BioData Mining

BioMed Central

Research

Partitioning clustering algorithms for protein sequence data sets

Sondes Fayech*, Nadia Essoussi and Mohamed Limam

Address: Department of Computer Science, LARODEC Laboratory, Higher Institute of Management, University of Tunis, Tunis, Tunisia

Email: Sondes Fayech* - sondes_el_feyech@yahoo.fr; Nadia Essoussi - nadia.essoussi@isg.rnu.tn;
Mohamed Limam - mohamed.limam@isg.rnu.tn

* Corresponding author

Published: 2 April 2009

BioData Mining 2009, **2**:3 doi:10.1186/1756-0381-2-3

Received: 13 November 2008
Accepted: 2 April 2009

This article is available from: http://www.biodatamining.org/content/2/1/3

Abstract

Background: Genome-sequencing projects are currently producing an enormous amount of new sequences and cause the rapid increasing of protein sequence databases. The unsupervised classification of these data into functional groups or families, clustering, has become one of the principal research objectives in structural and functional genomics. Computer programs to automatically and accurately classify sequences into families become a necessity. A significant number of methods have addressed the clustering of protein sequences and most of them can be categorized in three major groups: hierarchical, graph-based and partitioning methods. Among the various sequence clustering methods in literature, hierarchical and graph-based approaches have been widely used. Although partitioning clustering techniques are extremely used in other fields, few applications have been found in the field of protein sequence clustering. It is not fully demonstrated if partitioning methods can be applied to protein sequence data and if these methods can be efficient compared to the published clustering methods.

Methods: We developed four partitioning clustering approaches using Smith-Waterman local-alignment algorithm to determine pair-wise similarities of sequences. Four different sets of protein sequences were used as evaluation data sets for the proposed methods.

Results: We show that these methods outperform several other published clustering methods in terms of correctly predicting a classifier and especially in terms of the correctness of the provided prediction. The software is available to academic users from the authors upon request.

Background

In bioinformatics, the number of protein sequences is more than half a million, and it is necessary to find meaningful partitions of them in order to detect their functions. Early approaches of comparing and grouping protein sequences are alignment methods. In fact, pair-wise alignment is used to compare and to cluster sequences. There are two types of pair-wise sequence alignments, local and global [1,2]. Smith and Waterman local alignment algorithm [3] helps in finding conserved amino acid patterns in protein sequences. Needleman and Wunsch global alignment algorithm [4] attempts are made to align the entire sequence using as many characters as possible, up to both ends of each sequence. In order to cluster a large data set of proteins into meaningful clusters, the pair-wise alignment is computationally expensive because of the large number of comparisons carried out. In fact, each protein of the data set should be compared to all others of the data set.

For this reason the pair-wise alignment methods are not efficient to cluster a large set of data. These approaches do not consider the fact that the data set can be too large and may not fit into the main memory of some computers.

The main objective of the unsupervised learning technique is to find a natural grouping or meaningful partition using a distance function [5,6]. Clustering is a technique which has been extensively applied in a variety of fields related to life science and biology. In sequence analysis, clustering is used to group homologous sequences into gene or protein families.

Many methods are currently available for the clustering of protein sequences into families and most of them can be categorized in three major groups: hierarchical, graph-based and partitioning methods. Among these various methods, most are based on hierarchical or graph-based techniques and they were successfully established. In fact, COG [7] uses a hierarchical merging of clusters and a manual validation to prevent chaining of multi-domain families. ProtoNet [8] uses a special metric as described by Sasson et al. [9] to merge clusters. Picasso [10] uses multiple alignments as profiles which are then merged hierarchically. ClusTr [11] and GeneRage [12] use standard single linkage clustering approaches. SYSTERS [13] combines hierarchical clustering with graph-based clustering. ProtoMap [14] and N-cut [15,16] methods use graph-based clustering approaches. ProClust [17] uses an extension of the graph-based clustering approach proposed by [18]. ProClust algorithm is based on transitivity criterion and it is capable of handling multi-domain proteins. TribeMCL [19] applies the Markov clustering approach (MCL) described by Van Dongen [20]. This method operates on a graph that contains similarity information obtained by pair-wise alignment of sequences.

A small amount of partitioning techniques is used in the protein sequence clustering field. Guralnik and Karypis [21] have proposed one method based on a standard k-means approach where proteins are represented by vectors. However, no tool or database resulting from this interesting work has been made available to the scientific community. JACOP [22] uses the partitioning algorithm implemented under the name PAM (Partitioning Around Medoids) in the R statistical package [23]. JACOP is based on a random sampling of sequences into groups. It is available on the MyHits platform [24] where user can submit his own data set. Methods presented bellow are not generally tools since they cannot be applied to cluster a user-provided data set. In fact, several of these methods have been applied to large known data sets and user can only consult the resulting classifications stored in databases. Among the protein sequence clustering methods defined bellow only ProClust, TribeMCL and JACOP are accessed by the community and user can classify his own sequence set.

The main idea here is to design and develop efficient clustering algorithms based on partitioning techniques, which are not very investigated in protein sequence clustering field, in order to cluster large sets of protein sequences. In fact, the number of protein sequences available now is very important (in the order of millions) and hierarchical methods are computationally very expensive so they cannot be extended to cluster large protein sets. However, partitioning methods are very simple and more appropriate to cluster large data sets [22]. For these reasons, we propose here new clustering algorithms based on partitioning techniques which aim to find meaningful partitions, to improve the classification's quality and to reduce the computation time compared to the published clustering tools, ProClust, TribeMCL and JACOP, on different data sets.

Several partitioning clustering algorithms have been proposed in literature. K-means [23,25-27] is a standard partitioning clustering method based on K centroids of a random initial partition which is iteratively improved. LEADER [28,29] is an incremental partitioning clustering algorithm in which each of the K clusters is represented by a leader. CLARA (Clustering LARge Applications) [23] is a partitioning algorithm based on a combination of a sampling approach and the PAM algorithm. CLARANS (Clustering Large Applications based on RANdomized Search) [30] algorithm views the process of finding optimal medoids as searching through a certain graph, in which each node represents a set of medoids.

We adapted the partitioning algorithms cited bellows to protein sequence data sets. These proposed algorithms are named: Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS. Performance measures are used to evaluate the proposed methods and to compare them with ProClust, TribeMCL and JACOP results.

Methods

Algorithms Implementation

To facilitate subsequent discussion, the main symbols used through the paper and their definitions are summarized in Table 1.

