EVALUATION OF GRAPH BASED PROTEIN CLUSTERING METHODS

COSTA, G. G. L
Institute of Computing, State University of Campinas, Albert Einstein Avenue 1251
Campinas, São Paulo, Brazil

DIGIAMPIETRI, L. A.
Institute of Computing, State University of Campinas, Albert Einstein Avenue 1251
Campinas, São Paulo, Brazil

OSTROSKI, E. H.
Institute of Computing, State University of Campinas, Albert Einstein Avenue 1251
Campinas, São Paulo, Brazil

SETUBAL, J.C.
Virginia Bioinformatics Institute, Virginia Tech, VA 2406
Blacksburg, Virginia, United States

Protein clustering is widely used in order to characterize functionally proteins. Many automatics methods for protein-clustering use a graph-based approach. In this work, we propose a methodology for evaluation of the solution given by these methods.

1. Introduction

In the last decade more than two hundred prokaryote genomes and dozens of eukaryote genomes have been sequenced [1]. This has allowed to identify and to catalog thousands of genes from the sequenced organisms. With the exponential growth of genomic data, the notion of homologous gene families became central.

In this context of great growth of genomic data, it is not feasible to realize laboratory experiments to identify the function of each sequenced gene, and not even to use phylogenetic methods to group genes in a high throughput way. Therefore sequence similiarty algorithms are widely used to cluster gene/protein sequences. This paper focuses on the development of a methodology for evaluation and validation of protein clusters that are the result of computational clustering methods. This is motivated by the fact that a consistent and uniform methodology with that aim is still lacking.

This paper is organized as follows. In section 2 we present the used basic concepts. Then, in section 3, we describe briefly some of the many protein-clustering methods available in the literature. In section 4, we show the evaluation criteria used in this work. In section 5, we show the metrics applied to the solution given by each method. In section 6, we discuss the results. Finally, in section 7, we conclude and suggest future work.

2. Basic concepts

2.1. Homology

Homologous genes are categorized into two categories: *paralogs* (if these genes belong to the same organism) and *orthologs* (if the homologous genes belong to distinct organisms).

Ideally, one should create a phylogenetic tree of sequences involved in order to group them into homolog families. However, phylogenetic analyses are difficult and time consuming. Therefore, sequence similarity is used to infer homology relationships. Similarity, which allows a mathematic definition, can be measured through several algorithms, e.g [2]. The greatest difficulty is to infer homology relationships when there are no good direct similarities among two sequences, because, in this case, it is necessary to use indirect similarity relationships. These relationships can result in bad categorization of some proteins into families (false positives).

2.2. Protein domains

The function of a protein is determined by its tridimensional folding (tertiary structure). Nevertheless we can use the aminoacid sequence (primary structure) to deduce the folding pattern of the protein. Here, the concept of protein domain is central. Domains are functional substructures inside a protein. They are subsequences that induce some folding pattern and specific function.

When analyzing the primary sequence of a protein, we usually search for domains because they are the most conserved and important regions of the sequence in the functional viewpoint. So, when analyzing classification methods to generate gene families, we must have in mind the biologic reality that we want to map.

2.3. Classification into protein families

A graph based approach for the protein clustering problem is tipically formulated like this:

Input: a set of n aminoacid sequences representing proteins of one or more organisms.
Comparison: Sequences are compared all against all for obtaining some measure of similarity.
Output: A classification of the input sequences into groups (families).

Underlying of the comparison process there is a graph G where each protein sequence corresponds to a vertex. There is an edge between two vertices u and v iff they are enough similar according to a certain similarity criteria.

3. Existing Methods for Protein Family Classification

3.1. COGS

Tatusov et al [3] developed the COG database – Clusters of Orthologous groups. The objective of the COG project is the classification of proteins from complete genomes into orthologous groupings called COGs. The construction strategy of the COGs is based on the principle that any set of tree or more proteins from distant genomes that are more similar to each other than to other proteins constitute an orthologous group.

All proteins are compared all against all with BLASTP [2]. The resulting graph G is undirected and unweighted. The more evident paralogs are grouped in one vertex. Triangles of orthologous proteins are detected. This triangles are formed when the best alignment from a protein in the organism A in relation with its two orthologous from the organisms B and C is better than the second best alignment, been that the same relation stay valid for the three involved proteins and organisms. After this, the triangles that share one side are joined in the same COG creating bigger COGs.

