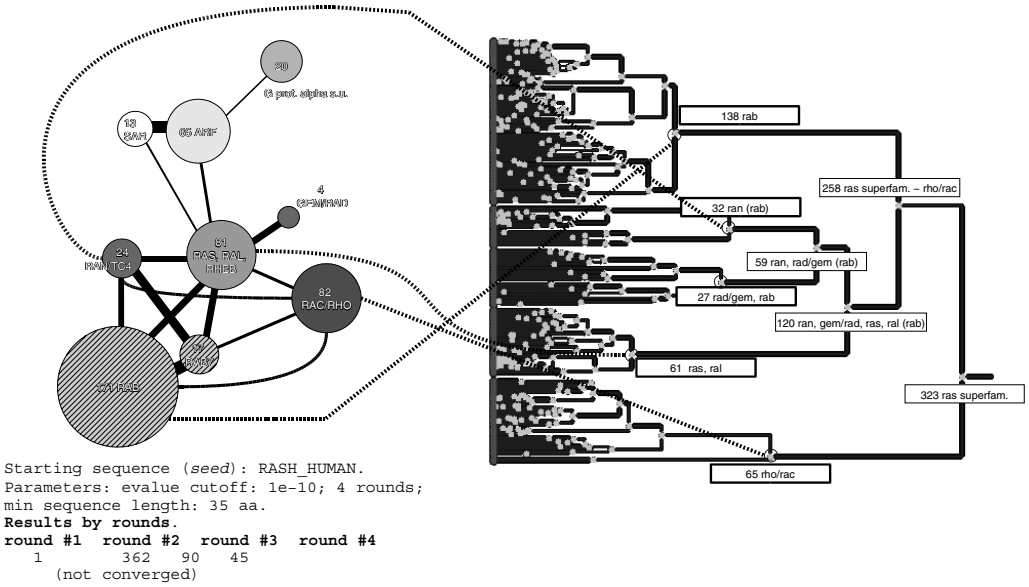


Fig. 2. Clusters generated starting with the human Hras (RASH_HUMAN) sequence. On the left-hand side are shown the Ncut results of the recursive sequence search begun with RASH_HUMAN in the SwissProt database. Only clusters with more than three sequences are represented, with 21 additional sequences forming 18 small clusters that mainly correspond to fragments and odd sequences. The information corresponding to all the clusters can be retrieved at http://www.pdg.cnb.uam.es/fabascal/RAS/P01112.faa.Cft.ncut.html. On the right-hand side the clusters are compared with the tree representation of the ProtoMap clusters. At the upper level of ProtoMap (cutoff 1e−0), rash_human forms clusters with all the sequences from the *ras* superfamily (ran, rab, rho…); the proteins ARF, SAR and G-alpha are located in a neighbouring cluster with 177 sequences, representing only the part of the known G-alpha sequences that were included in the four recursive BLAST searches. Neither the G-alpha, nor the SAR and ARF sequences are included in the ProtoMap tree represented in the figure. There is good agreement between the cluster structure and the ProtoMap groups, as related to the different SwissProt versions used (clustering was carried out with a SwissProt version containing 97 586 sequences, and ProtoMap with a previous SwissProt version (35) with 72 623 sequences).

Cluster size distribution obtained with the Ncut procedure (×2 and ×4 thresholds)

The Ncut algorithm (Figure 3, panels a and b) produces small clusters with no clear representation of the optimal 21-sequence class. Clusters with 2–3 sequences, often composed of similar sequences from the closely-related MG and MP genomes, are relatively abundant. Even if their proximal clusters contain related sequences, the Ncut algorithm fails to classify them together. For example, cluster MG092 contains two ribosomal S18 proteins from MG and MP, and of two neighbouring clusters, one contains the orthologous gene from *R. prowazekii*, and the other the thirteen remaining bacterial orthologues and an additional duplicated sequence of *M. tuberculosis*. The less similar *S. cerevisiae* orthologous gene does not appear in the initial sequence searches (*E*-value of 0.0004, i.e. above our cutoff). In this case the Ncut clustering result reflects the bias in sequence space created by the presence of similar sequences in this sparse representation of genomes.

Cluster MG241 represents the opposite case, where the clustering procedure correctly identifies the uniqueness of two Mycoplasma sequences, that together with two other paralogues in a neighbouring cluster seem to be a Mycoplasma-specific family.

Most of the other clusters with less than 21 sequences correspond to groups of closely-related sequences, such as the clusters with 16 bacteria-specific genes. For example, cluster MG073 obtained with the Ncut ×4 procedure contains 18 ‘excinuclease ABC subunit B’ genes that do not have orthologues in archea or yeast, with the exception of a *M. thermoautotrophicum* sequence that may have been acquired by horizontal gene transfer, and a duplicated gene fragment of *T. maritima*.

The Ncut ×2 procedure generates 40 clusters that contain 16–17 sequences, and of these 29 (72.5%) of them include no duplicated genes, while 37 (92.5%) of them contain less than four. Similarly, out of the 56 clusters of this size produced with the Ncut ×4 procedure, 44 of them (78.6%) have no duplication and as many as 51 (91.07%) have three or less. This indicates that they correspond to clusters composed basically of a single sequence from each one of the bacterial genomes.

