A New Alignment-Independent Algorithm for Clustering Protein Sequences

Abdellali Kelil
Department of Computer Sciences
Sherbrooke University
Sherbrooke, QC, Canada
Abdellali.Kelil@USherbrooke.ca

Shengrui Wang
Department of Computer Sciences
Sherbrooke University
Sherbrooke, QC, Canada
Shengrui.Wang@USherbrooke.ca

Ryszard Brzezinski
Department of Biology
Sherbrooke University
Sherbrooke, QC, Canada
Ryszard.Brzezinski@USherbrooke.ca

Abstract—**The rapid burgeoning of available protein data makes the use of clustering within families of proteins increasingly important, the challenge is to identify subfamilies of evolutionarily related sequences. This identification reveals phylogenetic relationships, which provide prior knowledge to help researchers understand biological phenomena. A good evolutionary model is essential to achieve a clustering that reflects the biological reality, and an accurate estimate of protein sequence similarity is crucial to the building of such a model. Most existing algorithms estimate this similarity using techniques that are not necessarily biologically plausible, especially for hard-to-align sequences such as multi-domain, circular-permutation and tandem-repeats protein sequences, which cause many difficulties for the alignment-dependent algorithms. In this paper, we propose a novel similarity measure based on matching amino acid subsequences. This measure, named SMS for Substitution Matching Similarity, is especially designed for application to non-aligned protein sequences. It allows us to develop a new alignment-independent algorithm, named CLUSS, for clustering protein families. To the best of our knowledge, this is the first alignment-free algorithm for clustering protein sequences. Unlike other clustering algorithms, CLUSS is effective on both alignable and non-alignable protein families.**

Keywords-component; Protein sequences; Clustering; Non-alignable; Biological function; Phylogeny

I. INTRODUCTION

With the rapid burgeoning of protein sequence data, the number of proteins for which no experimental data are available greatly exceeds the number of functionally characterized proteins. To predict a function for an uncharacterized protein, it is necessary not only to detect its similarities to proteins of known biochemical properties (i.e., to assign the unknown protein to a family), but also to adequately assess the differences in cases where similar proteins have different functions (i.e., to distinguish among subfamilies). One solution is to cluster each family into distinct subfamilies composed of functionally related proteins. Subfamilies resulting from clustering are easier to analyze experimentally. A subfamily member that attracts particular interest need be compared only with the members of the same subfamily. A biological function can be attributed with high confidence to an uncharacterized protein, if a well-characterized protein within the same cluster is already known. Conversely, a biological function discovered for a newly characterized protein can be extended over all members of the same subfamily.

The literature reports many algorithms that can be used to build protein clustering databases, such as the widely used algorithm BLAST [1] and its improved versions Gapped-BLAST and PSI-BLAST [2], as well as SYSTERS [3], ProtClust [4] and ProtoMap [5] (see [6] for a review). These algorithms have been designed to deal with large sets of proteins by using various techniques to accelerate examination of the relationships between proteins. However, they are not very sensitive to the subtle differences among similar proteins. Consequently, these algorithms are not effective for clustering protein sequences in closely related families. On the other hand, more specific algorithms have also been developed, for instance, the widely cited algorithms BlastClust **[**7**],** which uses score-based single-linkage clustering, TRIBE-MCL **[**8**],** based on the Markov cluster approach, and gSPC **[**9**],** based on a method that is analogous to the treatment of an inhomogeneous ferromagnet in physics. Almost all of these algorithms are either based on sequence alignment or rely on alignment-dependent algorithms for computing the similarity. However, they have several serious limitations, including the following:

- The results depend heavily on the algorithm selected and the parameters set by the user for the alignment algorithm (e.g., gap penalties). As far as easily alignable proteins are concerned, almost every existing alignment algorithm can yield good results. However, for protein sequences that are difficult to align, each alignment algorithm finds its own solution. Such variable results create ambiguities and can complicate the clustering task [10].
- For the case of non-alignable protein sequences (i.e., not yet definitively aligned and biologically approved) such as multi-domain, circular-permutation and tandem-repeats protein sequences, alignment-based algorithms do not succeed in producing biologically plausible results. This is due to the nature of the alignment approaches, which are based on the matching of subsequences in equivalent positions, while non-alignable proteins often have similar and conserved domains in non-equivalent positions [10].

There are other known difficulties that limit the reliability of alignment, especially for the case of hard-to-align protein sequences, such as "*repeat*", "*substitution*" and "*gap*" problems, which are well discussed by Higgins [11].

In this paper, we propose an efficient algorithm, CLUSS, for clustering protein families based on SMS, which is a new measure we propose for protein similarity. The novelty of CLUSS resides essentially in two features. First, CLUSS is applied directly to non-aligned sequences, thus eliminating the need for sequence pre-alignment. Second, it adopts a new measure of similarity, directly exploiting the substitution matrices generally used to align protein sequences and showing a great sensitivity to the relations among similar and divergent protein sequences.