Table 1: Summary of symbols and definitions

Symbols	Definitions
D	Data set of protein sequences to be clustered
K	Number of clusters
n	Number of proteins in *D*
O_i	a protein sequence *i* in *D*
q	Number of iterations

The main objective in the proposed algorithms Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS is to produce K clusters from a data set D of n protein sequences, so that the objective function f(V) is maximized.

f(V) is the global score function that evaluates the clustering quality and it is as follows

$$f(V) = \sum_{j \in [1..n]} Score(O_j, R_i), \tag{1}$$

Where R_i is the centroid of the group i for which belong the object O_j and Score (O_j, R_i) is the alignment score of the protein sequences O_j and R_i, calculated as follows

$$Score(A, B) = \sum_{i,j} S(A_i, B_j) - \sum_n g(n), \tag{2}$$

Where *S* (A_i, B_j) is the substitution score of the amino acid A_i by B_j as determined from a scoring matrix and *g(n)* is the total cost of penalties for a gap length *n*. The gap is defined as follows

$$g(n) = P_o + (n-1) * P_e, \tag{3}$$

Where P_o is the gap opening penalty and P_e is the gap extension penalty.

We chose Smith and Waterman local alignment algorithm for computing alignment score. The choice of this algorithm was motivated by the sensitivity for low-scoring alignments [31] compared to heuristic algorithms such as FASTA [32] and BLAST [33], and by execution time [34] compared to Needleman and Wunsch global alignment algorithm.

We present here Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS partitioning clustering algorithms for protein sequence sets.

Pro-Kmeans algorithm

The Pro-Kmeans algorithm proposed here, starts by a random partition of the data set D into K clusters and then uses the Smith Waterman algorithm to compare proteins of each cluster $S_{i(i \in [1..K])}$ and to compute SumScore(S_i, O_j) of each protein j in S_i as follows

$$SumScore(S_i, O_j) = \sum_{w \in [1..m] \neq j} Score(O_j, O_w), \tag{4}$$

Where *m* is the size of the subset S_i, for which belongs the object O_j.

The sequence O_j in each cluster S_i which has the maximum *SumScore*(S_i, O_j) is considered as the centroid R_i of the cluster. The Smith Waterman algorithm is used here also to compare each protein O_h of the data set *D* with centroids and to assign the object to the nearest cluster where the R_i have the maximum score of similarity with the object O_h. Pro-Kmeans proceeds to this procedure for a number of times, *q*, in order to maximize the *f(V)* function. Input parameters are the number of clusters, *K*, and of iterations, *q*, and as outputs the algorithm returns the best partition of the training base *D* and the center, or mean, of each cluster S_i. Pro-Kmeans algorithm is illustrated in Figure 1.

Pro-LEADER algorithm

Pro-LEADER is an incremental algorithm which selects the first sequence of the data set *D* as the first leader, and use the Smith Waterman algorithm to compute the similarity score of each sequence in *D* with all leaders. The algorithm detects the nearest leader R_i to each sequence O_j and compares the score, *Score*(R_i, O_j), with a pre-fixed *Threshold*. If the similarity score of R_i and O_j, is more than the *Threshold*, O_j is considered as a new leader and if not, the sequence O_j is assigned to the cluster defined by the leader R_i. Pro-LEADER is thus an incremental algorithm in which each of the *K* clusters is represented by a leader. The *K* clusters are generated using a suitable *Threshold* value. Pro-LEADER aims also to maximize the *f(V)* function. Input parameter is the similarity score *Threshold* to consider an object O_j as a new leader, and as outputs the algorithm returns the best partition of the training base *D* and the *K* leaders of the obtained clusters. The Pro-LEADER algorithm is fast, requiring only one pass through the data set *D*. Pro-LEADER algorithm is depicted in Figure 2.

Pro-CLARA algorithm

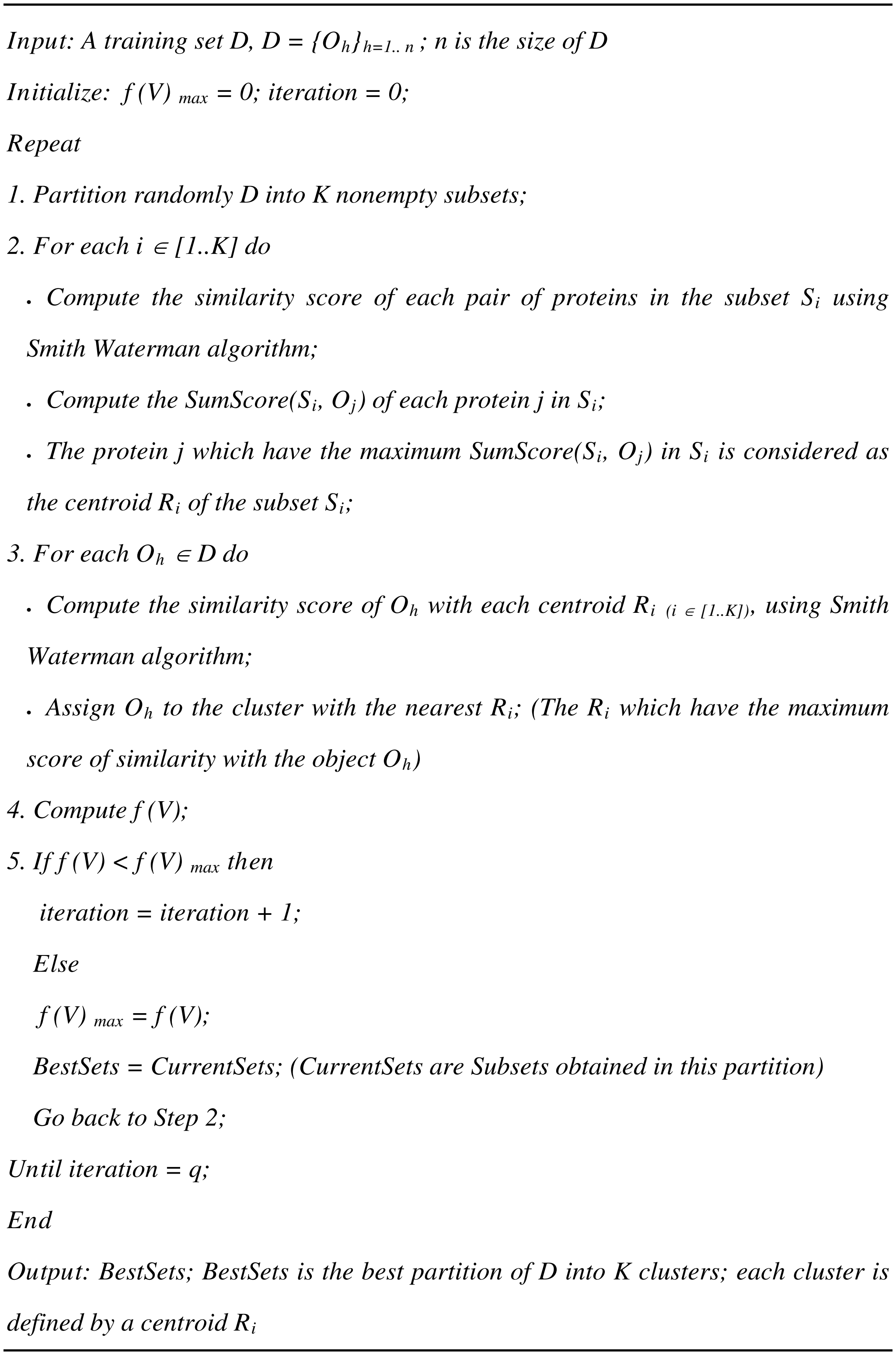
Pro-CLARA relies on the sampling approach to handle large data sets [23]. Instead of finding medoids for the entire data set, Pro-CLARA algorithm draws a small sample *S* of 40 + 2*K* sequences from the data set *D*. To generate an optimal set of medoids for this sample, Pro-CLARA applies the proposed PAM algorithm for protein sequence data sets, Pro-PAM algorithm,