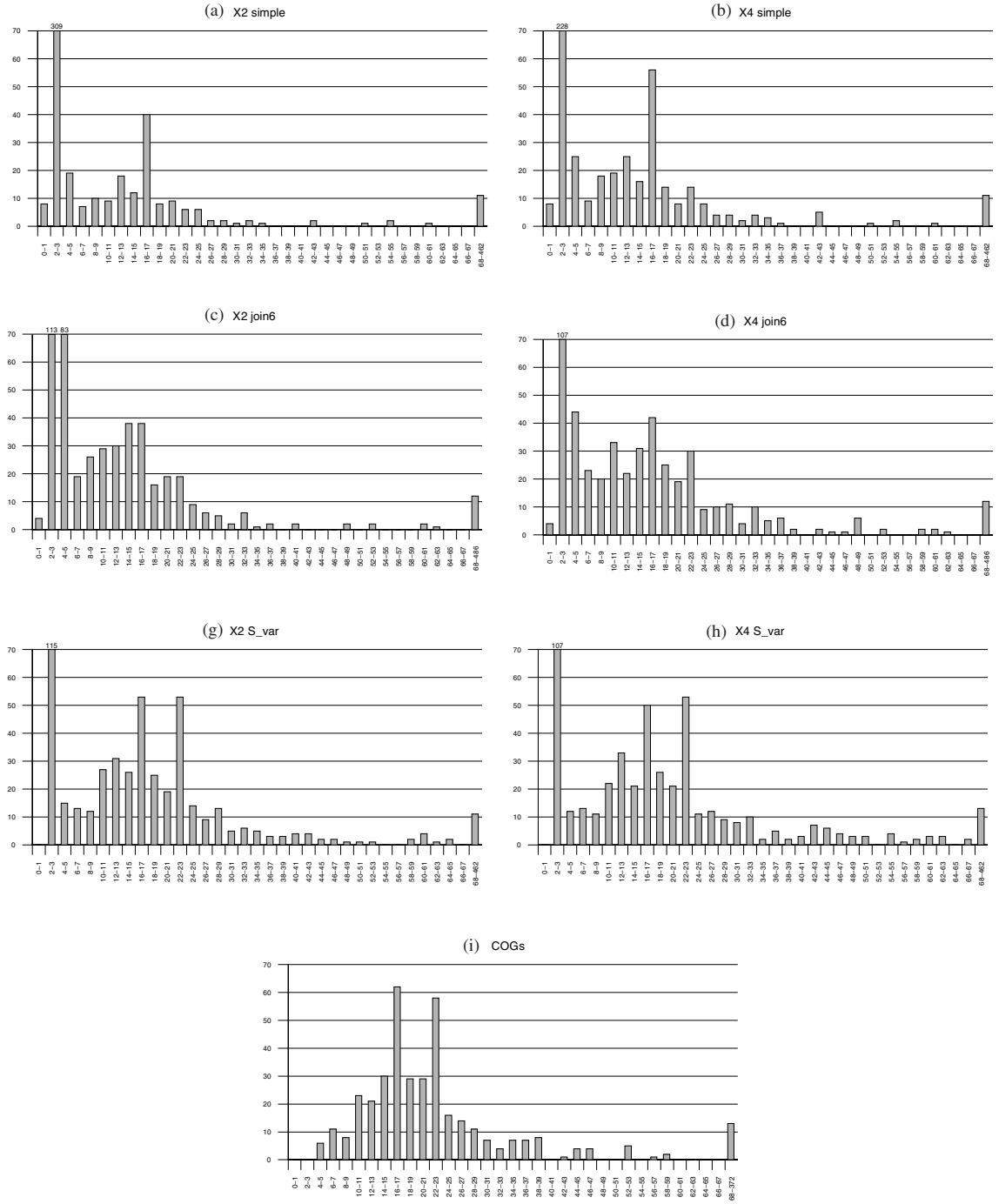


Fig. 3. Distribution of the size of the clusters obtained with each of the *M. genitalium* genes. The results of different clustering and merging procedures are presented. Panels (a) and (b) for the Ncut clustering procedure, (c) and (d) for the results of merging clusters with the join6 procedure, and (e) and (f) for the S_var merging algorithm. In panel (i), the size of the COGs groups is shown (in this case COGs are only produced for groups with more than three lineages).

The Ncut clustering produces a small number of clusters with more than 21 sequences. Most of these correspond to erroneous mixtures of genes, for example cluster MG345 contains 43 sequences of Isoleucyl and Valyl tRNA synthetases. It is difficult to separate these sequences by direct sequence analysis methods based on pairwise

comparisons. There are also 11 large clusters containing 458 ABC transporters, and these protein families constitute a difficult problem that will be discussed in detail later.

Merging clusters

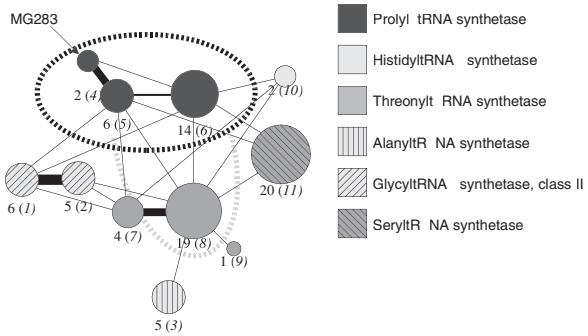
We have used various different algorithms to improve the composition of the clusters in terms of species representation. By merging neighbouring clusters it has been possible to obtain clusters with a composition closer to the ideal composition of orthologous families.