II. The New Similarity Measure SMS

Many approaches to measuring the similarity between protein sequences have been developed. Prominent among these are alignment-dependent approaches including the well-known algorithm BLAST [1] and its improved versions Gapped-BLAST and PSI-BLAST [2], which the programs are available at [7], as well as several others such as the one introduced by Varré *et al.* [10] based on movements of segments, and the recent algorithm Scoredist introduced by Sonnhammer *et al.* [13] based on the logarithmic correction of observed divergence. These approaches often suffer from accuracy problems, especially for hard-to-align proteins sequences. The similarity measures used in these approaches depend heavily on the alignability of protein sequences as well as on the quality of the alignment, which in turn depend on the alignment algorithm used and its chosen parameters. In many cases, alignment-free approaches can greatly improve protein comparison, especially for non-alignable protein sequences. These approaches have been reviewed in detail by several authors [14],[15]. Their major drawback, in our opinion, is that they consider only the frequencies and lengths of similar regions within proteins and do not take into account the biological relationships that exist between amino acids. To correct this problem, some authors [15] have suggested the use of the Kimura correction method [16] or other types of corrections, such as that of Felsenstein [17]. However, to obtain an acceptable phylogenetic tree, the approach described in [15] performs an iterative refinement including a profile-profile alignment at each iteration, which significantly increases its complexity. Considering this, we have developed a new approach mainly motivated by biological considerations and known observations related to protein structure and evolution. The goal is to make efficient use of the information contained in amino acid subsequences in the proteins, which leads to a better similarity measurement. The principal idea of this approach is to use a substitution matrix such as BLOSUM62 [18] or PAM250 [19] to measure the similarity between matched amino acids from the protein sequences being compared.

In this section, we will use the symbol |.| to express the length of a sequence. Let X and Y be two protein sequences belonging to the protein family F. Let x and y be two identical subsequences belonging respectively to X and Y; we use $\Gamma_{x,y}$ to represent the matched subsequence of x and y. We use l to represent the minimum length that $\Gamma_{x,y}$ should have; i.e., we will be interested only in $\Gamma_{x,y}$ whose length is at least l residues. We define $E^l_{X,Y}$, the key set of matched subsequences $\Gamma_{x,y}$ for the definition of our similarity function, as follows:

$$E^l_{X,Y}=\left\{\Gamma_{x,y}\;\middle|\;\begin{array}{c}|\Gamma_{x,y}|\geq l\,,\\ \left(\forall\Gamma_{x',y'}\in E^l_{X,Y}\right)\wedge\left(\Gamma_{x',y'}\neq\Gamma_{x,y}\right)\Longrightarrow(x'\not\subset x)\vee(y'\not\subset y)\end{array}\right\} \quad (1)$$

The symbols x' and y' in the formula are simply used as variables in the same way as x and y. The expression $(.\not\subset.)$ means that the first element is not included in the second one, either in terms of the composition of the subsequences or in terms of their respective positions in X. The matching set $E^l_{X,Y}$ will be used to compute the matching score of the sequence pair. What follows is an explanation of what the Formula (1) means:

- $|\Gamma_{x,y}|\geq l$: Means that each of the matched subsequences $\Gamma_{x,y}$ must have at least the minimum length l.
- $\left(\forall\Gamma_{x',y'}\in E^l_{X,Y}\right)\wedge\left(\Gamma_{x',y'}\neq\Gamma_{x,y}\right)\Longrightarrow(x'\not\subset x)\vee(y'\not\subset y)$: Means that for any matched subsequence $\Gamma_{x',y'}$ belonging to $E^l_{X,Y}$, $\Gamma_{x',y'}$ and $\Gamma_{x,y}$ being different implies that $\Gamma_{x',y'}$ is not included in $\Gamma_{x,y}$ according to the partial order induced by set inclusion. In other words, each of the $\Gamma_{x,y}$ in $E^l_{X,Y}$ is maximal.

To summarize, the formula means that the matching set $E^l_{X,Y}$ contains all the matched subsequences $\Gamma_{x,y}$ of maximal length (i.e., at least l) between the sequences X and Y. The use of logic expressions in the formula makes it very concise and easy to transform into conditions in a computer program.

The formula of $E^l_{X,Y}$ adequately describes some known properties of polypeptides and proteins. First, protein motifs (i.e., series of defined residues) determine the tendency of the primary structure to adopt a particular secondary structure, a property exploited by several secondary-structure prediction algorithms. Such motifs can be as short as four residues (for instance those found in β-turns), but the propensity to form an α-helix or a β-sheet is usually defined by longer motifs. Second, our proposal to take into account multiple (i.e., ≥ 2 residues) occurrences of a particular motif reflects the fact that sequence duplication is one of the most powerful mechanisms of gene and protein evolution, and if a motif is found twice (or more) in a protein it is more probable that it was acquired by duplication of a segment from a common ancestor than by acquisition from a distant ancestor.

The construction of $E^l_{X,Y}$ requires a CPU time proportional to $|X|*|Y|$. In practice, however, several optimizations are possible in the implementation, using encoding techniques to speed up this process. In our implementation of SMS, we used a technique that improved considerably the speed of the algorithm; we can summarize it as follows:

By the property that all possible matched subsequences satisfy $|\Gamma_{x,y}|\geq l$, we know that each $\Gamma_{x,y}$ in $E^l_{X,Y}$ is an expansion of a matched subsequence of length l. Thus, we first collect all

the matched subsequences of length l, which takes linear time. Secondly, we expand each of the matched subsequences as much as possible on the both left and right sides. Finally, we select all the expanded matched sequences that are maximal according to the inclusion criterion. This technique is very efficient for reducing the execution time in practice. However, due to the variable lengths of the matched sequences, it may not be possible to reduce the worst-case complexity to a linear time. In the Results section, we provide a time comparison between our algorithm and several existing ones.