The Pro-PAM algorithm proposed here, selects randomly K sequences from the data set as clusters, and then use the Smith Waterman algorithm to compute the total score TS_{ih} of each pair of selected sequence R_i and non selected sequence O_h. TS_{ih} is as follows

$$TS_{ih} = \sum_{j \in [1..m]} S_{jih}, \tag{5}$$

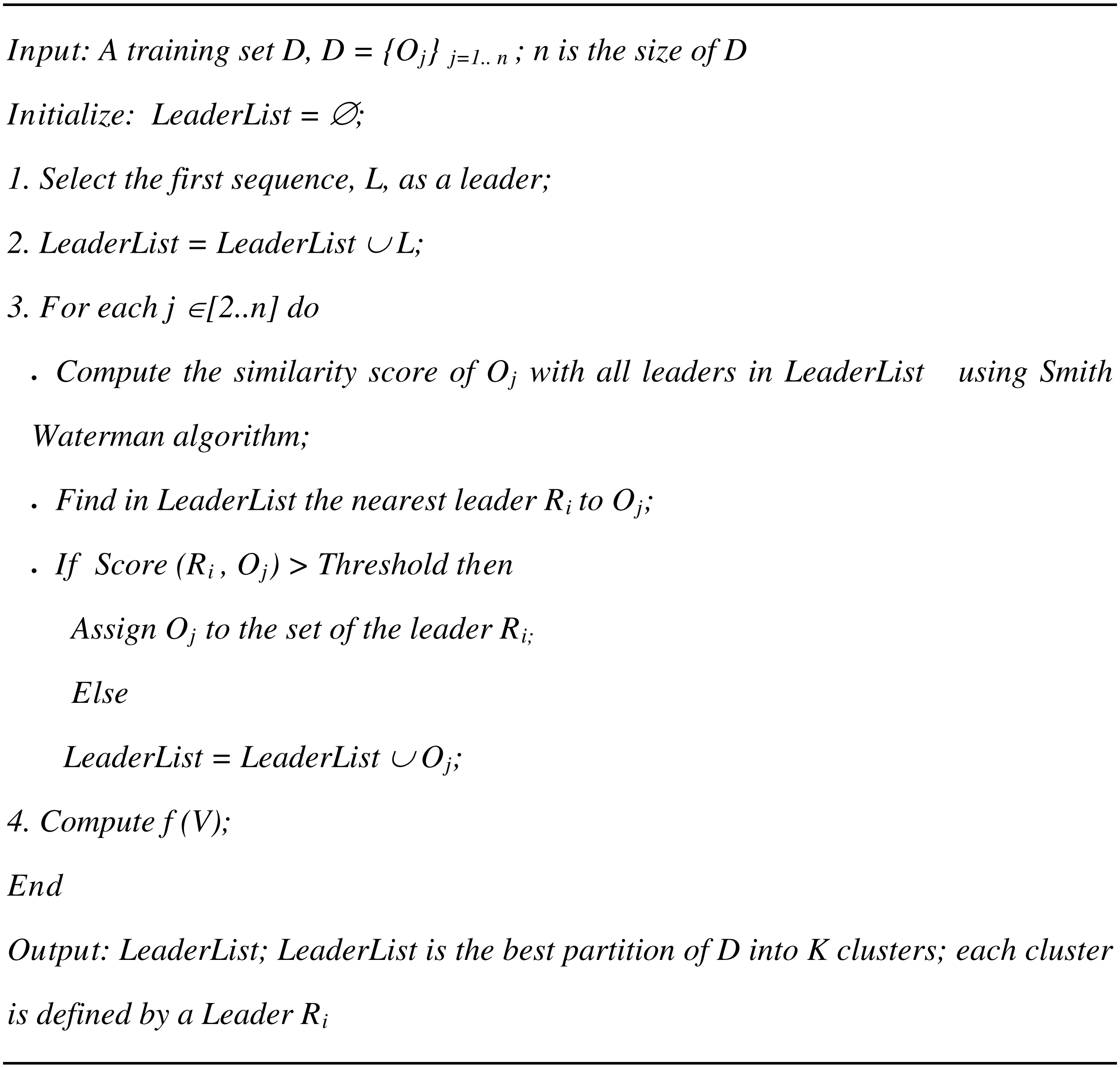
Where S_{jih} is the differential score of each pair of non-selected object O_h in *D* and selected object $R_{i(i \in [1..K])}$ with all non-selected objects O_j in *D*. S_{jih} is as follows

$$S_{jih} = Score(O_j, O_h) - Score(O_j, R_i); (O_j \neq R_{i(i \in [1..k])}). \tag{6}$$



```
Input: A training set D, $D = \{O_h\}_{h=1..n}$; n is the size of D
Initialize: $f(V)_{max} = 0$; iteration = 0;
Repeat
1. Partition randomly D into K nonempty subsets;
2. For each $i \in [1..K]$ do
   . Compute the similarity score of each pair of proteins in the subset $S_i$ using Smith Waterman algorithm;
   . Compute the SumScore($S_i$, $O_j$) of each protein j in $S_i$;
   . The protein j which have the maximum SumScore($S_i$, $O_j$) in $S_i$ is considered as the centroid $R_i$ of the subset $S_i$;
3. For each $O_h \in D$ do
   . Compute the similarity score of $O_h$ with each centroid $R_{i\ (i \in [1..K])}$, using Smith Waterman algorithm;
   . Assign $O_h$ to the cluster with the nearest $R_i$; (The $R_i$ which have the maximum score of similarity with the object $O_h$)
4. Compute $f(V)$;
5. If $f(V) < f(V)_{max}$ then
      iteration = iteration + 1;
   Else
      $f(V)_{max} = f(V)$;
   BestSets = CurrentSets; (CurrentSets are Subsets obtained in this partition)
   Go back to Step 2;
Until iteration = q;
End
Output: BestSets; BestSets is the best partition of D into K clusters; each cluster is defined by a centroid $R_i$
```

Figure 1
Pseudo code for Pro-Kmeans algorithm.