In Figure 3 (panels c and d), the result of merging neighbouring clusters using the join6 algorithm is shown. It can be appreciated that, together with the expected increase in cluster size, there is also a significant change in the quality of the clusters. In the case of MG431 (triosephosphate isomerase), the join6 procedure produces a cluster of size 20 by combining a cluster of size 16 that is made up of 15 bacteria genes and their yeast orthologue, with a small cluster that contains the four corresponding archea sequences. The mean capacity between the two original clusters is 6.3, which is above the minimum cutoff of 6. The reconstruction of clusters is not effective in identifying families in all cases, e.g. the MG429 cluster contains 11 phosphoenolpyruvate kinases, and is merged with a neighbouring cluster that contains 13 phosphoenolpyruvate synthases. Even if these two clusters are closely related (mean capacity of 23.2), it would have been better to keep them separate. These two cases illustrate the fact that a fixed cutoff (represented by the mean capacity) cannot determine the correct orthology relationships in all cases.

Merging of clusters with a relative entropy measurement

The S_var relative entropy procedure implies an incremental aggregation of neighbouring clusters that fulfil the conditions described in the Section **Methods**. Following the example of MG283 in detail can clarify the process (Figure 4). Initially MG283 forms a cluster with the other Mycoplasma Prolyl-tRNA synthetase, and then this cluster is merged with a cluster containing orthologues from the four archea, yeast and *B. burgdorferi*. The final construction of the family consists of a merge with the 14 remaining bacterial orthologues. It is also interesting to see how a merge with the cluster of 19 threonyl-tRNA synthetases sequences is correctly rejected by the procedure, as the genome distribution does not correspond with the expected model.

A second interesting example is the reconstruction around ribosomal protein S2 (MG070). This sequence is initially associated with its counterpart in *M. pneumoniae* ($\times 2$ procedure). The first cluster that is merged contains the 14 remaining bacterial sequences, and this then



Merging process

cluster id	4	5	6	8
#genes/#genomes	2/2	8/8	22/21	41/21
Srel (S var)	3.67	1.25	0.17	0.19

Fig. 4. Free representation of the results of the Ncut clustering algorithm applied to searches with the MG238 sequence (prolyl-tRNA synthetase). The clusters around the MG238 sequence are shown. The circles are proportional to the number of sequences in the cluster, and the connections represent the strength of the relationship between clusters. The black dotted ellipse indicates the limit of the merging procedure, that is, the last merging process accepted by the algorithm, while the gray dotted ellipse indicates the last attempted and rejected merging of clusters. The numerical values of relative entropy are also given in the figure.

is followed by a cluster with the corresponding yeast sequence. A cluster containing the four corresponding archea sequences is then joined to these, so completing the set of 21 orthologues. The procedure then goes on to merge a cluster that contains two additional yeast sequences (paralogues of the first yeast gene). Interestingly, the corresponding COG contains the same 23 sequences.

The distribution of cluster sizes clearly improves after applying the merging procedure (Figure 3, panels (e) and (f)), with an obvious enhancement in the number of clusters of size 16–17 and size 20–21, which are composed mainly of sequences from bacteria-only or all species genomes respectively. For example, of the two cases discussed above for the join6 procedure, cluster MG431 is completed with the full set of orthologues, and cluster MG429 is built correctly, avoiding erroneous inclusion of paralogous sequences.

For the 53 clusters with 16 or 17 sequences obtained by the $\times 2$ S_var procedure, 33 cases (62.3%) contained no duplicated genes, and 49 (92.5%) contained three or fewer duplications. Similarly, for sizes 20 to 23, there are 72 occurrences, with 57 of them (79.2%) containing four or fewer duplications. Consequently the clusters are now formed mainly by one sequence from each genome and include very few duplicated genes.

We observed that sizes 22–23 are more frequent than

sizes 20–21. For the 42 clusters of size 22 or 23 that contained few duplications (out of 53 clusters with this size), most of these contained yeast duplications (36 out of 42; 86%), and to a lesser extent (19%) *B. subtilis* duplications. It is interesting to note that the presence of duplicated genes does not seem to be solely related to the size of the genome, since here the two genomes are not very different in size (in our collection there are 4122 bacillus genes and 5932 yeast genes), and instead it seems to be more related to the history of the duplications. This degree of internal redundancy (Wolfe and Shields, 1997) is captured better in the S_one model than in the S_var model (Figure 5). The S_one model assumes the same number of genes per genome in each cluster, while the S_var model expects the number of genes per genome in clusters to be proportional to the total number of genes per genome. Although the S_one model is more realistic than the S_var model, it incorporates duplicated genes less efficiently, which is reflected in a larger number of clusters in the 16–17 range and fewer in the 20–23 range.

Comparison with COGs

In this section we compare our results with the well-established COGs classification. It is important to keep in mind that our approach is essentially different from that of COGs. COGs represent a global view of the genome relationships, and require analysis of full genomes, whilst our clustering approach addresses the local classification of neighbouring sequences, without requiring complete genomes. To make the two classifications comparable we have analyzed the same 21 genomes that are included in COGs.