Let M be a substitution matrix, and Γ a matched subsequence belonging to the matching set $E^{l}_{X,Y}$. We define a weight $W(\Gamma)$ for the matched subsequence Γ, to quantify its importance compared to all the other subsequences of $E^{l}_{X,Y}$, as follows:

$$W(\Gamma)=\sum_{i=1}^{|\Gamma|} M\left[\Gamma[i],\Gamma[i]\right] \tag{2}$$

Where $\Gamma[i]$ is the i^{th} amino acid of the matched subsequence Γ, and $W[\Gamma[i], \Gamma[i]]$ is the substitution score of this amino acid with itself. Here, in order to make our measure biologically plausible, we use the substitution concept to emphasize the relation that binds one amino acid with itself. The value of $M[\Gamma[i], \Gamma[i]]$ (i.e., within the diagonal of the substitution matrix) estimate the rate at which each possible amino acid in a sequence keep unchanged over time. For the pair of sequences X and Y, we define the matching score $s_{X,Y}$, understood as representing the substitution relation of the conserved regions in both sequences, as follows:

$$s_{X,Y}=\frac{\sum_{\Gamma\in E^{l}_{X,Y}} W(\Gamma)}{\max(|X|,|Y|)} \tag{3}$$

Finally, the pairwise similarity matrix S is calculated by applying the Pearson's correlation coefficient to the matrix s, as follows:

$$S_{X,Y}=\frac{\sum_{i=1}^{N}\left(s_{X,i}-\overline{s_X}\right)\left(s_{Y,i}-\overline{s_Y}\right)}{\sqrt{\sum_{i=1}^{N}\left(s_{X,i}-\overline{s_X}\right)^2}\sqrt{\sum_{i=1}^{N}\left(s_{Y,i}-\overline{s_Y}\right)^2}} \tag{4}$$

Where $S_{X,Y}$ is the similarity measure between the protein sequences X and Y, and $s_{X,i}$ and $s_{Y,i}$ are the matching scores between the protein sequence i with X and i with Y, and $\overline{s_X}$ and $\overline{s_Y}$ are the means of $s_{X,i}$ and $s_{Y,i}$ for all i values, respectively.

Our aim is to detect and make use of the significant motifs best conserved during evolution and to minimize the influence of those motifs that occur by chance. This motivates one of the major biological features of our similarity measure, the inclusion of all long conserved subsequences (i.e., multiple occurrences) in the matching, since it is well known that the longer the subsequences, the smaller the chance of their being identical by chance, and vice versa. Here we make use of the theory developed by Karlin *et al*. [20] to calculate, for each pair of sequences, the value of l, the minimum length of matched subsequences. According to theorem 1 of Karlin *et al*. in [20] we calculate $K_{r,N}$ as follows:

$$K_{r,N}=\frac{\log \mathrm{n}\left(|Q_1|,\cdots,|Q_{\mathrm{N}}|\right)+\log\lambda(1\text{-}\lambda)+0.577}{-\log\lambda} \tag{5}$$

$$\mathrm{n}\left(|Q_1|,\cdots,|Q_{\mathrm{N}}|\right)=\sum_{1\le i_1\le\cdots\le i_{\mathrm{r}}\le N}\prod_{v=1}^{r}\left|Q_{i_v}\right| \tag{6}$$

$$\lambda=\max_{1\le i_1\le\cdots\le i_r\le N}\left(\sum_{i=1}^{20}\prod_{j=1}^{r}p_i^{(\nu_j)}\right) \tag{7}$$

These formulas calculate $K_{r,N}$, the *expected length of the longest common word present by chance at least r times out of N m-letter sequences* [20] (i.e., $Q_1,\ldots,Q_N$), where $p_i^{(\nu)}$ is generally specified as the i^{th} residue frequency of the observed ν^{th} sequence.

According to the conservative criterion proposed by Karlin *et al.* [20], to measure the similarity between two protein sequences, we take into account in the calculation of our similarity measure SMS all subsequences present 2 times (i.e., r=2) out of the 2 sequences (i.e., N=2) which have a length that exceeds $K_{2,2}$ by at least two standard deviations. So, for each pair of protein sequences X and Y, we calculate a specific and appropriate value of l to calculate $S_{X,Y}$ the similarity measure.

III. The new clustering algorithm CLUSS

CLUSS is composed of three main stages. The first one consists in building a pairwise similarity matrix based on our new similarity measure SMS; the second, in building a phylogenetic tree according to the similarity matrix, using a hierarchical approach; and the third, in identifying subfamily nodes from which leaves are grouped into subfamilies.

A. Stage 1: Similarity matrix

Using one of the known substitution score matrices, such as BLOSUM62 [18] or PAM250 [19], and SMS our new similarity measure, we compute S, the (N x N) pairwise similarity matrix, where N is the number of sequences of the protein family F to be clustered, and $S_{i,j}$ is the similarity measure between the i^{th} and the j^{th} protein sequences of F. The construction of S takes CPU time proportional to $N(N\text{-}1)T^2/2$, with T the typical sequence length of the N sequences.

