An Unsupervised Protein Sequences Clustering Algorithm Using Functional Domain Information

Wei-Bang Chen, Chengcui Zhang, and Hua Zhong
Department of Computer and Information Sciences,
University of Alabama at Birmingham, Birmingham, Alabama 35294, USA
{wbc0522, zhang, hzhong}@cis.uab.edu

Abstract

In this paper, we present an unsupervised novel approach for protein sequences clustering by incorporating the functional domain information into the clustering process. In the proposed framework, the domain boundaries predicated by ProDom database are used to provide a better measurement in calculating the sequence similarity. In addition, we use an unsupervised clustering algorithm as the kernel that includes a hierarchical clustering in the first phase to pre-cluster the protein sequences, and a partitioning clustering in the second phase to refine the clustering results. More specifically, we perform the agglomerative hierarchical clustering on protein sequences in the first phase to obtain the initial clustering results for the subsequent partitioning clustering, and then, a profile Hidden Markove Model (HMM) is built for each cluster to represent the centroid of a cluster. In the second phase, the HMMs based k-means clustering is then performed to refine the cluster results as protein families. The experimental results show our model is effective and efficient in clustering protein families.

Keywords: Protein Sequences Clustering, Data Mining and Knowledge Discovery, Profile Hidden Markov Model (HMM), ProDom database.

1. Introduction

The functions and structures of protein sequences have great importance in biomedical research. Researchers have to spend a lot of time and resources on some routine experiments in order to understand the function and structure of various unknown protein molecules. In molecular biology research, homologous protein sequences with similar functions and structures [1, 2] often belong to the same protein family. On the other hand, protein sequences from the same protein family almost always have similar functions and structures. Thus, in predicting the most possible structures and functions of a previously unseen sequence, we can first check the protein family to which it most likely belongs and then use the common functions and structures from that family as its estimates. This process can be automated and the workload can be significantly reduced if we have a way to automatically predict the family membership for any unseen sequence. For this purpose, we can first cluster similar sequences into groups based on their sequence homology and then built a representative model for each group, which serves as the classifier for that group.

A protein domain is a part of a protein sequence and a structure that can evolve, function and exist independent of the rest of the polypeptide chain. One domain may appear in various proteins that are evolutionarily related. Molecular evolution gives rise to families of related proteins with similar sequences and structures. Domains are believed to be crucial in characterizing biological properties of proteins [3]. Therefore, it makes sense to increase the weights of domain areas when performing the pair-wise alignment of two protein sequences, and it should bring us closer to the desired clustering result.

The ProDom database (http://prodom.prabi.fr) we used in our experiment is a comprehensive database of protein domain families [4]. It uses all available protein sequences to perform Multiple Sequences Alignment to generate and store all the domain families in the database. BLAST-P [5] is introduced to align the sequences against its database to get the high-scoring domains. When 3D structures are available for target domains, the output is directly linked to both SWISS-MODEL and Geno3D servers for homology-based domain modeling. The BLAST searches suggest a possible domain arrangement for the query protein.

In this paper, we proposed a two-phase hybrid protein sequence clustering method which takes advantage of both the hierarchical and the partition clustering methods. The hierarchical clustering algorithm works by grouping data

978-1-4244-2660-7/08/$25.00 ©2008 IEEE

objects into a tree of different clusters based on the distance between data objects. Depending on the bottom-up or top-down fashion by which the hierarchical decomposition is formed, hierarchical clustering methods can be divided into two categories: the agglomerative and divisive hierarchical clustering methods. In hierarchical clustering, cutting the tree at a certain height can give a clustering at a selected precision. The user can specify a desirable number of clusters to perform the merging process. The major problem with hierarchical clustering is that it cannot go back and undo a previous clustering step. Therefore, a bad choice of merging or splitting made in some previous step will probably lead to a low-quality clustering result. The time-complexity of a naïve implementation of the algorithm is $O(n^3)$ [6].

