
Clustering algorithms adapted to protein sequence data sets

Sondes Fayech^{1,*}, Nadia Essoussi¹ and Mohamed Limam¹

¹ Department of Computer Science LARODEC, Higher Institute of Management, University of Tunis, Tunis, Tunisia

ABSTRACT

Motivation: Clustering is the unsupervised classification of a data set into groups of similar objects. This classification is based on a distance function which represents the evaluation criterion of the obtained groups. In the literature, protein sequence sets are divided into groups of homologous proteins using mostly alignment methods. These methods compare each protein sequence with all others of the data set. For that, this approach is not efficient for the classification of large protein sequence sets. In this paper, four clustering algorithms: K-means, LEADER, CLARA and CLARANS, are adapted to protein sequence data sets by introducing in these algorithms the Smith and Waterman algorithm to compare and cluster proteins and to evaluate the obtained protein groups using a new evaluation criterion. In order to evaluate and to compare these algorithms, we have tested them on a large protein sequence data set.

Results: The experimental results showed that all of these algorithms can be adapted to large biological data sets and can be used to cluster proteins into meaningful partitions. Comparison between these algorithms showed that the classification accuracies obtained by CLARA algorithm is better than those obtained by K-means, LEADER and CLARANS algorithms. The LEADER algorithm needs a short computation time to cluster protein sequence sets than K-means, CLARA and CLARANS algorithms.

1 INTRODUCTION

In bioinformatics, the number of protein sequences is now more than half a million, and it is necessary to find meaningful partitions of them in order to detect their functions. The earlier approaches of comparison 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 (Clote and Backofen, 2000; Mount, 2002). Local alignment helps in finding conserved amino acid patterns in protein sequences. Local alignment programs are based on Smith and Waterman algorithm (Smith and Waterman, 1981). In global alignment attempts are made to align the entire sequence using as many characters as possible, up to both ends of each sequence. Global alignment programs are based on Needleman and Wunsch algorithm (Needleman and Wunsch, 1970). For finding the similarity between two protein sequences, *PAM250*, *BLOSUM60*, *BLOSUM62*, scoring matrices are used (Clote and Backofen, 2000; Henikoff and Henikoff, 1992; Mount, 2002). These matrices contain the substitution costs for, all pairs of 20 amino acids. In order to calculate the alignment score, gap penalties are to be properly selected. The higher the score, the better the quality of the alignment. The alignment score is defined as

$$Score(A, B) = \sum_{i,j} S(A_i, B_j) - \sum_n g(n) \quad (1)$$

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 of n , and is defined as

$$g(n) = P_o + (n-1) * P_e \quad (2)$$

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

The pair-wise alignment of a large data set of proteins in order to cluster them into meaningful clusters is computationally expensive because of the large number of comparisons carried out. In fact, each protein of the data set must 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. Clustering is the division of data into groups of similar objects. The main objective of this unsupervised learning technique is to find a natural grouping or meaningful partition using a distance function (Cabena *et al.*, 1998; Fayyad, 1996; Keim *et al.*, 1994). The problem considered here is: Given a large set of protein sequences, design and implement efficient clustering algorithms to find meaningful partitions so as to improve the classification accuracy and reduce the computation time. These algorithms will use some known clustering techniques and will adapt them to biological data. Most clustering techniques adopt a hierarchical or a partitioning approach (Chen *et al.*, 1997). Hierarchical techniques split data in order to form a dendrogram. Hierarchical algorithms can be either divisive (top-down) or agglomerative (bottom-up). Single link and complete link are hierarchical agglomerative clustering algorithms (Anderberg, 1973). For both of them, time complexity is $O(n^2d)$ for distance computation and $O(n^3d)$ for complete clustering procedure, where d is the dimensionality (Jain *et al.*, 1999). Therefore, these algorithms are not suitable for large data sets. On the other hand, in literature, there are several hierarchical clustering methods that are used to cluster protein sequence data sets (Krause, 2005; Sasson *et al.*, 2003; Yona *et al.*, 2000) but methods based on partitional techniques are not explicitly available. The main task here is to split protein sequences into groups of similar objects and not to classify them using a dendrogram so in this paper only partitional techniques are considered.