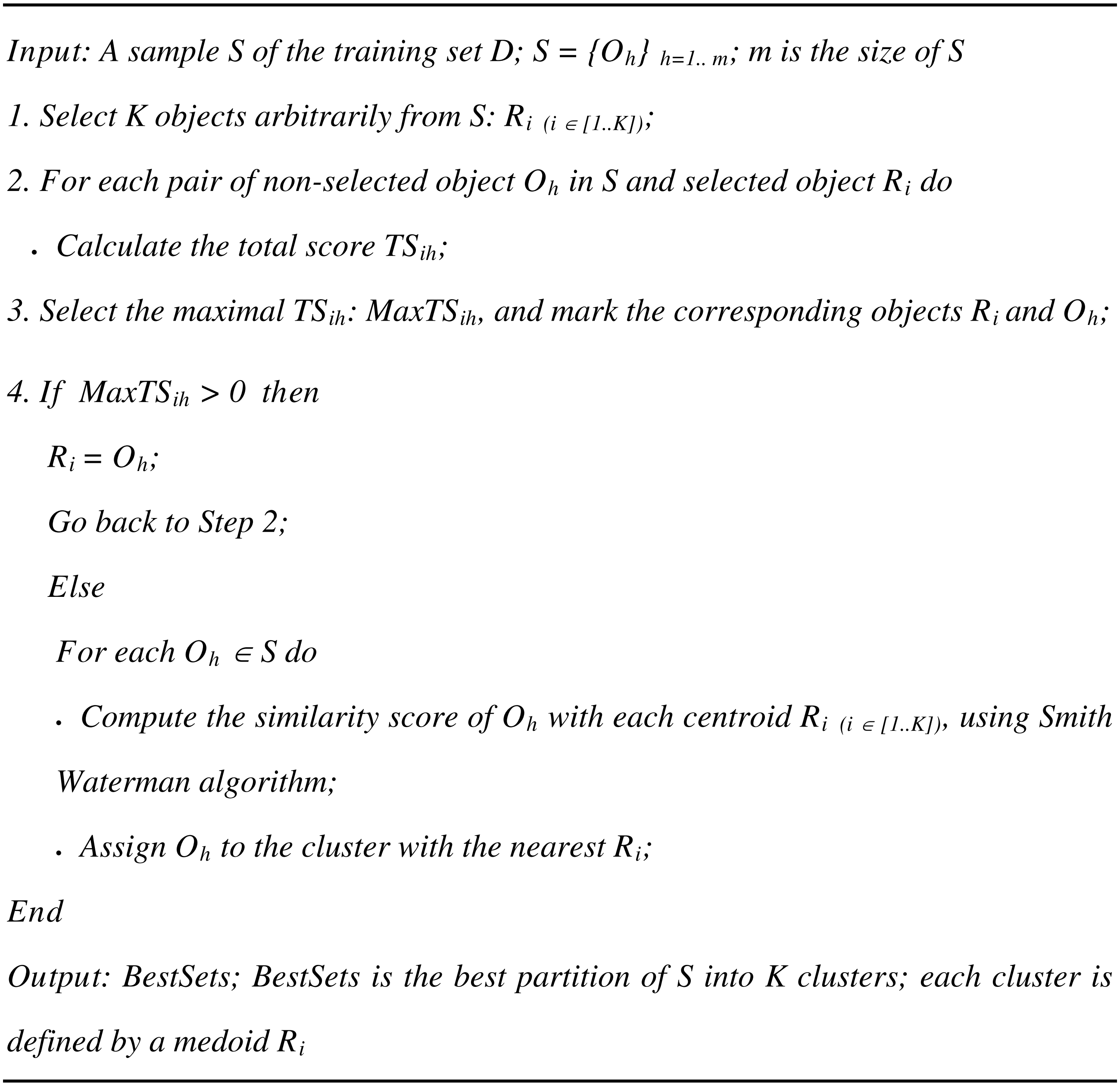


```
Input: A training set D, $D = \{O_j\}_{j=1..n}$; n is the size of D
Initialize: LeaderList = $\varnothing$;
1. Select the first sequence, L, as a leader;
2. LeaderList = LeaderList $\cup$ L;
3. For each $j \in [2..n]$ do
   • Compute the similarity score of $O_j$ with all leaders in LeaderList using Smith
     Waterman algorithm;
   • Find in LeaderList the nearest leader $R_i$ to $O_j$;
   • If Score ($R_i$, $O_j$) > Threshold then
        Assign $O_j$ to the set of the leader $R_i$;
        Else
       LeaderList = LeaderList $\cup$ $O_j$;
4. Compute f (V);
End
Output: LeaderList; LeaderList is the best partition of D into K clusters; each cluster
is defined by a Leader $R_i$
```

Figure 2
Pseudo code for Pro-LEADER algorithm.

Pro-PAM selects the maximal TS_{ih}, $MaxTS_{ih}$. If $MaxTS_{ih}$ is positive, the corresponding non selected sequence O_h will be selected, otherwise Smith Waterman algorithm is used to compare each protein O_h of the data set with all medoids $R_{i(i \in [1..K])}$, and to assign the sequence O_h to the nearest cluster. Input parameter of Pro-PAM is the number of clusters, *K*, and as outputs the algorithm returns the best partition of the protein sequence base and the medoid of each cluster. Pro-PAM algorithm is depicted in Figure 3.