The partitioning method is another widely used clustering method. *k*-means and *k*-medoids are two representatives of this method. The *k*-means method requires a parameter *k*, which indicates the number of clusters, and uses seeds to represent the centroids of clusters. An iterative relocation is often used in this kind of method to move data objects from one cluster to another according to some similarity measure until the there is no further move and finally it forms *k* clusters. In the iterative process, clusters are refined by maximizing the similarity within a cluster and minimizing the similarity between different clusters, and the centroids of clusters are updated accordingly at the end of each iteration.

There are usually two ways to generate the cluster centroids: the similarity-based methods and the model-based methods. The similarity-based methods generate the centroid of a cluster by examining the similarity score between each pair of data points in that cluster. The model-based methods generate a pseudo centroid for a cluster by building a model to represent that cluster [6]. With the model-based methods, the resulting model for a cluster directly characterizes that cluster rather than computing the similarity score, therefore, the model-based methods often provide better interpretability than do the similarity methods. In the second phase of our proposed clustering method, we use a probability modeling technique, the profile Hidden Markov Model [9] to represent each cluster. The Hidden Markov Model is widely used in time series data analysis, such as speech recognition and sequence analysis. In our case, we consider protein sequences a special type of time series data.

Our goal in this study is to develop an efficient and robust framework for protein sequence clustering. The first stage is the domain prediction stage in which it dynamically acquires the sequence domain boundaries by searching the ProDom database to find the best matching domain families in sequences. In the clustering stage, it performs a two-phase hybrid clustering which combines the strength of both the hierarchical agglomerative clustering and the *k*-means partition clustering to build a profile HMM for each cluster. As domain areas are considered especially important in sequence clustering, they will be given higher weights at each clustering step for more accurate clustering result.

Our experimental results demonstrate that the weight increment in functional domains significantly improves the accuracy of the clustering results in either the hierarchical clustering step or the partition clustering step or both.

In the rest of this paper, we describe the proposed framework in detail in Section 2. Section 3 demonstrates the experimental results, and Section 4 concludes this paper.

2. The proposed method

The proposed framework consists of two main stages - the domain prediction stage and sequence clustering stage. In the domain prediction stage, we utilize the ProDom database to predict domain boundaries of the protein sequences used in our experiments. In the clustering stage, we adopt a hybrid clustering algorithm to group protein sequences. We also increase the weight for the functional domains to improve the accuracy of clustering results. The high-level architecture of the proposed framework is illustrated in Figure 1.

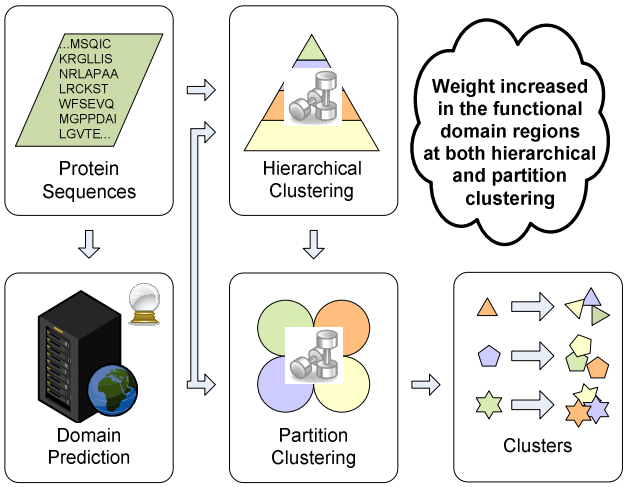


Figure 1. The workflow of the proposed framework

2.1. Domain prediction

As illustrated in Figure 1, in the proposed framework, the first stage is to discover the functional domains of a

given protein from its primary sequence. To predict the functional domains of protein sequences, we use the ProDom database as our domain predictor. The domain boundary prediction is achieved by using the BLAST-P to search a given sequence against the consensus sequence provided with the ProDom families and the multiple alignments provided with each ProDom family [7].