B. Stage 2: Phylogenetic tree

To build the phylogenetic tree, we have adopted a classical hierarchical approach. Starting from the protein sequences, each of which is considered as the root node of a subtree containing only one node, we iteratively join a pair of root nodes in order to build a bigger subtree. At each iteration, a pair of root nodes is selected if they are the most similar root nodes in terms of a similarity measure derived from the above similarity matrix S. This process ends when there remains only one subtree, which is the phylogenetic tree.

The similarity between two root nodes referred to above is computed in the following way. At the beginning of the

iteration, the similarity between any pair of nodes is initialized by the similarity matrix computed in Stage 1 (i.e., according to SMS). Let L and R be two nearest root nodes at a given iteration step; they are joined together to form a new subtree. Let P be the root node of the new subtree. P thus has two children, L and R. The similarity between the new root node P and any other root node K is defined as a weighted average of the similarity between the children of P and the node K:

$$S_{P,K}=\frac{d_L S_{L,K}+d_R S_{R,K}}{d_L+d_R} \tag{8}$$

Where $S_{L,K}$ and $S_{R,K}$ are the similarity values between the node K with L and K with R before the joining, and d_L and d_R are the numbers of leaves in the subtree rooted at L and R, respectively. Note that in order to keep the notation simple, $S_{P,K}$ is retained here to represent the similarity between any pair of nodes that do not have any descendant relationships in the phylogenetic tree.

C. Stage 3: Cluster extraction

Given F, a family of N protein sequences, after computing their similarity matrix and phylogenetic tree, CLUSS locates subfamily nodes in this tree using a procedure Ward's [21],[22] approach. The main idea is to extract from the phylogenetic tree a number of subtrees, each of which corresponds to a cluster, while optimizing a validation criterion. The criterion is in fact a trade-off between the within-cluster compactness and the between-cluster separation [23]. The different steps are summarized as follows:

1) Step 1 (Computing the weight of each node): First, each leaf node is considered as a subtree in the phylogenetic tree. We assign to each subtree L (i.e., an individual leaf represents one protein sequence) a weight W_L according to its importance in F. W_L depends on the number and closeness of the protein sequences that are in fact similar to L, and is thus intended to measure how well F is represented by this particular sequence. For this purpose, we make use of the Thompson [24] method in the definition of W_L:

$$W_L=\sum_{i\in\{\text{branch}(L\to P)-\{P\}\}}\frac{D_{\text{Parent}(i),i}}{d_{\text{Parent}(i)}} \tag{9}$$

Where P is the root of the phylogenetic tree, L a leaf in this tree, $branch(L\to P)-\{P\}$ the subset of nodes on the branch from L to P excluding P, $Parent(i)$ the parent of the node i, $D_{Parent(i),i}$ is the length of the branch connecting the node i to its parent (as defined in the previous phase), and $d_{Parent(i)}$ the number of leaves in the subtree rooted at the parent of i. According to this definition, the value of W_L is small if L is very representative and is large if L is not very representative. Iteratively, we assign to each internal subtree P the weight value W_P equal to the sum of the weights of its children W_L+W_R. We estimate the phylogenetic distance (i.e., this distance has no strict mathematical sense; it is merely a measure of the evolutionary distance between the nodes. It is closer to the notion of dissimilarity) from a node P to its children L and R as follows:

$$D_{L,P}=S_{L,R}\frac{W_R}{W_L+W_R} \tag{10}$$

$$D_{R,P}=S_{L,R}\frac{W_L}{W_L+W_R} \tag{11}$$

2) Step 2 (Computing co-similarity for all internal nodes): Iteratively, until the root of the phylogenetic tree is reached, we assign to the subtree rooted at each non-leaf node P the co-similarity value C_P (between its two child nodes), which is calculated according to the generalized Ward dissimilarity formula [21],[22] introduced by Batagelj [25], as follows:

$$C_P=S_{L,R}\frac{W_L W_R}{W_L+W_R} \tag{12}$$

Where W_L and W_R are the weights of L and R, respectively, and $S_{L,R}$ is the similarity between L and R computed in Stage 2.

By taking into account information about the neighborhood around each of the nodes L and R, the concept of co-similarity reflects the cluster compactness of all the sequences (leaf nodes) in the subtree. In fact, its value is inversely proportional to the within-cluster variance. When the subtree becomes larger, the co-similarity tends to become smaller, which means that the sequences within the subtree become less similar and the difference (separation) between sequences in different clusters becomes less significant.

3) Step 3 (Separating high co-similarity nodes from low co-similarity nodes): The CLUSS algorithm makes use of a systematic method for deciding which subtrees to retain as a trade-off between searching for the highest co-similarity values and searching for the largest possible clusters. We first separate all the subtrees into two groups, one being the group of high co-similarity subtrees and the other the low co-similarity subtrees. This is done by sorting all possible subtrees in increasing order of co-similarity and computing a separation threshold according to the method based on the maximum interclass inertia [26].

4) Step 4 (Extracting clusters): From the group of high co-similarity subtrees, we extract those that are largest. A high co-similarity subtree is largest if the following two conditions are satisfied: 1) it does not contain any low co-similarity subtree; 2) if it is included in another high co-similarity subtree, the latter contains at least one low co-similarity subtree. Each of these (largest) subtrees corresponds to a cluster and its leaves are then collected to form the corresponding cluster.