Several partitional clustering algorithms have been proposed in the literature, including K-means (Anil and Richard, 1988; Faber, 1994; Hartigan and Wong, 1979; Kaufman and Rousseeuw, 1990), LEADER (Can, 1993; Spath, 1980), CLARA (Kaufman and Rousseeuw, 1990) and CLARANS (Ng and Han, 1994). K-means is based on K centroids of the initial partition and is iteratively improved (Anil and Richard, 1988; Faber, 1994; Hartigan and Wong, 1979; Kaufman and Rousseeuw, 1990). LEADER is an incremental algorithm in which each of the K clusters is represented by a leader (Can, 1993; Spath, 1980). K clusters are generated using a suitable *Threshold* value. In this method, the first pattern is selected as

the leader of a cluster and the remaining patterns are classified depending on the existing leaders or may become leader of a new cluster. CLARA (Clustering LARGE Applications) is a combination of a sampling approach and the PAM algorithm (Kaufman and Rousseeuw, 1990). PAM algorithm selects K patterns arbitrarily as medoids and then iteratively improves upon this selection (Kaufman and Rousseeuw, 1990). Instead of finding medoids, each of which is the most centrally located object in a cluster, for the entire data set, CLARA draws a sample from the data set and uses the PAM (Partitioning Around Medoids) algorithm to select an optimal set of medoids from the sample. To alleviate sampling bias, CLARA repeats the sampling and clustering process multiple times and, subsequently, selects the best set of medoids as the final clustering. CLARANS (Clustering Large Applications based on RANdomized Search) algorithm views the process of finding optimal medoids as searching through a certain graph, in which each node represents a set of medoids (Ng and Han, 1994). Two nodes are neighbors if their sets differ by only one object. Instead of using an exhaustive search strategy, CLARANS adopts serial randomized search. That is, starting from an arbitrary node in the graph, CLARANS randomly checks one of its neighbors. If the neighbor clustering results are better, CLARANS proceeds to this neighbor; otherwise, CLARANS randomly checks another neighbor until a better neighbor is found or a pre-determined maximal number of neighbors has been reached. To avoid being trapped in a suboptimal solution, CLARANS repeatedly starts from different initial nodes and selects the best node as the final clustering. CLARANS aims to minimize the TC_{ih} cost. The cost differential of two nodes i and h , TC_{ih} , is measured as

$$TC_{ih} = \sum_{j \in [1..n]} C_{jih} , \quad (3)$$

Where

$$C_{jih} = d(O_i, O_h) - d(O_i, O_j) , \quad (4)$$

and $d(O_i, O_j)$ is the dissimilarity between objects O_i and O_j

All of the above mentioned algorithms, K-means, LEADER, CLARA and CLARANS, aim to minimize the average dissimilarity, *Averagediss*, and it is defined as

$$Averagediss = \frac{\sum_{i \in [1..n]} d(O_i, rep(M, O_i))}{n} , \quad (5)$$

Where M is a set of selected medoids, and $rep(M, O_i)$ returns a medoid in M which is closest to O_i .

In this article, we propose to adapt the above mentioned algorithms, K-means, LEADER, CLARA and CLARANS, to biological data sets. Performance measures are usually used to judge the goodness of a clustering result. Our procedure is introduced in Section 2. A set of experiments based on a large data set was conducted and the results are illustrated in

Section 3. Finally, research contributions are presented in Section 4.

2 METHODS

This section presents the K-means, LEADER, CLARA and CLARANS clustering algorithms in new versions adapted to biological data sets.

To facilitate subsequent discussion, the main symbols used through the paper and their definitions are summarized in Table1.

Symbols	Definitions
D	Data set of protein sequences to be clustered
K	Number of clusters
n	Number of proteins in D (training base)
m	Number of proteins in testing base
O_i	a protein sequence i in D
q	Number of iterations

Table1. Summary of symbols and definitions

The goal in the target algorithms K-means, LEADER, CLARA and CLARANS is to produce K clusters from a 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:

$$f(V) = \sum_{j \in [1..n]} \text{Score}(O_j, R_i) , \quad (6)$$

Where R_i is the centroid of the group i for which belong the object O_j and $\text{Score}(O_j, R_i)$ is the alignment score of the protein sequences O_j and R_i , calculated using Equation (1).

The objective function $f(V)$ in the new versions of K-means, LEADER, CLARA and CLARANS algorithms adapted to biological data is measured based on the similarity score between protein sequences. For this reason, we have selected the Smith Waterman algorithm (Smith and Waterman, 1981) to compare proteins and we have calculated the obtained similarity score using the *BLOSUM62* scoring matrix, a gap opening as 10 and a gap extension as 2 (Essoussi and Fayeche, 2007).