Distribution of COGs sizes

COGs present two well-defined populations of cluster sizes (Figure 3, panel g), corresponding to groups with 16–17 and 22–23 genes. In the 16–17 range there are 62 clusters that contain an MG gene, 40 (65%) that contain no duplicated genes from the same genome, with most of them (50 clusters, 81% of the total) containing three or less duplications. In the 20–23 sequences range there are 87 COGs, with 10 of them (11%) having no duplicated sequences, and 67 (77%) having less than four duplications. Both the distribution and the overall number of duplications are very similar to those obtained with the S_var procedure (see Figure 6).

Direct comparison of the clusters with the COG groups

We first check the degree of similarity in the composition of Ncut clusters, S_var merged clusters and COG groups, using our definitions of ‘coherent’ and ‘identical’ clusters (see Section **Methods**). The Ncut clusters are similar or smaller than the corresponding COGs (accumulation of points on or below the diagonal in Figure 7). The behaviour of Ncut clusters of sizes of around 17 is interesting. These clusters correspond in many cases to COG groups with representation from all the genomes, indicative of the tendency of the Ncut procedure to produce small clusters with closely-related sequences, in this case bacterial sequences. The S_var clusters are very similar to the COGs groups (diagonals in Figure 7); for example there are many clusters and COGs groups composed of 16–17 and 20–23 sequences. This is particularly clear for the $\times 4$ S_var clusters, where from the 350 comparable clusters (those that contain three or more lineages), 298 of them (85%) have coherent distributions, and 159 (53%) identical ones (see Section **Methods** for definitions).

In a number of cases, COGs contain more sequences than the Ncut and S_var clusters. For the $\times 4$ NCut clustering it is obvious that our implementation of the Ncut algorithm breaks up the sequence space at a fine grain level. In the case of S_var, the reconstruction process is sometimes caught in a relative entropy local minimum and in others the expected distribution does not adequately represent the real species distribution.

In some cases our clusters are bigger than the COG groups (8 cases for $\times 4$ Ncut and 48 for $\times 4$ S_var), with a considerable difference of more than 15 additional genes. These cases tend to be related to super-families such as ABC transporters, for which the difficulty has already been mentioned, and with the merging of two groups of orthologous sequences (especially in the case of S_var).

There are 11 cases out of the 150 comparable clusters (7.3%), in which COGs and Ncut clusters ($\times 2$ procedure) have different compositions (not coherent clusters as defined in Section **Methods**). This increases to 7.5% for the $\times 2$ join6 procedure (25 out of 332 cases), and to 12.5% (43 out of 344) of the $\times 2$ S_var cases. One example might be the *M. genitalium* ‘chain release factor A’ sequence that is the origin of the MG258 cluster. This cluster contains 32 genes (Ncut procedure), including a mixture of protein release chain factor-A and factor-B sequences. The corresponding COGs groups are COG0216 with 20 sequences of ‘chain release factor A’ (18 of them included in the Ncut cluster), and COG1186 which contains 14 sequences of release factor B. It is interesting to note that the two extra sequences incorporated in COG0216 correspond to second copies of the yeast and *E. coli* genes (genes YLR281c and yaeJ). These sequences are not incorporated within our clusters since they are very distantly related to the rest of the group. Therefore, even if the Ncut procedure does not separate these two groups of release factors when it should, it does correctly identify the two extra sequences as external to the cluster (see phylogenetic tree at http://www.pdg.cnb.uam.es/GenoClustering.html).