IV. Results

To illustrate its efficiency, we tested CLUSS extensively on a variety of protein datasets and databases and compared it with several mainstream clustering algorithms. We analyzed the results obtained for the different tests with support from the literature and functional annotations. Full data files and results cited in this section are available at CLUSS website.

A. The clustering quality measure

To highlight the functional characteristics and classifications of the clustered families, we introduce the $Q_{measure}$ which quantifies the quality of a clustering by measuring the percentage of correctly clustered protein sequences based on their known functional annotations. This measure can be easily adapted to any protein sequence database. The $Q_{measure}$ is defined as follows:

$$Q_{measure} = \frac{\left(\sum_{i=1}^{C} P_i\right) - U}{N} \quad (13)$$

Where N is the total number of clustered sequences, C is the number of clusters obtained, P_i is the largest number of sequences in the i^{th} cluster obtained belonging to the same function group according to the known reference classification, and U is the number of unclustered sequences. For the extreme case where each cluster contains one protein with all proteins classified as such, the $Q_{measure}$ is "0", since C becomes equal to N, and each P_i the largest number of obtained sequences in the i^{th} cluster is "1".

B. COG and KOG databases

To illustrate the efficiency of CLUSS in grouping protein sequences according to their functional annotation and biological classification, we performed extensive tests on the phylogenetic classification of proteins encoded in complete genomes, commonly named the Clusters of Orthologous Groups of proteins database (COG) [27]. As mentioned in the web site for the database, the COG (i.e., for unicellular organisms) and KOG (i.e., for eukaryotic organisms) clusters were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each COG and KOG consists of individual proteins or groups of paralogs from at least 3 lineages and each thus corresponds to an ancient conserved domain. COG and KOG contain (i.e., to date) 192,987 and 112,920 classified protein sequences, respectively.

To perform a biological and statistical evaluation of CLUSS, we randomly generated two sets of 1000 large subsets, one from the COG classification and the other one from the KOG classification. Each subset contains between 47 and 1840 non-orphan protein sequences (i.e., each selected protein sequence has at least one similar from the same functional classification) from at least 10 distinct groups from COG or KOG classification. We tested CLUSS on both sets of 1000 subsets using each of the substitution matrices BLOSUM62 [18] and PAM250 [19]. The average $Q_{measure}$ value of the clusterings obtained for the COG classification is superior to **90%** with a standard deviation of **6.24%**, and the value for the KOG classification is superior to **86%** with a standard deviation of **7.45%**. The results obtained show clearly that CLUSS is indeed effective in grouping sequences according to the known functional classification of COG and KOG databases.

In the aim of comparing the efficiency of CLUSS to that of alignment-dependent clustering algorithms, we performed tests using CLUSS, BlastClust [7], TRIBE-MCL [8] and gSPC [9] on the COG and KOG classifications. In all of the tests performed, we used the widely known protein sequence comparison algorithm ClustalW [28] to calculate the similarity measure matrices used by TRIBE-MCL [8] and gSPC [9]. Due to the complexity of alignment, these tests were done on two sets of six randomly generated subsets, named C1 to C6 for COG and K1 to K6 for KOG. The obtained results are summarized in Table I.

The results in Table I show clearly that CLUSS obtained the best $Q_{measure}$ compared to the other algorithms tested. Globally, the clusters obtained using our new algorithm CLUSS correspond better to the known characteristics of the biochemical activities and modular structures of the protein sequences according to COG and KOG classifications.

The execution time reported in Table I for algorithm comparison, show clearly that the fastest algorithm is BlastClust [7], closely followed by our algorithm CLUSS, while TRIBE-MCL [8] and gSPC [9], which use ClustalW [28] as similarity measures, are much slower than BlastClust [7].

TABLE I. *QUALITY MEASURES* (Q_M) AND EXECUTION TIMES (IN SECONDS) OBTAINED ON EACH COG AND KOG SUBSET. NUMBERS OF PROTEINS ARE INDICATED BETWEEN BRAKETS.

Protein subsets	CLUSS		BlastClust		T-MCL		gSPC	
	Q_m	*Time*	Q_m	*Time*	Q_m	*Time*	Q_m	*Time*
C1 (336)	95.21	116	81.25	10	92.26	332	93.45	340
C2 (214)	96.74	49	84.22	7	88.78	141	93.92	146
C3 (215)	91.34	74	87.50	14	83.68	273	73.26	285
C4 (355)	93.21	86	82.81	12	78.59	315	79.71	324
C5 (667)	97,56	667	94.00	105	63.46	5393	70.01	5338
C6 (309)	92.99	68	88.02	18	87.70	224	88.99	239
K1 (363)	93.38	414	67.21	44	69.69	1168	76.85	1209
K2 (425)	93.71	289	31.01	27	68.70	1208	53.64	1230
K3 (411)	91.25	258	42.33	55	74.85	270	75.91	325
K4 (360)	96.42	361	38.88	127	66.66	1123	67.22	1220
K5 (326)	95.67	221	77.91	33	75.46	688	82.51	718
K6 (590)	92,94	779	50.33	405	85.25	3782	66.94	4181

C. G-proteins family

The G-proteins [29] (i.e., for guanine nucleotide binding proteins) belong to the larger family of the GTPases. Their signaling mechanism consists in exchanging guanosine diphosphate (GDP) for guanosine triphosphate (GTP) as a general molecular function to regulate cell processes (reviewed extensively in [30]). This family has been the subject of a considerable number of publications by researchers around the world, so we considered it a good reference classification to test the performance of CLUSS. The sequences belonging to this family used in our experimentation are available at CLUSS website. The experimental results obtained using the algorithms CLUSS, BlastClust [7], TRIBE-MCL [8] and gSPC [9] are summarized in Table II. The clustering results for the G-protein family show clearly that although this family is known to be easy to align, which should have facilitated the clustering task of the alignment-dependent algorithms, CLUSS yields a clustering with Q-measure value of **92.74%**, the highest of all the algorithms tested. Thus, the results obtained by CLUSS are much closer to the known classification of the G-protein family than those of the other algorithms tested are.