2.1 Clustering algorithm adapted to biological data sets

2.1.1 K-means algorithm

K-means (Anil and Richard, 1988; Faber, 1994; Hartigan and Wong, 1979; Kaufman and Rousseeuw, 1990) proposed here,

start by a random partition of the data set D into K clusters and then use the Smith Waterman algorithm, referenced by *WatermanAlgorithm*, 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

$$SumScore(S_i, O_j) = \sum_{w \in [1..m] \neq j} Score(O_j, O_w) \quad (7)$$

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

The sequence O_j in each cluster S_i which have the maximum $SumScore(S_i, O_j)$ is considered as the centroid R_i of the cluster. The *WatermanAlgorithm* 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 . K-means 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 .

K-means algorithm adapted to biological data set is illustrated in Figure1.

2.1.2 LEADER algorithm

The LEADER algorithm (Can, 1993; Spath, 1980) is very fast, requiring only one pass through the data set D . It is thus not necessary to store the data in core, but it is sufficient to read it once from disk. LEADER algorithm for biological data sets select the first sequence of the data set D as the first leader, and use the *WatermanAlgorithm* to compute the similarity score of each sequence in D with all leaders. The algorithm detect the nearest leader R_i to each sequence O_j (which have the maximum score of similarity with the sequence) and compare the score, $Score(R_i, O_j)$, with a prefixed *Threshold*. If the similarity score of R_i and O_j , $Score(R_i, O_j)$, 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 . 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. LEADER aims also to maximize the $f(V)$ function (Equation (6)). 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.

LEADER algorithm adapted to biological data set is depicted in Figure2.

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 *WatermanAlgorithm*;
- Compute the $SumScore(S_i, O_j)$ of each protein j in S_i based on Equation (7);
- 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 *WatermanAlgorithm*;
- 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 $f(V)$ based on Equation (6);

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

Figure1. K-means algorithm for biological data sets

Input: A training set D , $D = \{O_j\}_{j=1..n}$; n is the size of D

Initialize: LeaderList = \emptyset ;

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 *WatermanAlgorithm*;
- 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 $f(V)$ based on Equation (6);

End

Output: LeaderList; LeaderList is the best partition of D into K clusters; each cluster is defined by a Leader R_i

Figure2. LEADER algorithm for biological data sets

2.1.3 CLARA algorithm

CLARA relies on the sampling approach to handle large data sets (Kaufman and Rousseeuw, 1990). Instead of finding medoids for the entire data set, CLARA algorithm for biological data draw a small sample S of $40 + 2K$ sequences from the data set D and applies the PAM algorithm, *PAMAlgorithm* (Figure3), to generate an optimal set of medoids for the sample. To alleviate sampling bias, CLARA repeats the sampling and clustering process a pre-defined number of times, q , and subsequently selects as the final clustering result the set of medoids with the maximal $f(V)$ (Equation (6)).

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} ;
 - $TS_{ih} = \sum_{j \in [1..m]} S_{jih}$ where;
 - $S_{jih} = \text{Score}(O_j, O_h) - \text{Score}(O_j, R_i); (O_j \neq R_i (i \in [1..K]))$
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 *WatermanAlgorithm*;
- 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

Figure3. PAM algorithm for biological data sets

Input parameters of CLARA 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.

CLARA algorithm adapted to biological data set is detailed in Figure4.

2.1.4 CLARANS algorithm

CLARANS (Ng and Han, 1994) algorithm for biological data set starts from an arbitrary node C in the graph, $C = [R_1, R_2, \dots, R_k]$, which represents an initial set of medoids. CLARANS randomly selects one of C neighbors, N , which differ compared to C by only one sequence. If the total score of the selected neighbour, TS'_{ih} , is more than that of the current node TS_{ih} , CLARANS proceeds to this neighbor and continues the neighbor selection and comparison process.

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 *PAMAlgorithm* (Figure3) 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 *WatermanAlgorithm*;
 - Assign O_h to the cluster with the nearest R_i ;
4. Compute $f(V)$ based on Equation (6);
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

Figure4. CLARA algorithm for biological data sets

Otherwise, 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 the maximal number of neighbors, $Maxneighbor$, is defined as (Ng and Han, 1994)

$$Maxneighbor = \text{Max}((1.25\% * K * (n - K)); 250) \quad (8)$$

CLARANS algorithm for biological data set aims to maximize the total score, TS_{ih} , differential of each pair of non-selected object O_h in D and selected object $R_i (i \in [1..K])$. The TS_{ih} is defined as

$$TS_{ih} = \sum_{j \in [1..n]} S_{jih} \quad (9)$$

Where

$$S_{jih} = \text{Score}(O_j, O_h) - \text{Score}(O_j, R_i); (O_j \neq R_i (i \in [1..K])) \quad (10)$$

CLARANS algorithm adapted to biological data use also the *WatermanAlgorithm* 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 $f(V)$ (Equation (6)).