This criterion has the advantage of accommodate too proteins that evolved very fast and do not have a great similarity. In agreement with this criterion, even proteins with sequence relatively distant can be grouped in the same COG. Nevertheless, there is a non-repressible pos analyses phase that become the process dependent of a great manual intervention. Each family is analyzed manually and is dismounted or grouped with other families if necessary.

New members are added to the original COGs through the program COGNITOR that, given a query protein, determines, among the COGs stored, what is the COG that best accommodates this protein. It is necessary that a protein presents at least two best hits with the member of the same COG to become candidate to enter in this COG.

This classification is of great importance to the gene functional category prediction and annotation. It was demonstrated that, typically, between 95% and 97% of the COGNITOR predictions do not require corrections.

3.2. Connected components

For the project Xylella fastidiosa [4], the following similarity criteria was adopted: two proteins are similar if their alignment has a e-value (measured by

BLASTP) less or equal then 10e-5 with a coverage of at least 60% of the query sequence and 30% of the subject sequence.

Thus, the underlying graph G is undirected and not weighted. Families are given by the connected components of G. This approach is also known as single linkage clustering and it demands that a protein has to be similar to at least one protein in a group to be included in that group. As a consequence of the methodology, families are disjoint.

There are two crucial problems in this approach. First, results depend critically on threshold values for the statistical significance (e-value) and alignment coverage. It’s difficult to empirically find good values. This comes from the fact that some families are more cohesive, that is, proteins are more near from each other, while others are sparser. Thus, threshold values that lead to some good cohesive families might erroneously divide others.

Besides, there is the problem of potential long chains. For example, let X, Y and Z be three proteins. In this example let’s consider that there is no coverage cutoff. X contains domains {A,B,C}. Y, domains {C,D,E} and Z the domains {D,E,F}. The graph subjacent to proteins X,Y and Z is isomorph to P_3, however X and Z share no domains. These chains could be arbitrary long.

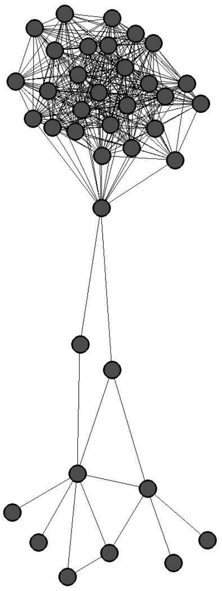

Figura 1 Long chains. A sample family in the superior side and members incorrectly assigned to this group by single linkage clustering

3.3. Cliques (Complete Linkage Clustering)

During the realization of *Agrobacterium tumefaciens* genome project [5] was been observed that use only connected components to determine the families was a criterion maybe too permissive. It was decided to find a new criterion based in cliques. The new methodology proposed by Almeida [6] uses two CS's that will be called here CS1 and CS2. Both these criteria are based in thresholds values to the statistical significance and alignment cover, been CS1 more restrictive than CS2. Starting from CS1, a graph G is constructed and all its maximal cliques are determined. The families are determined in the next step, through the union of each maximal clique C with the proteins that yet do not belong to any family and that present a satisfactory similarity (in agreement with CS2) with some member of C.

In the case of, in the second step, a protein presenting similarities with proteins of distinct cliques, an untie algorithm decides in which family this protein must be included. Therefore, the result families are disjointed. The graph G subjacent to this process is undirected and without weights in the edges.

This method, in spite of seems to be stronger than the previous, too presents problems. Although the clique idea is intuitively good, it is used only in a partial way. The method, by obligate that a gene belong to only one family, take some relatively arbitraries decisions, what can generate incorrect and incomplete families.

3.4. Componentes Biconexos

This approach was introduced by KIM [7]. The algorithm is based on bi-connected components and articulation points. Proteins are compared all against all in order to obtain some similarity measure and form an undirected weighted graph. Biconnected components of this initial graph are candidates to become families and the articulation points are marked as possible points of union between families.