In general, this searching process results in more than one domain returned by the ProDom database. Therefore, we select the top three high-scoring hits in the prediction results as the estimated functional domain boundaries. Since the protein sequences used in HMM profiling need to be in multiple-aligned format which may contain gaps, and the query sequences are in FASTA format that have no gaps, it is essential to map the retrieved positions of these domain boundaries from the FASTA format to the multiple-aligned format.

In order to obtain the mapped domain boundary positions, we do the transformation as follows - If p is the position of a domain boundary in FASTA format, we locate the position in multiple-aligned format by using the procedure described below.

First, we initialize a counter C and enumerate the number of characters from the start position of a multiple-aligned sequence. Then, if the retrieved character is a gap, we skip it and move to the next position. If the retrieved character is an amino acid, we increase the value of the counter C by one and move to the next position. We continue this process until the value of counter C is equal to p. At this point, the current position p' in the multiple-aligned sequence is the corresponding transformed position for p in the FASTA format of that sequence. In this way, we map the positions of the retrieved functional domain boundaries to their correspondents in the multi-aligned format, and store the converted positions for calculating the similarity in the subsequent clustering processes. Equation 1 is the formula used to transform the boundary position between the FASTA and the multiple-aligned formats:

$$p' = p + \text{number\_of\_gaps before } p' \qquad (1)$$

where p' is the transformed boundary position in the multi-aligned format. In this way, we transform the positions of the retrieved functional domain boundaries, and store the converted positions for calculating the similarity in the subsequent clustering processes.

2.2. Hierarchical clustering

The hierarchical clustering step plays an important role as providing some pseudo-supervised information in this unsupervised protein sequences clustering algorithm. More specifically, we adopt the agglomerative hierarchical clustering algorithm as the first step in this stage to provide the initial clusters and guide the subsequent k-means clustering.

In the agglomerative clustering step, initially each protein sequence is considered as an independent data object which forms a cluster of its own. The agglomerative hierarchical clustering algorithm progressively merges two clusters with the highest similarity score. Gradually, the fusions cause the remaining clusters grow larger and larger. This merging process will not stop until the desired number of clusters is formed, and its entire progress can be represented by a dendrogram which illustrates how two clusters are merged at each successive stage of analysis.

Essentially, in each step, the merge of two clusters totally depends on the measurement of the pair-wised similarity score. Hence, the similarity measure is the key factor in the hierarchical clustering. There are several approaches to determine the similarity between two clusters, including single-link, complete-link, and average-link approaches [6]. In the proposed hybrid algorithm, we adopt the single-link approach to measure the similarity score between two clusters. The similarity score of two clusters is then determined by the similarity score of the two most similar sequences from the two sequences.

In addition to the various approaches used in the similarity score measurement, determining where to increase the weight is another important issue since the locations of functional domains in each sequence might be different. This matters especially when computing the similarity score of two sequences from different protein families. To deal with this issue, we increase the weight of those segments on the sequences that are located within any union region of the functional domains of the two sequences. We exemplified the functional domain union regions in Figure 2.

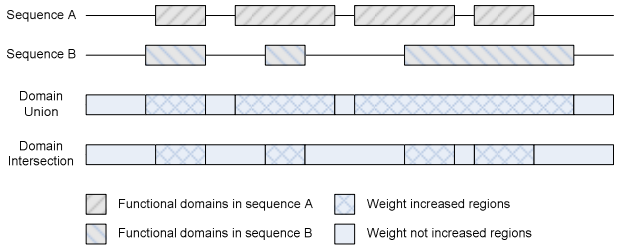


Figure 2. Weight increased regions

The similarity measure between any two sequences can be computed by Equation 2.

$$SM_{ij} = \sum_{p=1}^{n} w_p \times sm_p + w \times \rho$$
$$\begin{cases} w_p = 2, p \in \text{weight increased regions} \\ w_p = 1, p \in \text{weight not increased regions} \end{cases} \quad (2)$$

where SM_{ij} is the similarity score between two sequences i and j; sm_p is the substitution matrix (BLOSUM62) score (see Table 1) of the p^{th} character pair in the aligned sequences; ρ is the space penalty times the number of spaces that are inserted during the process of multiple alignment; w_p is the weight applied at the position p. In this study, we tried different values for w_p and found that an increased weight of 2 (compared with the weight 1 for non-domain areas) has produced the best result. Hence, a high similarity score suggests a good match between the two sequences.