We can make the same observation about the execution times of the different algorithms in Table II as in Table I.

TABLE II. *QUALITY MEASURE (Q_M)* AND EXECUTION TIMES (IN SECONDS) OBTAINED ON G-PROTEINS FAMILY.

Protein subset	CLUSS		BlastClust		T-MCL		gSPC	
	Qm	*Time*	*Qm*	*Time*	*Qm*	*Time*	*Qm*	*Time*
G-proteins 381 proteins	**92.7**	85	42.1	**14**	40.7	419	52.6	432

D. The 33 (α/β)8-barrel proteins

To show the performance of CLUSS with multi-domain protein families which are known to be hard to align and have not yet been definitively aligned, experimental tests were performed on the group of the 33 (α/β)8-barrel proteins, a group within Glycoside Hydrolases family 2 (GH2), from the CAZy database [29], studied recently by Côté *et al.* [32] and Fukamizo *et al.* [33]. The periodic character of the catalytic module known as "*(α/β)8-barrel*" makes these sequences hard-to-align using classical alignment approaches. The difficulties in aligning these modules are comparable to the problems encountered with the alignment of tandem-repeats, which have been exhaustively discussed by Higgins [11]. The FASTA file and clustering results of this subfamily are available on the CLUSS website. This group of 33 protein sequences includes "*β-galactosidase*", "*β-mannosidase*", "*β-glucuronidase*" and "*exo-β-D-glucosaminidase*" enzymatic activities, all extensively studied at the biochemical level. These sequences are multi-modular, with various types of modules, which complicate their alignment. Clustering such protein sequences using the alignment-dependent algorithms thus becomes problematic. In our experiments, we tested quite a few known algorithms to align the 33 protein sequences, such as MUSCLE [34], ClustalW [28], MAFFT [35], T-Coffee [36] etc. The alignment results of all these algorithms are in contradiction with those presented by Côté *et al.* [32] that in turn are supported by the structure-function studies of Fukamizo *et al.* [33]. This encouraged us to perform a clustering on this particular GH2 subfamily, to compare the behavior of CLUSS with BlastClust [7], TRIBE-MCL [8] and gSPC [9] in order to validate the use of CLUSS on the hard-to-align proteins. An overview of the results is given below. The corresponding names and database entries of the 33 (α/β)8-barrel proteins group are indicated at CLUSS website.

1) CLUSS results: The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by CLUSS into five subfamilies, organized in five main branches, details in Table III for corresponding cluster of each protein sequence and Figure 1 for the resulted phylogenetic tree. The first and the second branch correspond, respectively, to the first and the second clusters, which include enzymes with "*β-mannosidase*" activities; the third branch corresponds to the third cluster, which includes enzymes with "*β-glucuronidase*" activities; the fourth branch corresponds to the forth cluster, which includes enzymes with "*β-galactosidase*" activities; the fifth branch corresponds to the fifth cluster, which includes enzymes with "*exo-β-D-glucosaminidase*" activities.

2) BlastClust results: The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by BlastClust [7] into five subfamilies (see Table III). Almost all the enzymes were clustered in the appropriate clusters, except for seven proteins that were unclustered, among which we find the following well-classified enzymes: the "*β-galactosidase*" enzymes GaA, GaK and GaC; the "*β-mannosidase*" enzyme UnBc; and the "*exo-β-D-glucosaminidase*" enzyme CsAo.

3) Tribe-MCL results: The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by TRIBE-MCL [8] into two mixed subfamilies (see Table III). We find the "*β-mannosidase*" enzymes MaA, MaC and MaT grouped in the "*β-galactosidase*" subfamily. Furthermore, the "*exo-β-D-glucosaminidase*" and "*β-glucuronidases*" enzymes are grouped in the same subfamily.

4) gSPC results: The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by gSPC [9] into three subfamilies (see Table III). Almost all the enzymes were grouped in the appropriate subfamily, except for the "*β-galactosidases*" and the "*β-glucuronidases*" which were grouped in the same subfamily.