Pro-CLARA uses the optimal set of medoids $R_{i(i \in [1..K])}$ obtained by Pro-PAM and the Smith Waterman algorithm to compare each protein O_h of the data set *D* with all medoids $R_{i(i \in [1..K])}$, and to assign the sequence O_h to the nearest cluster. In order to alleviate sampling bias, Pro-CLARA repeats the sampling and the clustering process a pre-defined number of times, *q*, and subsequently selects as the final clustering result the set of medoids with the maximal *f (V)*. Input parameters of Pro-CLARA algorithm are the number of clusters, *K*, and of iterations, *q*, and as



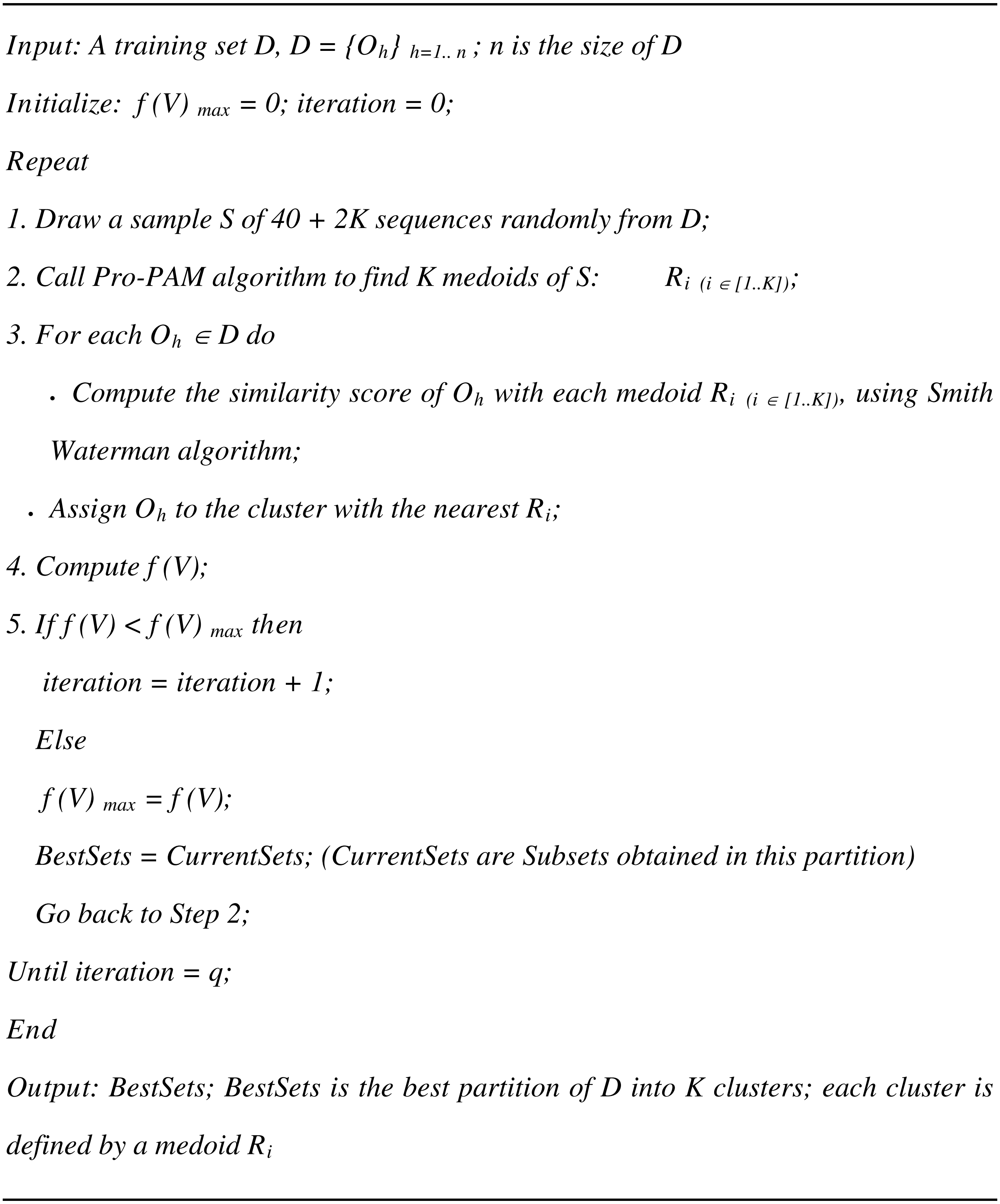
```
Input: A sample S of the training set D; $S = \{O_h\}_{h=1..m}$; m is the size of S
1. Select K objects arbitrarily from S: $R_{i\ (i \in [1..K])}$;
2. For each pair of non-selected object $O_h$ in S and selected object $R_i$ do
   • Calculate the total score $TS_{ih}$;
3. Select the maximal $TS_{ih}$: $MaxTS_{ih}$, and mark the corresponding objects $R_i$ and $O_h$;
4. If $MaxTS_{ih} > 0$ then
     $R_i = O_h$;
     Go back to Step 2;
   Else
     For each $O_h \in S$ do
       • Compute the similarity score of $O_h$ with each centroid $R_{i\ (i \in [1..K])}$, using Smith Waterman algorithm;
       • Assign $O_h$ to the cluster with the nearest $R_i$;
End
Output: BestSets; BestSets is the best partition of S into K clusters; each cluster is defined by a medoid $R_i$
```

Figure 3
Pseudo code for Pro-PAM algorithm.

outputs the algorithm returns the best partition of the training base *D* and the *K* medoids of the obtained clusters. Pro-CLARA algorithm is detailed in Figure 4.

Pro-CLARANS algorithm

Pro-CLARANS algorithm starts from an arbitrary node *C* in the graph, $C = [R_1, R_2, ..., R_k]$, which represents an initial set of medoids. Pro-CLARANS randomly selects one of *C* neighbors, *C**, which differs by only one sequence. If the total score of the selected neighbour, TS_{ih} (Equation (5)), is higher than that of the current node TS'_{ih}, Pro-CLARANS proceeds to this neighbor and continues the neighbor selection and comparison process. Otherwise, Pro-CLARANS randomly checks another neighbor until a better neighbor is found or the pre-determined maximal number of neighbours to check, *Maxneighbor*, has been reached. In this study *Maxneighbor* is defined as proposed by [30]



```
Input: A training set D, $D = \{O_h\}_{h=1..n}$; n is the size of D
Initialize: $f(V)_{max} = 0$; iteration = 0;
Repeat
1. Draw a sample S of 40 + 2K sequences randomly from D;
2. Call Pro-PAM algorithm to find K medoids of S:    $R_{i\ (i \in [1..K])}$;
3. For each $O_h \in D$ do
     • Compute the similarity score of $O_h$ with each medoid $R_{i\ (i \in [1..K])}$, using Smith Waterman algorithm;
   • Assign $O_h$ to the cluster with the nearest $R_i$;
4. Compute $f(V)$;
5. If $f(V) < f(V)_{max}$ then
     iteration = iteration + 1;
   Else
   $f(V)_{max} = f(V)$;
   BestSets = CurrentSets; (CurrentSets are Subsets obtained in this partition)
   Go back to Step 2;
Until iteration = q;
End
Output: BestSets; BestSets is the best partition of D into K clusters; each cluster is defined by a medoid $R_i$
```

Figure 4
Pseudo code for Pro-CLARA algorithm.

$$Maxneighbor = Max((1.25\% * K * (n - K)); 250). \quad (7)$$

Where the maximal number of neighbours must be at least a threshold value 250 or obtained using the number of clusters *K* and the number of sequences, *n*, in the data set as: *1.25%*K*(n-K)*.