Input parameters of 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.

CLARANS algorithm adapted to biological data set is detailed in Figure5.

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 ; ($C = [R_1, R_2, \dots, R_k]$)

2. Set $j = 1$;

3. Repeat

- Consider a random neighbor N of C ;
- Compute TS_{ih} of N and TS'_{ih} of C based on Equation (9);

• **If** $TS_{ih} > TS'_{ih}$ **then**

$C = N$;

$j = j + 1$;

Else

$j = j + 1$;

Until $j = \text{Maxneighbor}$;

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 *Waterman Algorithm*;
- Assign O_h to the cluster with the nearest R_i ;

5. Compute $f(V)$ based on Equation (6);

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

Figure5. CLARANS algorithm for biological data sets

2.2 Experimental procedure

To evaluate the K-means, LEADER, CLARA and CLARANS clustering algorithms adapted to biological data sets, we have used them on a large data set: "Training data set". We have obtained from the training phase K clusters and each cluster is defined by a medoid (centroid or leader). Then we have used the training phase results to cluster a different data set: "Test data set". The results obtained from the test phase are used to calculate the accuracy of each algorithm and to compare them. The detailed Test phase algorithm is depicted in Figure 6.

3 EXPERIMENTS AND RESULTS

Experiments on well-known data sets were performed. First, we describe the experiments and then discuss and compare

the obtained results to deduce an efficient clustering algorithm that can be used with biological data sets.

Input: A test set TS , $TS = \{O_j\}_{j=1..m}$; m is the size of TS

Initialize: Initialize the counter, TotalMatch = 0;

1. For each $j \in [1..m]$ **do**

- Compute the similarity score of each O_j with all medoids R_i ($i \in [1..K]$), using *Waterman Algorithm*;

- Find the nearest medoid R_i to O_j : NR_i ;

- PredictedClass = Class(NR_i);

- **If** PredictedClass = RealClass **then**

TotalMatch = TotalMatch + 1;

2. Accuracy = ((TotalMatch/m)*100);

End

Output: The accuracy of the used clustering algorithm adapted to biological data on the test set, TS .

Figure6. Test phase algorithm

3.1 Protein sequence data set

To evaluate the performance of the proposed clustering algorithms adapted to biological data sets, 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 grouped into 12 classes. Protein sequences of Hydrolases protein family have been collected from (<http://www.brenda.uni-koeln.de>). Hydrolases protein family sequences have been categorized into 8 classes according to their function and 3737 sequences have been considered from this family. From Globins protein family (Cathy, 2006), sequences have been collected randomly from 8 different classes and 292 sequences have been selected from the data set *IPR000971* in (<http://srs.ebi.ac.uk>). Thus, totally we have considered 28 different classes containing sequences according to protein functions. We have considered these groups of protein sequences as they have been classified according to functions by scientists/experts. The data set considered has totally 4922 sequences. From this, randomly 3500 sequences were selected for training and 1422 for testing. The experiments were done on Intel Pentium4 processor based machine, having a clock frequency of 2.4 GHZ and 512 MB of RAM.

3.2 Experimental Results

The experimental results are obtained using default values as follows. In K-means, CLARA and CLARANS algorithms, the number of iterations q is fixed to 5 (Ng and Han, 1994) and the number of clusters K is fixed to 28. After a number of simulations (80 simulations) we have deduce that the best clustering results are obtained if the parameter $K = 28$ (Dubes, 1987). The only needed parameter in LEADER algorithm is the *Threshold* score. After a lot of simulations (50 simulations), the *Threshold* value is fixed to 350 (Can, 1993).

The evaluation criteria of the proposed algorithms adapted to biological data in this study are the clustering quality

(accuracy), the training time and the testing time (Zait and Messatfa, 1997). Experimental results of these algorithms are summarized in Table2.

Algorithm	Classification Accuracy (%)	Training Time (mn)	Test Time (mn)
CLARA	71.0	10095	35.42
CLARANS	69.0	14500	34.85
K-means	62.0	4205.16	35.7
LEADER	45.0	297.65	36.21

Table2. Clustering accuracy and computational time results

In our study, experimental results of the new versions of K-means, LEADER, CLARA and CLARANS adapted to biological data and tested on a large protein sequence data set show that concerning the accuracy criterion, CLARA is more performing than all others algorithms with an accuracy of 71%, and that CLARANS and K-means algorithms also can be used to cluster large biological data set with an acceptable clustering accuracy of 69% and 62% respectively. In our experiments, LEADER algorithm is the less perform with an accuracy of about 45%. On the other hand, concerning the computational time criterion, LEADER is more faster compared to all others algorithms in training phase and CLARANS is the slowly one. On the other hand, K-means algorithm is more efficient compared to CLARA algorithm which needs more time to carry out the training phase. In the test phase, the four algorithms have practically the same execution time (Table2).