A second incoherency between the $\times 2$ Ncut procedure

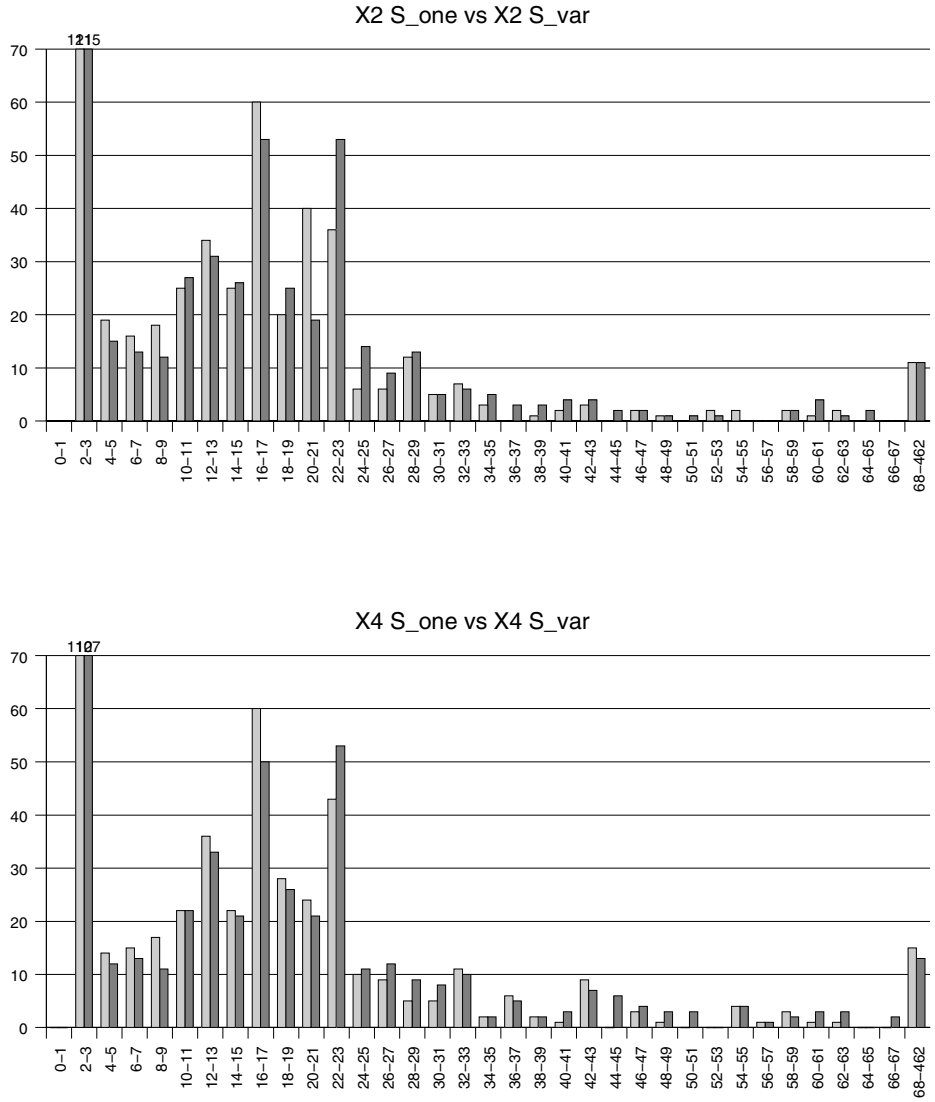


Fig. 5. Comparison of two models of distribution of genome sequences (S_var and S_one models). The S_var model assumes a contribution of genes from each genome proportional to the total number of genes in that genome, while the S_one model implies the expectation of groups with one gene contributed from each genome in the clusters. The representation is similar to the one in Figure 3. Light gray, S_one; dark gray, S_var.

and COGs illustrates how the manual work carried out during construction of the COGs attenuates some of the problems of the automatic procedures. The cluster around the MG457 gene (annotated as cell division protein ftsH), corresponds to COG0465, which contains 27 sequences and is labeled as ATP-dependent Zn proteases. The Ncut cluster contains 61 genes, with 26 of them corresponding to the COG0465 group and the rest to other COGs (COG0464, COG1222 and COG1223). Apart from the error implied in the creation of the Ncut cluster by the addition of extra sequences, it is interesting to see how gene HI1465 from *Haemophilus influenza*, the one missing from COG0465, is not included in the cluster because the absence of a long N-terminal fragment decreases the pairwise similarity. The other nine discrepancies between Ncut and COGs correspond to different ABC transporters.

The MG120 cluster serves as an example of an imperfect match with COGs when the S_var procedure is applied (Figure 8). MG120 and MG121 are annotated as 'uncharacterized ABC transporters components of the permease complex,' and belong to a COG group with 21 genes (COG1079). In Ncut clustering the MG120 gene and the

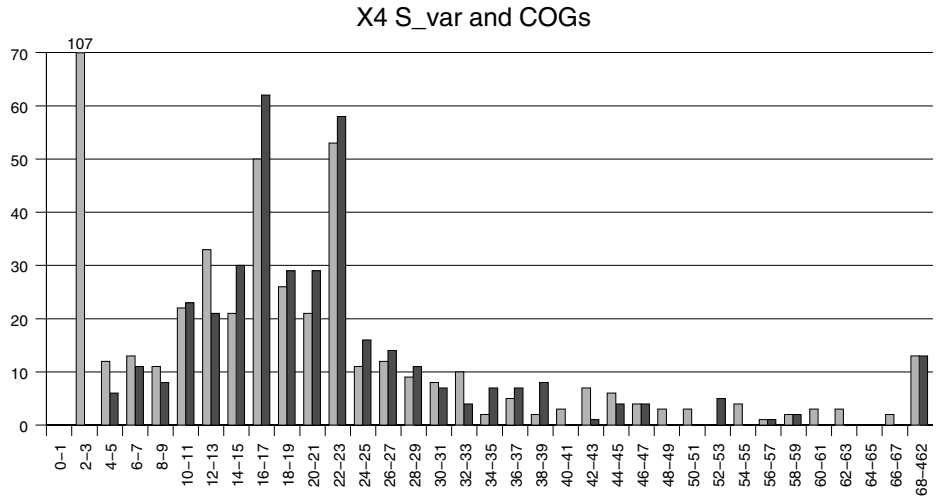


Fig. 6. Comparison of the distribution of cluster sizes for S_var and COGs. The representation is similar to the one in Figure 3. Light gray, S_var; dark gray, COGs.

MP relative form a separate cluster since the relative distance with the corresponding sequences in other species is very large. In the S_var procedure this cluster is merged with a cluster of six sequences that contains the orthologues from K, V, B, O, L and A (see Section **Methods** for abbreviations). The next candidate cluster is a single sequence cluster that is not merged because it increases the relative entropy. The corresponding COG is larger, and includes an orthologous group of paralogues. In the corresponding COG there are also genes that we did not identify in the recursive sequence searches.