Pro-CLARANS algorithm aims to maximize the total score, TS_{ih}

Pro-CLARANS algorithm use then the Smith Waterman algorithm to compute the similarity score of each sequence O_h in D with each medoid $R_{i(i \in [1..K])}$ and to assign it to the nearest cluster. The algorithm repeats the clustering process a pre-defined number of times, *q*, and selects as the final clustering result the set of medoids with the maximal *f (V)*. Input parameters of Pro-CLARANS algorithm are the number of clusters, *K*, and of iterations, *q*, and as outputs the algorithm returns the best partition of the training base *D* and the *K* medoids of the obtained clusters. Pro-CLARANS algorithm is detailed in Figure 5.

The proposed algorithms presented here have been implemented in Java package. All of these algorithms used the EMBOSS ftp://emboss.open-bio.org/pub/EMBOSS/ implementation of the Smith and Waterman local alignment algorithm for computing alignment score.

BLOSUM62 (Blocks Substitution Matrix) [35] was chosen to compute amino acids substitution scores [36]. We chose the default penalties proposed by Smith and Waterman EMBOSS implementation as gap opening (P_o) and gap extension penalties (P_e) (P_o = 10 and P_e = 2).

Performance measure

To evaluate the Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS clustering algorithms, a large data set, Training data set, is used. We obtained from the training phase *K* clusters and each cluster is defined by a medoid (centroid or leader). The training phase results are used to cluster a different data set named, Test data set. Smith Waterman algorithm is used to compare each protein sequence on the test data set with all medoids $R_{i(i \in [1..K])}$ obtained from the training phase, and to assign each sequence to the nearest cluster. The predicted family group of each sequence is which of the nearest medoid.

The results obtained from the test phase are used to calculate the *Sensitivity* and the *Specificity* of each algorithm and to compare them with results of the published clustering tools, ProClust, TribeMCL and JACOP, tested on the same set: "Test data set".

Sensitivity specifies the probability of correctly predicting a classifier and it is defined as

$$Sensitivity = TP/(TP + FN), \quad (8)$$

and *Specificity* the probability that the provided prediction is correct and it is defined as

$$Specificity = TP/(TP + FP), \quad (9)$$

where *TP* (True Positives) is the number of correctly identified true homologues pairs, *FN* (False Negatives) is the number of not identified true homologues pairs and *FP* (False Positives) is the number of non-homologue pairs predicted to be homologue. A pair of sequences is considered truly homologous, if both are in the same family group.

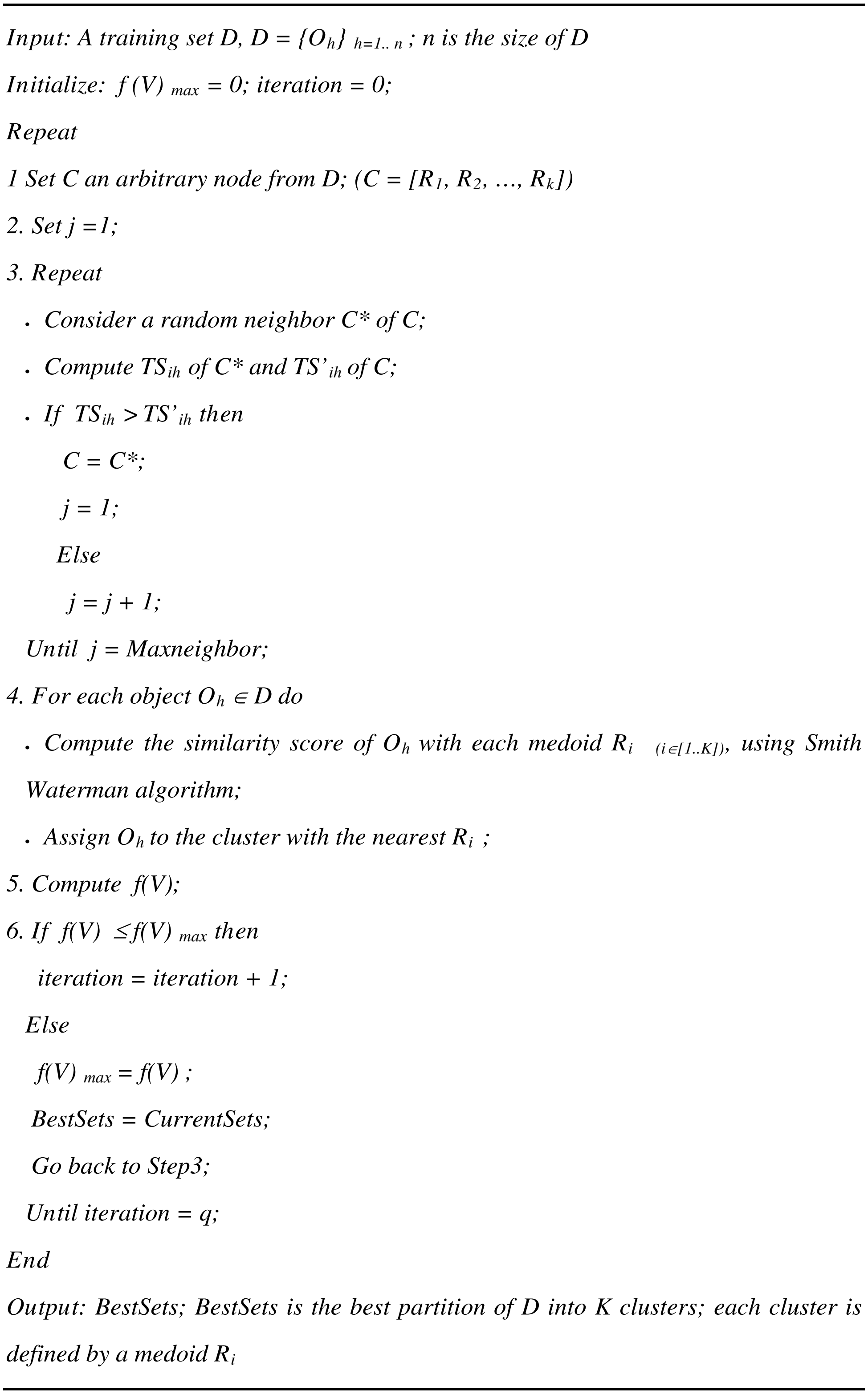
Protein sequence data sets

To evaluate the performance of the proposed clustering algorithms Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS, and to compare their results with the available graph-based clustering tools ProClust and TribeMCL and the only available partitioning clustering tool JACOP, protein sequence families with known subfamilies/groups are considered. Protein sequences of HLA protein family have been collected from ftp://ftp.ebi.ac.uk/pub/databases/imgt/mhc/hla. From this set, we have randomly selected 893 sequences named DS1 and grouped into 12 classes. Protein sequences of Hydrolases protein family have been collected from http://www.brenda-enzymes.org/. Hydrolases protein family sequences are categorized into 8 classes according to their function and 3737 sequences, named DS2, have been considered from this family. From Globins protein family [37], sequences have been collected randomly from 8 different classes and 292 sequences, named DS3, have been selected from the data set *IPR000971* in http://srs.ebi.ac.uk. Thus, totally 28 different classes containing sequences are considered as they have been classified by scientists/experts.