The process is repeated with several threshold values in a range and the bi-connected components and articulation points are marked for each of these values. Thus, the classification resultant is basically hierarchical and can lead well with cohesive and sparse families.

3.5. Componentes Fortemente Conexos

This is the approach followed by YONA et al [8]. The usual similarity measures are obtained and are used to create a directed weighted graph, where edges are directed from query to subject (as returned by BLAST).

Edge weights are computed according to the rank in the hit list, rather than the raw similarity score. For a given (query) protein, the edge weight in an edge that links it to one of its BLAST hits is a function of the ranking of that hit in the hit list. Families are taken to be the strongly connected components in the resulting graph.

3.6. Tribes-MCL

The emphasis of this method is on algorithm MCL – Markov Clustering [9]. The similarity criteria used is simple: proteins are compared all against all with BLAST and edges are weighted with –log(e-value). The underlying graph is thus directed.

Despite the simplicity of this similarity criteria, the algorithm MCL can still detect homology relationships. The algorithm represents a process that captures the concept of random walks in a graph and does it deterministically. This is achieved representing the graph as a Markov Matrix and establishing the algebraic operators that transform the matrix probabilities.

Edge weights are understood as probabilities of visiting them. For each application of the algebraic operators, probabilities of regions of greater flow are increased and probabilities of regions of less flux are diminished. After many iterations, algorithm tends to find natural groups of the protein space. Process is repeated until groups remain unchanged.

3.7. OrthoMCL

This technique was developed focusing on eukaryote proteins. Just like Tribes-MCL, the OrthoMCL [10] algorithm use MCL in the last phase. However, it differs from Tribes-MCL in the way edges are weighted.

In OrthoMCL, from the initial all against all comparison, hits whose e-value is greater than 1E-5 are filtered out. Then the method identifies putative paralogy and orthology relationships. According to LI et al [10], two ortholog/paralog sequences are reciprocal best hits. Paralogs are those proteins that belong to the same organism and that are more similar to each other than to any protein in the grouping belonging to another organism.

The underlying graph only contains edges of putative orthologs and paralogs, according to the criteria described. Edges are weighted with –log(e-value), without any coverage requirement for the alignments.

Then there is a phase of normalization of edge weights. In order to eliminate the interference of the high score of recent paralogs in relation to ortholog edges, weights are normalized by the score of all orthologs between two organisms (or by the weight of all recent paralogs in the same organism).

Finally, the underlying directed weighted graph is represented as a symmetrical matrix that is fed to the MCL algorithm. Then MCL finds groupings that will be output as the protein families.

Table 1 Graph based protein clustering methods and theirs main characteristics

	Directed	Weighted	Graph structure	Flexibility iofthreshold values	Family hierarchization	Produces disjoint families	Manual intervention
Xylella fastidiosa	N	N	Connected components	N	N	Y	Lower
A. tumefaciens	N	N	Cliques	Partial	N	Y	Lower
Gbag	N	N	Biconnected components	Y	Y	N	Lower
Protomap	Y	Y	Strongly connected components	Y	N	Y	Lower
COG	N	Y	K_3	N/D	-	-	Higher
Tribes-MCL	Y	Y	Markov Clusters	-	N	Y	Lower
OrthoMCL	N	Y	Markov Clusters	-	N	Y	Lower
NCUT	Y	Y	Cuts	Y	N	Y	Lower

4. Evaluation criteria

In the former section, we define the problem in a general aspect. Some variables were left behind. Most protein clustering methods differ in the way they treat them.

Some variables are the similarity criteria (SC), presence or absence of oriented edges of the graph G, the graph structure of the families and disjunction or not of the output families. This work also shows how different approaches in the literature treat these questions.

Generally, all methods deal with a compromise between homogeneity and separation. Families should be intrinsically cohesive and extrinsically separated from others families and there are some metrics (presented in section 5) that can evaluate this compromise.

We use HAMAP [11], a well-formed and specialist-validated protein family set to check how these metrics can recognize good families or orient an automatic classification method. In order to evaluate how the many protein-clustering methods available in the literature perform, we have input to them proteins pertaining to some special HAMAP families.