DISCUSSION

We propose an algorithm for detecting protein families based on analysis of the relationships in a restricted region of sequence space. This approach is different from other strategies that consider relationships in the full sequence space, such as ProtoMap and GeneRAGE, which are based on the detection of characteristics of the different protein families (methods based on sequence profiles such as PFAM—Bateman *et al.*, 2000), or on phylogenetic analysis of sequences from complete genomes (COGs).

The detection of protein families can be carried out with or without complete genomes. In the first case, a populated representation of sequences in the sequence space is enough for detecting groups that correspond to protein families and subfamilies. For the latter case, the application to sets of small genomes necessitates the use of additional information related to the expected representation of the genomes in the corresponding families.

The method proposed here addresses these two problems: the first by clustering sequences around a given query protein based on information on the pairwise similarities, and the second by merging neighbouring clusters to create families of potential orthologues.

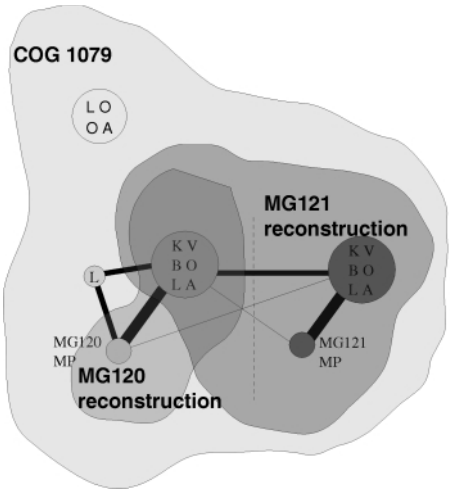


Fig. 8. Comparison of the results obtained for the MG120 gene as compared with the corresponding COGs group. The ×2 clustering results are shown using **circles** for the clusters, **letters** indicating the genomes represented in each cluster, and **lines** showing the connections between sequences of different clusters. The line thickness indicates the strength of the connection. Finally, the **dotted lines** indicate the separation between groups of orthologous sequences.

The first step (clustering) produces groups of clearly-related sequences with the corresponding connections be-