The data set considered has a total of 4922 sequences, named DS4, out of which 3500 sequences (practically 70% of the dataset DS4) are randomly for training, and 1422 for testing (practically 30% of the dataset DS4). The same method to obtain the training and the test sets are used on DS1, DS2 and DS3: randomly 70% of the set is selected for the training set and 30% for the test set [see Additional file 1].

Results and Discussion

Experiments are conducted on Intel Pentium4 processor based machine, having a clock frequency of 2.4 GHZ and 512 MB of RAM. Experimental results are obtained using default values as follows. In Pro-Kmeans, Pro-CLARA and Pro-CLARANS algorithms, the number of iterations *q* is fixed to 5 [30] and the number of clusters *K* is fixed to 28. After a number of simulations, we find that the best clus-



```
Input: A training set D, $D = \{O_h\}_{h=1..n}$ ; n is the size of D
Initialize: $f(V)_{max} = 0$; iteration = 0;
Repeat
1 Set C an arbitrary node from D; ($C = [R_1, R_2, \ldots, R_k]$)
2. Set j =1;
3. Repeat
  • Consider a random neighbor C* of C;
  • Compute $TS_{ih}$ of C* and $TS'_{ih}$ of C;
  • If $TS_{ih} > TS'_{ih}$ then
       C = C*;
       j = 1;
     Else
       j = j + 1;
  Until j = Maxneighbor;
4. For each object $O_h \in D$ do
  • Compute the similarity score of $O_h$ with each medoid $R_i$ $_{(i \in [1..K])}$, using Smith
  Waterman algorithm;
  • Assign $O_h$ to the cluster with the nearest $R_i$ ;
5. Compute f(V);
6. If $f(V) \leq f(V)_{max}$ then
     iteration = iteration + 1;
  Else
     $f(V)_{max} = f(V)$ ;
     BestSets = CurrentSets;
     Go back to Step3;
  Until iteration = q;
End
Output: BestSets; BestSets is the best partition of D into K clusters; each cluster is
defined by a medoid $R_i$
```

Figure 5
Pseudo code for Pro-CLARANS algorithm.

Table 2: Performance of the three other tools (ProClust, TribeMCL and JACOP) and our four proposed methods on DS1, DS2, DS3 and DS4 data sets with respect to two clustering quality measurements: Sensitivity (Sens.) and Specificity (Spec.)

Algorithms	DS1		DS2		DS3		DS4	
	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
ProClust	50.64	56.77	48.71	61.86	46.09	55.14	46.39	51.07
TribeMCL	46.09	52.89	41.42	52.14	41.04	47.48	51.22	56.46
JACOP	99.92	66.27	99.96	70.06	99.96	73.96	99.92	94.42
Pro-Kmeans	92.38	99.90	55.32	98.01	63.30	96.92	56.06	99.56
Pro-LEADER	90.21	91.40	53.15	91.24	52.96	74.06	23.34	95.70
Pro-CLARA	93.60	99.92	73.28	99.26	81.53	98.60	77.84	99.66
Pro-CLARANS	93.10	99.90	78. 62	98.70	76.24	97.34	62.06	99.09

DS4 is a very large data set which contains all sequences of DS1 (HLA protein family), DS2 (Hydrolases protein family) and DS3 (Globins protein family).

tering results are obtained when the parameter K = 28 [38]. In Pro-LEADER algorithm, the *Threshold* value is fixed to 350 after a number of simulations [28].

Experimental results of Pro-Kmeans, Pro-LEADER, Pro-CLARA, Pro-CLARANS, ProClust, TribeMCL and JACOP algorithms on DS1, DS2, DS3 and DS4 are summarized in Table 2.

In our experiments, the use of partitioning clustering methods Pro-Kmeans, Pro-LEADER, Pro-CLARA, Pro-CLARANS and JACOP have improved sensitivity and specificity of hierarchical methods, ProClust and TribeMCL.

We have demonstrated the performance of the proposed Pro-Kmeans, Pro-LEADER, Pro-CLARA, Pro-CLARANS algorithms for the clustering of protein sequences using similarity.

Experiments show also that on the considered data sets DS1, DS2, DS3 and DS4, the higher probability of correctly predicting a classifier (*Sensitivity*) is obtained using JACOP method. Pro-CLARA method gives the higher probability that the provided prediction is correct (*Specificity*) although that, Pro-Kmeans, Pro-LEADER, and, Pro-CLARANS obtain also good results. The use of Pro-LEADER method on very large and heterogeneous set, DS4, is not very valuable. In fact the number of not identified true homologues pairs (False Negatives) is very important for that, the obtained *Sensitivity* is limited to 23.34.

Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS result confirm that those proposed partitioning methods are valuable, reliable tools for the automatic functional clustering of protein sequences. The use of these methods instead of alignment methods or the classic known clustering methods by biologists can improve the clustering sensitivity and specificity and reduce significantly the computational time. The proposed methods can be used by new biologists especially to cluster a large data set of proteins into meaningful clusters in order to detect their functions.

Conclusion

Similar protein sequences probably have similar biochemical function and three dimensional structures. If two sequences from different organisms are similar, they may have a common ancestor sequence and hence they are said to be homologous. Protein sequence clustering, using Pro-Kmeans, Pro-LEADER, Pro-CLARA and Pro-CLARANS methods helps in classifying a new sequence, retrieve a set of similar sequences for a given query sequence and predict the protein structure of an unknown sequence. We noticed that the classification of large protein sequence data sets using clustering techniques instead of only alignment methods reduce extremely the execution time and improve the efficiency of this important task in molecular biology.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SF incepted and directed the research and wrote the manuscript. NE and ML participated in the coordination and the direction of the whole study. All authors read and approved the final manuscript.