TABLE III. CLUSTERING RESULTS ON 33 $(\alpha/\beta)_8$-BARREL GROUP

#	Protein name	Côté & Fukamizo	CLUSS	BlastClust	T-MCL	gSPC
1	UnA	**1**	**1**	1	1	1
2	UnBv	**1**	**1**	1	1	1
3	UnBc	**1**	**1**	/	1	1
4	UnBm	**1**	**1**	1	1	1
5	UnBp	**1**	**1**	1	1	1
6	UnR	**1**	**1**	1	1	1
7	MaA	**2**	**2**	2	2	1
8	MaB	**2**	**2**	2	1	1
9	MaH	**2**	**2**	2	1	1
10	MaM	**2**	**2**	2	1	1
11	MaC	**2**	**2**	2	2	1
12	MaT	**2**	**2**	2	2	1
13	GIC	**3**	**3**	3	2	2
14	GIE	**3**	**3**	3	2	2
15	GIH	**3**	**3**	3	2	2
16	GIL	**3**	**3**	3	2	2
17	GIM	**3**	**3**	3	2	2
18	GIF	**3**	**3**	3	2	2
19	GIS	**3**	**3**	3	2	2
20	GaEco	**4**	**4**	4	2	2
21	GaA	**4**	**4**	/	2	2
22	GaK	**4**	**4**	/	2	2
23	GaC	**4**	**4**	/	2	2
24	GaEcl	**4**	**4**	4	2	2
25	GaL	**4**	**4**	4	2	2
26	CsAo	**5**	**5**	/	2	3
27	CsS	**5**	**5**	5	2	3
28	CsG	**5**	**5**	5	2	3
29	CsM	**5**	**5**	5	2	3
30	CsN	**5**	**5**	/	2	3
31	CsAn	**5**	**5**	/	2	3
32	CsH	**5**	**5**	5	2	3
33	CsE	**5**	**5**	5	2	3

Globally, the clustering of the 33 $(\alpha/\beta)_8$-barrel proteins generated by CLUSS corresponds better to the known characteristics of their biochemical activities and modular structures than do those yielded by the other algorithms tested. The results obtained with our new algorithm were highly comparable with those of the more complex, structure-based analysis performed by Côté *et al.* [32] and Fukamizo *et al.* [33].

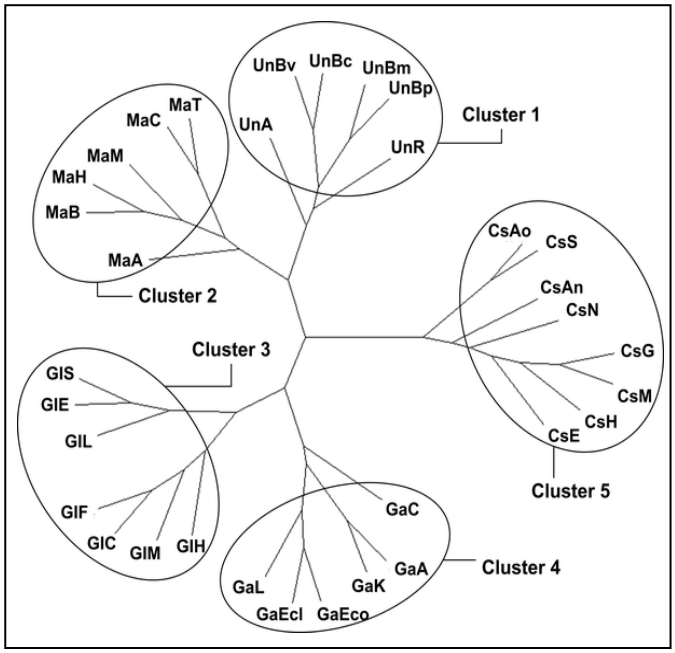

Figure 1. Phylogenetic tree of 33 $(\alpha/\beta)8$-barrel

V. DISCUSSION

The new similarity measure SMS presented in this paper makes possible to measure the similarity between protein sequences based solely on the conserved motifs. Its major advantage compared to the alignment-dependent approaches is that it gives significant results with protein sequences independent of their alignability, which allows it to be effective on both easy-to-align and hard-to-align protein families. This property is inherited by CLUSS, our new clustering algorithm, which uses it as its similarity measure. CLUSS used jointly with SMS is an effective clustering algorithm for protein sets with a restricted number of functions, which is the case of almost all protein families. It more accurately highlights the characteristics of the biochemical activities and modular structures of the clustered protein sequences than do the alignment-dependent algorithms.

So far, our similarity measure has been based on pre-determined substitution matrices. A possible future development is to propose an approach to automatically compute the weights of the conserved motifs instead of relying on pre-calculated substitution scores. There is also a need to speed up the extraction of the conserved motifs and the clustering of the phylogenetic tree, to scale the algorithm on datasets that are much larger in size with many more biological functions.

We believe that CLUSS is an effective method and tool for clustering protein sequences to meet the needs of biologists in terms of phylogenetic analysis and function prediction. In fact, CLUSS gives an efficient evolutionary representation of the phylogenetic relationships between protein sequences. This algorithm constitutes a significant new tool for the study of protein families, the annotation of newly sequenced genomes and the prediction of protein functions, especially for proteins with multi-domain structures whose alignment is not definitively established. Finally, the tool can also be easily adapted to cluster other types of genomic data.

REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, W. Myers, and D. J. Lipman, "Basic local alignment search tool," J. Mol. Bio., 215, pp. 403–410, 1990.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," Nucl. Acids Res., 25, pp. 3389–3402, 1997.

[3] A. Krause, J. Stoye, and M. Vingron, "The SYSTERS protein sequence cluster set.," Nucl. Acids Res., 28, pp. 270–272, 2000.

[4] P. Pipenbacher, A. Schliep, S. Schneckener, A. Schonhuth, D. Schomburg, and R. Schrader, "ProClust: Improved clustering of protein sequences with an extended graph-based approach," Bioinformatics, 18, pp. S182–S191, 2002.