Table 1. A part of BLOSUM62 Substitution Matrix

	C	S	T	P	…
C	**9**	-1	-1	-3	…
S	-1	**4**	1	-1	…
T	-1	1	**4**	1	…
P	-3	-1	1	**7**	…
…	…	…	…	…	…

After iteratively computing the similarity score in a pair-wised manner, we may merge two clusters as one in each cycle. This iterative process will continue until the desired number of clusters is obtained, that is, the entire dataset is clustered into k clusters. The clustering results are then passed to the partitioning clustering for further refinement.

2.3. Partitioning clustering

In Phase II, we use the pre-clustering result from the hierarchical clustering step as the initial partition of sequences. We then use a profile Hidden Markov Model to represent the centroids of clusters to do the k-means partitioning clustering. The model building and the sequences membership update are iteratively refined until there is no further change in the clusters [6, 8-10].

The domain areas of a profile HMM are decided by its member sequences. We select the union of all the sequences domain areas as the model's domain areas. We dynamically update the domain areas of a profile HMM after each iteration of clustering.

To explain the clustering process, say we want to cluster a dataset D into k clusters. D contains d sequences of length n. S_i denotes the i^{th} sequence in the dataset. R_k denotes the set of sequences in cluster k. M_k denotes the profile Hidden Markov Model for the cluster k. P_{ik} (S_i | M_k) denotes the probability of S_i under model M_k.

In a profile HMM M_k, m_j denotes the j^{th} state in the HMM profile. C_{ij} denotes the j^{th} character in i^{th} sequence. p_{ij} denotes the probability of C_{ij} in the m_j state.

- **The HMMs based k-means clustering algorithm**

Initialization:
According to the pre-clustering result, each sequence S_i is assigned to a cluster k.

Repeat:

1. Build a profile Hidden Markov Model M_k to represent the centroid of cluster k with the probability of each observed alphabet in R_k ($|R_k| = n$).

$$M_k = (m_1, m_2, ..., m_n) \quad (3)$$

2. Compute the probability P_{ik} (S_i | M_k) for cluster k by the following equation:

$$P_{ik} = \sum_{j=1}^{n} w_p p_{ij}$$
$$\begin{cases} w_p = 2, p \in \text{domain regions} \\ w_p = 1, p \in \text{non-domain regions} \end{cases} \quad (4)$$

3. Evaluate sequence S_i based on its P_{ik} values. A high score implies that sequence S_i is likely a member of cluster k. Find the maximum value of P_{ik} among all ks via the above equation. Reassign sequence S_i to its best-matched cluster to refine the clustering result.

Until:
No further changes in each cluster.

The HMM M_k consists of a connected set of states; m_j consists of the set of probabilities of alphabets at position j. The probability set is obtained by the following steps:

1. Count the occurrence of each alphabet in σ at the position j over all sequences in R_k.
2. In order to avoid the zero probabilities, we add one to each count of alphabet as a pseudo count method. The denominator of the probability is the sum of the number of sequences in that cluster and the number of alphabets in σ [11].
3. Compute the probabilities for each alphabet in σ at the position j.

To avoid the empty cluster problem, that is, an empty cluster forms during the clustering process, the system will check the clustering result at the end of each iteration to find those empty clusters. Once an empty cluster is found, we will move a sequence which has the lowest probability of belonging to its origin cluster to the empty cluster.