4 CONCLUSION

In this paper, we have proposed new versions of the clustering algorithms K-means, LEADER, CLARA and CLARANS adapted to biological data sets. For this reason, we have used the Smith and Waterman alignment method in these algorithms in order to compare sequences and to compute their similarity scores. We have changed also the evaluation criterion of the obtained clusters to adapt it to biological data. In fact, similar protein sequences probably have similar biochemical function and three dimensional structure. If two sequences from different organisms are similar, they may have a common ancestor sequence and sequences are said to be homologous. Protein sequence clustering, using the proposed clustering algorithms adapted to biological data sets, helps in classifying a new sequence, retrieve a set of similar sequences for a given query sequence and predicting the protein structure of an unknown sequence. We can deduce also from this study that the classification of large protein sequence data sets using the clustering techniques instead of only alignment methods will extremely reduce the execution time and will improve the efficiency of this important task in molecular biology. We further aim to compare the classification accuracy and computation time of

the proposed algorithms, K-means, LEADER, CLARA and CLARANS adapted to biological data sets, with few more algorithms which will be also adapted to biological data sets.

REFERENCES

- Anderberg, M.R. (1973) Cluster Analysis for Applications. Academic Press, Inc., New York, NY.
- Anil, K. J. and Richard, C. D. (1988) Algorithms for Clustering Data. Prentice-Hall.
- Cabena P. *et al.*, (1998) Discovering Data Mining: From Concept to Implementation. Prentice Hall PTR, Upper Saddle River, NJ.
- Can, F. (1993) Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.*, 11, 2, 143–164.
- Cathy, H. (2006) The Universal Protein Ressource (UniProt): an expanding universe of protein information. *Nucleic Acids Res.*, 34, 87-191.
- Chen, M. S. *et al.*, (1997) Data Mining: An Overview from a Database Perspective. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, 6.
- Clote, P. and Backofen, R. (2000) Computational Molecular Biology - An Introduction. John Wiley & Sons, Ltd.
- Dubes, R. C. (1987) How many clusters are best? - an experiment. *Pattern Recogn.* 20, 6, 645–663.
- Essoussi, N. and Fayech, S. (2007) A comparison of four pair-wise sequence alignment methods. *Bioinformatics*, 2, 166-168.
- Faber, V. (1994) Clustering and the continuous k-means algorithm. *Los Alamos Science*, 22, 138–144.
- Fayyad, U. M. (1996) Data mining and knowledge discovery: Making sense out of data. *IEEE Expert*, 11, 20–25.
- Hartigan, J. and Wong, M. (1979) Algorithm AS136: A k-means clustering algorithm. *Applied Statistics*, 28, 100-108.
- Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Science, USA.*, 89(10), 915-919.
- Jain, A.K. *et al.* (1999) Data clustering: A Review. *ACM Computing Surveys*. 31 (3), pp. 264-323.
- Kaufman, L. and Rousseeuw, P. J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, Inc., New York.
- Keim, D. *et al.* (1994) Supporting Data Mining of Large Databases by Visual Feedback Queries. *Proceedings of the 10th Data Engineering Conference*, pp 302-313.
- Krause, A. (2005) Large scale hierarchical clustering of protein sequences. *BMC bioinformatics*, 6(15), 1-12.
- Mount, D.W. (2002) Bioinformatics - Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, New York.
- Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of the proteins. *J. Mol.* 48, 443-453.
- Ng, R. and Han, J. (1994) Efficient and Effective Clustering Methods for Spatial Data Mining. *Proceedings of International Conference on Very Large Data Bases*, Santiago, Chile, pp144-155.
- Sasson, O. *et al.*, (2003) ProtoNet: hierarchical classification of the protein space. *Nucleic Acids Research*, 31, 348-352.

- Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, 195-197.
- Spath, H. (1980) Cluster analysis algorithms. Ellis Horwood, Chichester, UK.
- Yona, G. *et al.*, (2000) ProtoMap: automatic classification of protein sequences and hierarchy of protein families. *Nucleic Acids Research*, 28, 49-55.
- Zait, M. and Messatfa, H. (1997) A Comparative Study of Clustering Methods. *Future Generation Computer System*, Vol. 13, pp.149-159.

Additional files provided with this submission:

Additional file 1: supplementary materials.zip, 1733K

<http://www.biodatamining.org/imedia/4977494752360028/supp1.zip>