[5] G. Yona, N. Linial, and M. Linial, "ProtoMap: Automatic classification of protein sequences and hierarchy of protein families," Nucl. Acids Res., 28, pp. 49-55, 2000.

[6] K. Sjölander, "Phylogenomic inference of protein molecular function: Advances and challenges," Bioinformatics, 20, pp. 170-179, 2000.

[7] Basic Local Alignment Search Tool: www.ncbi.nlm.nih.gov/BLAST.

[8] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," Nucl. Acids Res., 30, pp. 1575-1584, 2002.

[9] I. V. Tetko, A. Facius, A. Ruepp, and H. W. Mewes, "Super Paramagnetic Clustering of Protein Sequences," BMC Bioinformatics, 6, pp. 82, 2005.

[10] D. W. Mount, Bioinformatics. Sequence and Genome Analysis (2nd ed.), Cold Spring Harbor Laboratory Press, New York, 2004.

[11] D. Higgins, Multiple alignment. In The Phylogenetic Handbook. Edited by Salemi M, Vandamme A. M. Cambridge University Press 45, pp 45-71, 2004.

[12] J. S. Varré, J. P. Delahaye, and E. Rivals, "The transformation distance: A dissimilarity measure based on movements of segments," Bioinformatics, 15, pp. 194–202, 1999.

[13] E. L. L. Sonnhammer, and V. Hollich, "Scoredist: A simple and robust sequence distance estimator," BMC Bioinformatics, 6, pp 108, 2005.

[14] G. Reinert, S. Schbath, and M. S. Waterman, "Probabilistic and statistical properties of words: An overview," J. Comp. Biol., 7, pp. 1-46, 2000.

[15] R. C. Edgar, "Local homology recognition and distance measures in linear time using compressed amino acid alphabets," Nucl. Acids Res., 32, pp. 380-385, 2004.

[16] M. Kimura, "Evolutionary rate at the molecular level. Nature," 217, pp. 624–626, 1968.

[17] J. Felsenstein, "An alternating least squares approach to inferring phylogenies from pairwise distances," Syst. Biol., 46, pp. 101, 1997.

[18] S. Henikoff, and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," Proceedings of the National Academy of Sciences of the United States of America,"89, pp. 10915-10919, 1992.

[19] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt, "A model of evolutionary change in proteins," Atlas of Protein Sequence and Structure, vol. 5, suppl. 3, pp. 345-352, 1978.

[20] S. Karlin, and G. Ghandour, "Comparative statistics for DNA and protein sequences: Single sequence analysis," Proc. Natl. Acad. Sci. USA, 82, pp. 5800-5804, 1985.

[21] J. H. Ward, "Hierarchical Grouping to Optimize an Objective Function," J. Am. Stat. Assoc. 58, pp. 236-244, 1963.

[22] J. H. Ward, and M. E. Hook, "Application of a Hierarchical Grouping Procedure to a Problem of Grouping Profiles," Educ. Psychol. Meas,, 23, pp. 69-82, 1963.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification, second edition," John Wiley and Sons, 2001.

[24] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "Improved sensitivity of profile searches through the use of sequence weights and gap excision," Comput. Appl. Biosci., 10, pp. 19–29, 1994.

[25] V. Batagelj, Generalized Ward and related clustering problems. In Classification and Related Methods of Data Analysis, edited by H. H. Bock, Amsterdam: Elsevier, pp. 67–74, 1998.

[26] N. Wicker, G. R. Perrin, J. C. Thierry, and O. Poch, "Secator: A Program for Inferring Protein Subfamilies from Phylogenetic Trees," Mol. Biol. Evol., 18, pp. 1435-1441, 2001.

[27] Phylogenetic classification of proteins encoded in complete genomes: www.ncbi.nlm.nih.gov/COG.

[28] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," Nucl. Acids Res., 22, pp. 4673-4680, 1994.

[29] GPCRIPDB: Information system for GPCR interacting proteins: www.gpcr.org.

[30] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, L. Zipursky and J. Darnell, Molecular Cell Biology, 5th ed. New York and Basingstoke: W.H. Freeman and Co., 2004.

[31] The carbohydrate-active enzymes (CAZy) database: www.cazy.org/.

[32] N. Côté, A. Fleury, E. Dumont-Blanchette, T. Fukamizo, M. Mitsutomi, and R. Brzezinski, "Two exo-β-D-glucosaminidases/exochitosanases from actinomycetes define a new subfamily within family 2 of glycoside hydrolases," Biochem. J., 394, pp. 675–686, 2006.

[33] T. Fukamizo, A. Fleury, N. Côté, M. Mitsutomi, and R. Brzezinski, "Exo-β-D-glucosaminidase from Amycolatopsis orientalis: Catalytic residues, sugar recognition specificity, kinetics, and synergism," Glycobiology, 16, pp. 1064-1072, 2006.

[34] R. C. Edgar, "MUSCLE: A multiple sequence alignment method with reduced time and space complexity," BMC Bioinformatics, 5, pp. 113, 2004.

[35] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, "MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform," Nucl. Acids Res., 30, pp. 3059-3066, 2002.

[36] C. Notredame, D. Higgins, and J. Heringa, "T-Coffee: A novel method for multiple sequence alignments," Journal of Molecular Biology, 302, pp. 205-217, 2000.