3. Experimental results

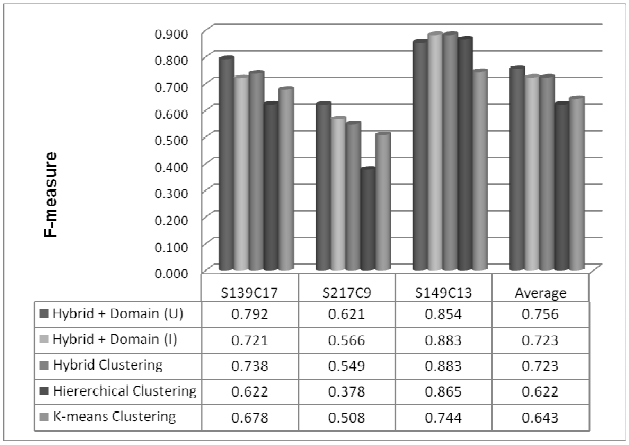
The dataset we used in our clustering experiment is obtained from the Cytochrome P450 Homepage (http://drnelson.utmem.edu/CytochromeP450.html), which is maintained at the University of Tennessee Health Science Center. 429 protein sequences from 65 protein families are selected.

The performance of our clustering method is evaluated by the F-measure [12]. The higher the F-measure value the better the clustering result is. We randomly select three sequence subsets to perform the clustering. Subset 1 (S135C17) has 135 protein sequences in 17 families, Subset 2 (S149C13) has 149 sequences in 13 families and Subset 3 (S217C9) has 217 sequences in 9 families.

In our experiment, the hybrid method with domain prediction is compared with three other algorithms: the two-phase hybrid method without domain information, the hierarchical clustering method, and the standard *k*-means partitioning method. In our experiments, we also tested two approaches of increasing the domain weight for computing the similarity score between a sequence and another sequence/model. The two different methods include: increasing the weight for the union of domain areas between a sequence and another sequence/model ("Hybrid+Domain (U)" in Figure 3), and increasing the weight for the intersection of domain areas between them ("Hybrid+Domain (I)" in Figure 3). Figure 3 demonstrates the performance of the five different clustering algorithms evaluated in our experiments. The average F-measure of "Hybrid+Domain (U)" with domain prediction is 0.756 which is the highest among all the five methods. The average F-measure of "Hybrid+Domain (I)" is 0.723 which is almost the same as that of the hybrid clustering method without domain prediction. This is probably because "Hybrid+Domain (U)" not only increases the weight of common domain areas between sequences (or sequence-model), it also increases the penalty caused by the mismatch of domain areas between sequences (or sequence-model). Different function domains always have significant differences in their primary sequence alignments. Proteins belonging to different families always have different function domains in their primary sequences, while proteins within the same family always have similar function domains. Increasing the weight for the union of two sequences' domain areas can thus magnify the similarity between two sequences in the same family and in the meanwhile magnify the dissimilarity between sequences from different families in clustering.

As shown in Figure 3, the proposed method with hybrid clustering and domain prediction is also significantly better than the hierarchical agglomerative clustering method (0.622) and the standard *k*-means partitioning method (0.643).

In brief, our proposed two-phase hybrid clustering method using domain prediction produces the best result among all the methods we used in the comparison, which also demonstrates the discriminative power of protein domains in the protein clustering and classification.



	S139C17	S217C9	S149C13	Average
Hybrid + Domain (U)	0.792	0.621	0.854	0.756
Hybrid + Domain (I)	0.721	0.566	0.883	0.723
Hybrid Clustering	0.738	0.549	0.883	0.723
Hiererchical Clustering	0.622	0.378	0.865	0.622
K-means Clustering	0.678	0.508	0.744	0.643

Figure 3. Algorithm comparison results

4. Conclusions and future work

In this paper, we propose an unsupervised hybrid clustering algorithm using the functional domain information. In the proposed hybrid clustering algorithm, the hierarchical clustering algorithm pre-clusters the given sequences as the initial partition for the *k*-means clustering method based on profile HMMs. Our experimental results demonstrate the robustness and effectiveness of the proposed framework in clustering protein sequences.