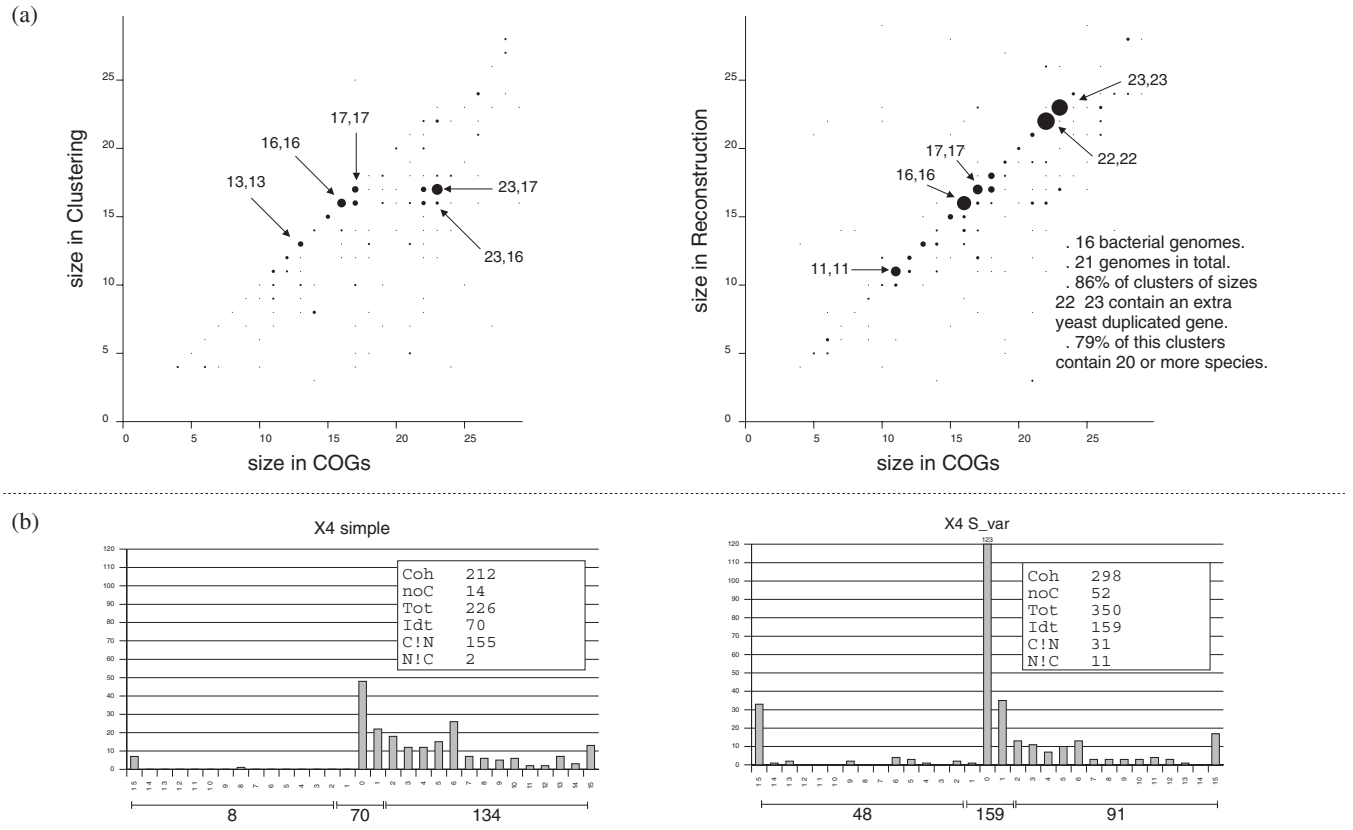


Fig. 7. Comparison of the content of the Ncut and S_var clusters with the corresponding COGs groups. In (a) the sizes in COGs and Ncut/S_var of the coherent clusters are shown. The size of the dots is proportional to the number of occurrences. In (b) the projection of the *diagonals* is shown. The x-axis represents the size difference between coherent clusters, and the y-axis the number of occurrences. In the **box** the number of clusters coherent with the COGs groups (**Coh**) and the number of identical clusters (**Idt)** are shown. 'Coherent' are those clusters that are subsets of the COG groups, and 'identical' are those composed of the same number of sequences (plus or minus one sequence). **NoC** are the non-coherent clusters and **Tot** is the total number of comparable clusters (Coh + NoC). **C ! N** is the number of cases in which COGs, but not NCUT, gets a cluster with more than three lineages. **N ! C** are the cases where there are clusters (with more than three lineages) for NCUT but not for COGs.

tween groups. The second process (merging of clusters) is designed to produce groups of equilibrated genome composition that may capture the classification of orthologous sequences when genomic sequences are analyzed. The detection of orthologous sequences is a key problem in comparative genomics.

We have applied the clustering process to different proteins, generating their corresponding families. For the example of the *ras* protein family, presented here, it is possible to see how the Ncut algorithm generates connected clusters of sequences that compare favourably with those produced by other methods, such as ProtoMap.

The merge procedure has been applied to the clusters generated with sequences from 21 complete genomes around each one of the *M. genitalium* proteins. This scenario was selected so as to be able to compare the results with the COGs standard classification, since COGs has been obtained under expert supervision.

It is interesting to note that the initial clustering process has difficulties with the definition of tight evolutionary relationships in small data sets such as the one considered here, and the clusters generated tend to be smaller than the corresponding COG groups.

Merging the initial clusters produced groups of very similar size and composition to those reported in COGs. This demonstrates that the information contained in the pairwise similarities, the clustering process and the algorithm for aggregation into larger groups, are all appropriate for the reproduction of groups of closely-related sequences with an equilibrated phylogenetic composition. The main difficulty was encountered with very large families, such as the ABC transporter superfamily. These

transporters contain two domains, an ATP cassette subunit and a trasmembrane subunit, that can be coded in the same or different genes (Dean and Allikmets, 1995), making the relationships between domains very difficult to analyze in evolutionary terms. In a few cases our clustering procedure creates groups of paralogues (orthologous groups of paralogues) when the species representation does not fit well with the expected distribution.

Application to the selection of targets in structural proteomics

In the context of structural proteomics projects, the present algorithms can first be used to produce clusters of closely-related sequences around any potential target. These clusters correspond in most cases to well-established protein families with a good separation of their subfamilies. In this sense, our approach is similar to other proposals for the selection of protein targets for structural analysis based on the classification of protein families (Elofsson and Sonnhammer, 1999; Linial and Yona, 2000; Brenner, 2000; Vitkup *et al.*, 2001; May, 2001; Heger and Holm, 2000). The clustering process can generate these families automatically by inspecting closely-related sequences around potential targets, without the need for maintaining a full classification of the sequence space.

The second set of methods dedicated to the aggregation of potentially orthologous sequences can be useful in deciphering the functional correspondence between related sequences. It has been repeatedly stated that detecting orthologous sequences is necessary for the prediction of function, but there is also the potential danger of assigning identical functions to paralogous sequences (Bork and Koonin, 1998; Smith and Zhang, 1997; Doerks *et al.*, 1998; Henikoff *et al.*, 1997; Bork *et al.*, 1998; Devos and Valencia, 2000). Indeed, the selection of orthologous sequences is the goal of databases such as COGs. In the context of the selection of targets for structural proteomics, the detection of possibly orthologous sequences can be important in at least two scenarios. In the first, it is conceivable that structural groups may be interested in investigating proteins with the same function in different organisms for properties such as thermal stability, or to address practical questions such as the selection of sequences from organisms with particularly favourable growth conditions. The second application for which this method can be useful is for the selection of proteins with similar structure but probably differing functions, i.e. paralogues in the same or different organisms. The algorithms proposed here, which are capable of automatically reproducing classifications which are similar to those of COGs, can be used as a first step in these directions, with the advantages of being completely automatic and applicable to any collection of sequences.

ACKNOWLEDGEMENTS

We would like to acknowledge the suggestions of O.Olmea (Mount Sinai School of Medicine, NY) for the application of the clustering strategies, and the graph-based representation of the recursive search results. The continuous support and interesting discussions of the Protein Design Group members are also acknowledged. Our work has benefited from the interesting ideas on the Ncut algorithm as described in G.Yona's PhD work (Yona, 1999), and from use of the MESCHACH numerical library, made public by D.E.Stewart and Z.Leyk. This work has in part been supported by a grant from the Spanish Ministry of Science and Technology (CICYT), and by a fellowship from Madrid's local government.