Additional material

Additional File 1

Training and test data sets*. The file contains text files which correspond to the used data set in this study in fasta format. The file contains two directories: the training base which has 3500 sequences and the test base which has 1422 sequences. The considered data set, named DS4, has a total of 4922 sequences out of which 3500 sequences (practically 70% of the dataset DS4) are randomly selected for training, and 1422 for testing (practically 30% of the dataset DS4). This dataset contains proteins selected from HLA (DS1), Hydrolases (DS2) and Globins (DS3) protein families.*

Click here for file
[http://www.biomedcentral.com/content/supplementary/1756-0381-2-3-S1.zip]

References

1. Clote P, Backofen R: **Computational Molecular Biology – An Introduction.** John Wiley & Sons, Ltd; 2000.
2. Mount DW: **Bioinformatics – Sequence and Genome Analysis.** Cold Spring Harbor Laboratory Press, New York; 2002.
3. Smith TF, Waterman MS: **Identification of common molecular subsequences.** *J Mol Biol* 1981, **147:**195-197.
4. Needleman SB, Wunsch CD: **A general method applicable to the search for similarities in the amino acid sequence of the proteins.** *J Mol Biol* 1970, **48:**443-453.
5. Cabena P, *et al.*: **Discovering Data Mining: From Concept to Implementation.** Prentice Hall PTR, Upper Saddle River, NJ; 1998.
6. Fayyad UM: **Data mining and knowledge discovery: Making sense out of data.** *IEEE Expert* 1996, **11:**20-25.
7. Tatusov R, Fedorova N, Jackson J, Jacobs A, Kiryutin B, Koonin E, Krylov D, Mazumder R, Mekhedov S, Nikolskaya A, Rao B, Smirnov S, Sverdlov A, Vasudevan S, Wolf Y, Yin J, Natale D: **The COG database: an updated version includes eukaryotes.** *BMC Bioinformatics* 2003, **4:**41.
8. Kaplan N, Sasson O, Inbar U, Friedlich M, Fromer M, Fleischer H, Portugaly E, Linial N, Linial M: **ProtoNet 4.0: a hierarchical classification of one million protein sequences.** *Nucleic Acids Res* 2005:D216-8.
9. Sasson O, Linial N, Linial M: **The metric space of proteins-comparative study of clustering algorithms.** *Bioinformatics* 2002, **18(Suppl 1):**S14-21.
10. Herger A, Holm L: **Picasso: generating a covering set of protein family profiles.** *Bioinformatics* 2001, **17(3):**272-9.
11. Kriventseva E, Servant F, Apweiler R: **Improvements to CluSTr: the database of SWISS-PROT + TrEMBL protein clusters.** *Nucleic Acids Res* 2003, **31(1):**388-9.
12. Enright A, Ouzounis C: **GeneRAGE: a robust algorithm for sequence clustering and domain detection.** *Bioinformatics* 2000, **16(5):**451-7.
13. Krause A, Stoye J, Vingron M: **Large scale hierarchical clustering of protein sequences.** *BMC Bioinformatics* 2005, **6:**15.
14. Yona G, Linial N, Linial M: **ProtoMap: automatic classification of protein sequences and hierarchy of protein families.** *Nucleic Acids Res* 2000, **28(1):**49-55.
15. Shi J, Malik J: **Normalized cuts and image segmentation.** *Proceedings of the IEEE conference on Computer Vision Pattern Recognition* 1997:731-737.
16. Wu Z, Leahy R: **An optimal graph theoretic approach to data clustering: theory and its application to image segmentation.** *PAMI* 1993, **11:**1101-1113.
17. Pipenbacher P, Schliep A, Schneckener S, Schönhuth A, Schomburg D, Schrader R: **ProClust: improved clustering of protein sequences with an extended graph-based approach.** *Bioinformatics* 2002, **18(Suppl 2):**S182-91.
18. Bolten E, Schliep A, Schneckener S, Schomburg D, Schrader R: **Clustering protein sequences-structure prediction by transitive homology.** *Bioinformatics* 2001, **17(10):**935-41.
19. Enright A, Van Dongen S, Ouzounis C: **An efficient algorithm for large-scale detection of protein familes.** *Nucleic Acids Res* 2002, **30(7):**1575-84.
20. Van Dongen S: **Graph clustering by flow simulation.** In *Phd Thesis* University of Utrecht, The Netherlands; 2000.
21. Guralnik V, Karypis G: **A scalable algorithm for clustering sequential data.** *SIGKDD Workshop on Bioinformatics, BIOKDD* 2001.
22. Sperisen P, Pagni M: **JACOP: a simple and robust method for the automated classification of protein sequences with modular architecture.** *BMC Bioinformatics* 2005, **6:**216.
23. Kaufman L, Rousseeuw P: **Finding Groups in Data: An Introduction to Cluster Analysis.** John Wiley & Sons, Inc., New York; 1990.
24. Pagni M, Ioannidis V, Cerutti L, Zahn-Zabal M, Jongeneel C, Hau J, Martin O, Kuznetsov D, Falquet L: **MyHits: improvements to an interactive resource for analyzing protein sequences.** *Nucleic Acids Res* 2007:W433-37.
25. Anil KJ, Richard CD: **Algorithms for Clustering Data.** Prentice-Hall; 1988.
26. Faber V: **Clustering and the continuous k-means algorithm.** *Los Alamos Science* 1994, **22:**138-144.
27. Hartigan J, Wong M: **Algorithm AS136: A k-means clustering algorithm.** *Applied Statistics* 1979, **28:**100-108.
28. Can F: **Incremental clustering for dynamic information processing.** *ACM Trans Inf Syst* 1993, **11(2):**143-164.
29. Spath H: **Cluster analysis algorithms.** Ellis Horwood, Chichester, UK; 1980.
30. Ng R, Han J: **Efficient and Effective Clustering Methods for Spatial Data Mining.** In *Proceedings of International Conference on Very Large Data Bases* Santiago, Chile; 1994:144-155.
31. Brenner SE, Chothia C, Hubbard TJ: **Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships.** *Proc Natl Acad Sci USA* 1998, **95:**6073-6078.
32. Pearson WR, Lipman DJ: **Improved tools for biological sequence comparison.** *Proc Natl Acad Sci USA* 1988, **85:**2444-2448.
33. Altschul S, Gish W, Miller W, Myers E, Lipman D: **Basic local alignment search tool.** *J Mol Biol* 1990, **215:**403-410.
34. Essoussi N, Fayech S: **A comparison of four pair-wise sequence alignment methods.** *Bioinformation* 2007, **2:**166-168.
35. Henikoff S, Henikoff J: **Performance evaluation of amino acid substitution matrices.** *Proteins* 1993, **17:**49-61.
36. Schneckener S: **Positionsgenaues Alignment von Proteinsequenzen.** In *PhD Thesis* Universität zu köln; 1998.
37. Cathy H: **The Universal Protein Ressource (UniProt): an expanding universe of protein information.** *Nucleic Acids Res* 2006, **34:**87-191.
38. Dubes RC: **How many clusters are best?** *Pattern Recogn* 1987, **20(6):**645-663.