In order to make our benchmark, we have tried to choose HAMAP protein families that are not so separated from each other. To accomplish this task, we have mounted an undirected graph *G* where each vertex represents an original HAMAP family and each edge *(u,v)* indicates that there is at least one blast hit between one protein in family *u* and one protein in family *v*. Figure 2 shows graph G.

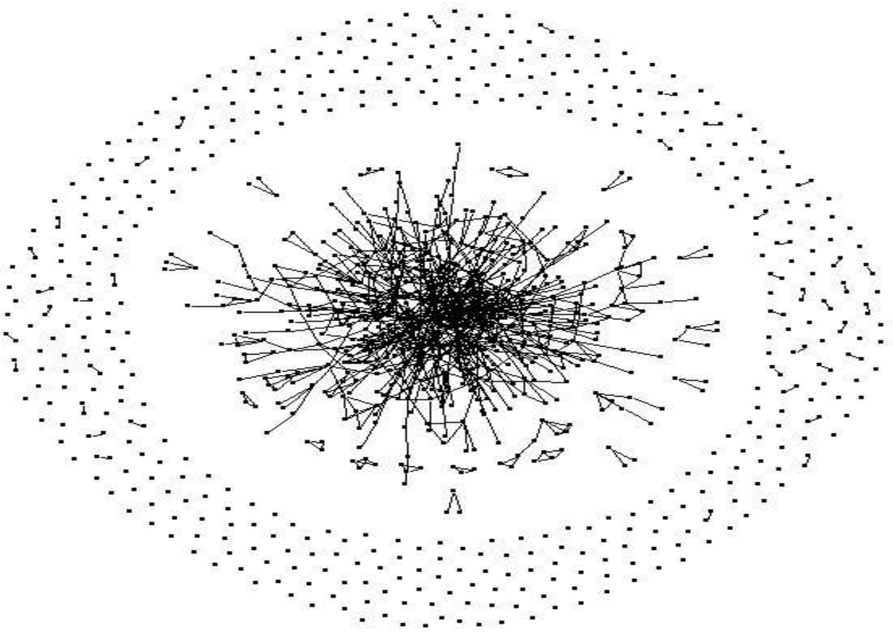

Figura 2 HAMAP families. Each vertex represents a family. Edges represent blast similarities between members of adjacent families

We have included in our reference set (RS) proteins from biconnected components of G whose families are disjoint.

This has resulted in 2708 proteins from 73 original HAMAP families, including homologs from 419 species.

With this RS, we are able to establish metrics to evaluate how well protein-clustering methods behave on reconstructing the original HAMAP families. These metrics will be presented in section 5.

In order to normalize the way metrics are obtained, the similarity graph is constructed doing all vs all comparison with blast. Edges are weighted with –log(e-value). Edges with weight less or equal than 2 are removed from the graph.

5. Metrics

In this section we will describe the three kinds of metrics used in this work. First, we will present metrics used for evaluating individual groups. Second, we will show metrics used for evaluating a solution, i.e. the whole grouping of proteins outputted by a method. Third, we will establish metrics to compare one solution (given by a method) to one set of well-formed and manually validated families.

Almost all metrics in the first and second subsections are pure mathematical concepts that evaluate general graph clustering. Since our problem is biological, we may test which of those metrics are promising to separate good protein groupings from bad ones. In the third section, metrics can directly map the biological meaning desired.

First, we may state the nomenclature used. Let G= (V,E) be a connected and undirected graph. Let $m=|E|$, $n=|V|$ e $C= (C_{1,}...C_k)$ a partition of V. We call the set C a *clustering* and call each inducted subgraph of G by C_i a *cluster*. For *m(C)* we denote the size of the subset of E formed by edges internal to some cluster (intra-cluster edges). Additionally we define *n(Ci)* as the number of vertices in a cluster..

5.1. Metrics for evaluating individual groups

5.1.1. Completeness

Completeness was proposed by [12] and indicates the fraction of the sum of edges of the weights of the edges internal to a given C_i by the maximum possible sum of edges weights w_{max} (when C_i is complete and all edges have the maximum possible weight). So, we have:

$$\textbf{Completeness}(\mathbf{C_i})= \frac{\sum_{e\in C_i} w(e)}{w_{\max}}$$

where

$$w_{\max} = \frac{n(n-1)}{2}k$$

and k is a constant that indicates the maximum possible weight that can be assigned to a single edge.