In this study, we conduct several experiments to explore the proper strategy for increasing the weight within functional domains. The experimental results show that increasing the weight in the functional domains can improve the accuracy of the clustering results. At its current stage, our algorithm depends on manually querying the online database for predicting the locations of functional domains. However, not every sequence has its domain information stored in the database. In order to further apply our algorithm to unknown sequences, we need to go one step further with the structural prediction

for functional domains. In principal, the primary sequence can be used to predict secondary structures and even the higher level of 3D structure. The exploration in predicting protein structures from their primary sequences will provide the ability to cluster unknown sequences.

In addition to domain prediction, our proposed framework has good scalability since we can store the profile HMMs in database. These profile HMMs represent the model of the corresponding protein family. Hence, we can use them as classifiers to predict the membership of any unknown sequence. Similarly, we can first use domain prediction algorithm to predict the domain areas of an unknown sequence and then increase the weight of these areas in the classification process. Since the length of unknown sequences might be different from the model, we can apply a dynamic programming step before we match unknown sequences against the model. The dynamic programming will find the optimal alignment of an unknown sequence against the model. With this process, we can group unknown protein sequences into protein families represented by profile HMMs, which can be further used to predict their biological functions.

A more systematic way of increasing the weight for domain areas is another important task in our future work. Different function domains may have different lengths. The normalization of the weight increment on functional domains with respect to their lengths is critical in eliminating the bias that is caused by the domain length difference.

5. Acknowledgement

This research of Dr. Chengcui Zhang is supported in part by NSF DBI-0649894 and UAB ADVANCE Faculty Research Award through NSF.

6. References

[1] A. J. Enright and C. A. Ouzounis, "GeneRAGE: a robust algorithm for sequence clustering and domain detection", *Bioinformatics*, vol. 16, pp. 451-7, 2000.

[2] E. V. Kriventseva, M. Biswas, and R. Apweiler, "Clustering and analysis of protein families", *Curr Opin Struct Biol*, vol. 11, pp. 334-9, 2001.

[3] C. J. Ku and G. Yona, "Domain-based protein hierarchy and detection of semantically significant domain architectures", presented at 13th Annual International Conference on Intelligent Systems for Molecular Biology, Detroit, MI, USA, June 2005.

[4] C. Bru, E. Courcelle, S. Carrere, Y. Beausse, S. Dalmar, and D. Kahn, "The ProDom database of protein domain families: more emphasis on 3D", *Nucleic Acids Res*, vol. 33, pp. D212-5, 2005.

[5] 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", *Nucleic Acids Res*, vol. 25, pp. 3389-402, 1997.

[6] J. Han and M. Kamber, *Data mining: concepts and techniques*, San Francisco, CA, USA: Morgan Kaufmann Publishers, 2001.

[7] "The ProDom Help File. (http://prodom.prabi.fr/)."

[8] S. R. Eddy, "Hidden Markov models", *Curr Opin Struct Biol*, vol. 6, pp. 361-5, 1996.

[9] S. R. Eddy, "Profile hidden Markov models", *Bioinformatics*, vol. 14, pp. 755-63, 1998.

[10] I. S. Main and I. Dubchak, "Representing and reasoning about protein families using generative and discriminative methods", *Journal of computational biology : a journal of computational molecular cell biology*", vol. 7, pp. 849-62, 2000.

[11] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*, Cambridge, UK: Cambridge University Press, 2000.

[12] J. Seo, M. Bakay, Y. W. Chen, S. Hilmer, B. Shneiderman, and E. P. Hoffman, "Interactively optimizing signal-to-noise ratios in expression profiling: project-specific algorithm selection and detection p-value weighting in Affymetrix microarrays", *Bioinformatics*, vol. 20, pp. 2534-44, 2004.