REFERENCES

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.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Bateman,A., Birney,E., Durbin,R., Eddy,S.R., Howe,K.L. and Sonnhammer,E.L. (2000) The pfam protein families database. *Nucleic Acids Res.*, **28**, 263–266.

Bork,P. and Koonin,E.V. (1998) Predicting functions from protein sequences—where are the bottlenecks? *Nature Genet.*, **18**, 313–318.

Bork,P., Dandekar,T., Diaz-Lazcoz,Y., Eisenhaber,F., Huynen,M. and Yuan,Y. (1998) Predicting function: from genes to genomes and back. *J. Mol. Biol.*, **283**, 707–725.

Brenner,S.E. (2000) Target selection for structural genomics. *Nat. Struct. Biol.*, **7**, 967–969.

Chothia,C. and Lesk,A.M. (1986) The relation between the divergence of sequence and structure in proteins. *EMBO J.*, **5**, 823–826.

Dean,M. and Allikmets,R. (1995) Evolution of ATP-binding cassette transporter genes. *Curr. Opin. Genet. Dev.*, **5**, 779–785.

Devos,D. and Valencia,A. (2000) Practical limits of function prediction. *Proteins*, **41**, 98–107.

Doerks,T., Bairoch,A. and Bork,P. (1998) Protein annotation: detective work for function prediction. *Trends Genet.*, **14**, 248–250.

Elofsson,A. and Sonnhammer,E.L. (1999) A comparison of sequence and structure protein domain families as a basis for structural genomics. *Bioinformatics*, **15**, 480–500.

Enright,A.J. and Ouzounis,C.A. (2000) GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, **16**, 451–457.

Fitch,W.M. (1970) Distinguishing homologous from analogous proteins. *Syst. Zool.*, **19**, 99–113. No abstract available.

Garcia-Ranea,J.A. and Valencia,A. (1998) Distribution and functional diversification of the ras superfamily in *Saccharomyces cerevisiae*. *FEBS Lett.*, **434**, 219–225.

Gerstein,M. (1998) Measurement of the effectiveness of transitive sequence comparison, through a third 'intermediate' sequence. *Bioinformatics*, **14**, 707–714.

Heger,A. and Holm,L. (2000) Towards a covering set of protein family profiles. *Prog. Biophys. Mol. Biol.*, **73**, 321–337.

Henikoff,S., Greene,E.A., Pietrokovski,S., Bork,P., Attwood,T.K. and Hood,L. (1997) Gene families: the taxonomy of protein paralogs and chimeras. *Science*, **278**, 609–614.

Koonin,E.V., Mushegian,A.R. and Bork,P. (1996) Non-orthologous gene displacement. *Trends Genet.*, **12**, 334–336.

Linial,M. and Yona,G. (2000) Methodologies for target selection in structural genomics. *Prog. Biophys. Mol. Biol.*, **73**, 297–320.

Lupas,A., Van Dyke,M. and Stock,J. (1991) Predicting coiled coils from protein sequences. *Science*, **252**, 1162–1164.

May,A.C. (2001) Optimal classification of protein sequences and selection of representative sets from multiple alignments: application to homologous families and lessons for structural genomics. *Protein Eng.*, **14**, 209–217.

Minieka,E. (1978) *Optimization Algorithms for Networks and Graphs*, Chapter 4, Marcel Dekker, New York.

Park,J., Teichmann,S., Hubbard,T. and Chothia,C. (1997) Intermediate sequences increase the detection of homology between sequences. *J. Mol. Biol.*, **273**, 349–354.

Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1992) *Numerical Recipes in C: the Art of Scientific Computing*. William,H.Press (ed.), Cambridge University Press, New York.

Shi,J. and Malik,J. (1997) Normalized cuts and image segmentation. *Proc. IEEE Conf. Comp. Vision Pattern Recognit.*, 731–737.

Smith,T.F. and Zhang,X. (1997) The challenges of genome sequence annotation or 'The devil is in the details'. *Nat. Biotechnol.*, **15**, 1222–1223.

Tatusov,R.L., Galperin,M.Y., Natale,D.A. and Koonin,E.V. (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.*, **28**, 33–36.

Tatusov,R.L., Koonin,E.V. and Lipman,D.J. (1997) A genomic perspective on protein families. *Science*, **278**, 631–636.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Valencia,A., Chardin,P., Wittinghofer,A. and Sander,C. (1991) The ras protein family: evolutionary tree and role of conserved amino acids. *Biochemistry*, **30**, 4637–4648.

Vitkup,D., Melamud,E., Moult,J. and Sander,C. (2001) Completeness in structural genomics. *Nat. Struct. Biol.*, **8**, 559–566.

Wolfe,K.H. and Shields,D.C. (1997) Molecular evidence for an ancient duplication of the entire yeast genome. *Nature*, **387**, 708–713.

Wootton,J.C. and Federhen,S. (1993) Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.*, **17**, 149–163.

Wu,Z. and Leahy,R. (1993) An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *PAMI*, **11**, 1101–1113.

Yona,G., Linial,N. and Linial,M. (1999) ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins*, **37**, 360–378.

Yona,G. (1999) *Methods for Global Organization of the Protein Sequence Space*, PhD Thesis, Hebrew University.

Zuckerkandl,E. and Pauling,L. (1965) Molecules as documents of evolutionary history. *J. Theor. Biol.*, **8**, 357–66.