Completeness can be directly obtained in O(m) and its computation for not weighted graphs is straightforward.

5.1.2. Separation

We can evaluate the separation between a cluster C_i and the remaining graph as the fraction of the sum of weights of the edges internal to C_i and the sum of the weights of the edges that leave the cluster C_i, normalized by the number of nodes in C_i.

Separation(C_i)= $$\frac{\sum_{e \in C_i} w(e)}{\left(\sum_{e=(v,w), v \in C_i, w \notin C_j} w(e)\right) n(C_i)}$$

5.1.3. Diameter

Diameter borrows the concept of graph diameters, that is, the the length of the longest shortest path between any two graph vertices of a graph. For not weighted graphs, diameter can be defined as the number of edges that must be traversed in the longest shortest path. For weighted graphs, diameter can be calculated as the sum of edges weights in such path, i.e.

diam(C_i)=max d(u,v), u $\in$ C_i, v $\in$ C_i

where *d(u,v)* is the sum of edge weights across the shortest path between *u* and *v* in C_i.

5.1.4. Conservation of the domain architecture

We say that a family conserves the domain architecture if all proteins in this family have the same domains at the same order, as found by software HMMPFAM. We only consider proteins that have, at least, one domain.

5.2. Metrics for evaluating a solution

For *solution*, we understand the set of protein families outputted by a method. In this section, we show metrics that indicate the intrinsic quality of a solution.

5.2.1. Coverage

Coverage is the fraction of intra-cluster edges by the whole set of edges of G, ie.

cov(C)= $\dfrac{m(C)}{m}$

It is easy to obtain and it is extensible for weighted graphs. In this case, we can define:

wCov(C)= $\dfrac{\sum_{e=(u,v),u,v\in C_i} w(e)}{\sum_{e\in G} w(e)}$

5.2.2. Internal Consistency

Internal consistency, proposed by BRANDES [12] evaluates the number of correctly interpreted pairs, that is, within all protein pairs, the number of them that satisfy one of the following criteria:

I – v,w ∈ C_i, (v,w) ∈ C

II – v ∈ C_i, w ∈ C_j, i≠j, (v,w)∉ E

Strictly, performance is the proportion of these correctly interpreted pairs within the set of all pairs of nodes. It can be calculated by the following formula.

intConsistency(C)= $\dfrac{m(C)+\sum_{\{v,w\}\notin E, v\in C_i, w\in C_j, i\neq j} 1}{\frac{1}{2}n(n-1)}$

However, calculating performance following this formula is quadratic on the number of nodes. It is easier counting the errors and indirectly calculating performance. According to this approach, we have:

1-intConsistency(C)= $\dfrac{2m(1-2\operatorname{cov}erage(C))+\sum_{i=1}^{k} n(C_i)(n(C_i)-1)}{n(n-1)}$

5.3. Metrics for comparing a solution with a reference set of protein families

First, analogous to the set *C* of protein families in a given solution, we define the set $R=(R_1,R_2,\ldots,R_n)$ of protein families in the reference set.

In order to count true positives, true negatives, false positives and false negatives, we use the following equations.

#NTP = Number of vertices pairs {u,v} correctly grouped in the same cluster Ci, ie. $u,v\in C_i$, $u,v \in R_j$

#NTN = Number of vertices pairs {u,v} correctly grouped in different clusters, i.e. $u,v \notin Ci$, $u,v \notin Rj$, plus the number of singletons in *C* that are also singletons in *R*.

#NFP= Number of vertices pairs {u,v} erroneously grouped in the same cluster, i.e. $u,v \in C_i$, $u,v \notin R_{j.}$

#NFN= Number of vertices pairs {u,v} erroneously grouped in different clusters, i.e. $u,v \notin C_i$, $u,v \in R_{j.}$, plus the number of singletons in C_i that are grouped with some other vertex in R_j.

#T= #NTP + #NTN + #NFP + #NFN

Thus we have:

True positives= #NTP / #T
True negatives= #NTN / #T
False positives= #NFP / #T
False negatives= #NFN / #T

6. Results and Discussion

We have evaluated two methods, single-linkage clustering and Tribes-MCL, with two different parameter set each. Single-linkage clustering was tested with cutoffs of 1E-20 (solution **A**) and 1E-50 (solution **B**) for the e-value and a cutoff of 60% for both query and subject sequences. For Tribes-MCL, we have used inflation values of 1.1 (solution **C**) and 2.0 (solution **D**). Both methods applied were applied to proteins in RS, and results were compared with the original HAMAP assignments. Results are summarized in table 2.

Table 2 Metrics for obtained for the original HAMAP family assignments and solutions A,B,C and D

	HAMAP	A-2	B-7	C-9	D-6
#Families	73	76	165	25	58
Average Completeness	0.77	0.65	0.64	0.47	0.81
Average Separation	1.71	24.48	9.37	109.9	11.6
Average Diameter	1.14	1.36	1.07	2.00	0.56
Maximum Diameter	2	2	2	3	1
Coverage	0.76	0.94	0.76	0.99	0.81
Weighted Coverage	0.92	0.98	0.96	1.00	0.97
IntConsistency	0.99	1.00	0.99	0.99	0.99
#families that conserve domain architecture	43	25	55	9	10
%TRUE POSITIVES	-	3.5%	3.0%	3.6%	3.5%
%TRUE NEGATIVES	-	95.3%	95.8%	94.3%	96.1%
%TRUES	-	98.8%	98.8%	97.9%	99.6%
%FALSE POSITIVES	-	1.1%	0.6%	2.1%	0.4%
%FALSE NEGATIVES	-	0.1%	0.6%	0%	0%

7. Future Work and Conclusion

In this work, we have mounted one benchmark. Metrics obtained for this benchmark can show us the relationship between the mathematical concepts of clustering analysis and the biological reality of protein families. Future work includes mounting other benchmarks in order to do a criterious statistical analysis on the metrics over well-formed families and identify which of them are significant and the target values for good families. Then, we must develop an automatic protein-clustering method that targets these values.

Another expansion of this work is to use the domain architecture information in order to improve the similarity graph before applying the protein clustering methods. Domain conservation among families show that this information can be helpful.

The main contribution of this work is a step towards a methodology for protein clustering methods validation.

References

1. NCBI,Genbank (website:http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html)
2. S. F. Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, V. 25, pp 3389-3402, 1997.
3. R.L. Tatusov, E.V. Koonin, and D.J. Lipman. A genomic perspective on protein families. Science, 278(5338):631-637,1997
4. Simpson et al. The genome sequence of the plant pathogen Xylella fastidiosa. Nature, v. 406, pp. 151-157, (2000)
5. D. W. Wood et al. The Genome of the Natursal Genetic Engineer Agrobacterium tumefaciens C58. Science, 294:2317-2323.
6. N.F.A. Junior. Ferramentas para comparação genômica. PhD thesis, Instituto de Computação - UNICAMP (2002)
7. Kim, Sun. Graph theoretic sequence clustering algorithms and their applications to genome comparison. In Wu, C.H., Wang, P., Wang, J.T.L., eds.: Computational Biology and Genome Informatics. World Scientific, 2003.
8. G. Yona, N. Linial, M. Linial. ProtoMap: automatic classification of protein sequences and hierarchy of protein families. Nucleic Acids Research, volume 28, 49-55 (2000).
9. A. J.ENRIGHT, V. Kunin and C. Ouzonis. A Protein families and TRIBES in genome sequence space. *Nucleic Acids Research*, V.31, pp4632-4638, 2003.
10. L. Li, C.J. Stoeckert, D.S. Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Resource, Vol. 13, No. 9, 2178-2189 (2003).

11. A. Veuthey, C. Lachaize, A. Gattiker, K. Michoud, C. Rivoire, A. Auchincloss, E. Coudert, E. Gasteiger, P. Kersey, M. Pagni, A. Bairoch. The HAMAP project: High quality Automated Microbial Annotation of Proteomes.
12. M. U. Brandes et al. Experiments on graph clustering algorithms. In Proceedings of the 11th Annual European Symposium on Algorithms (ESA’03), LNCS 2832, pp. 568–579, Springer-Verlag, 2003.