A Robust Method for Biological Sequence Clustering

Wei-Bang Chen, Chengcui Zhang
Computer and Information Sciences Department,
University of Alabama at Birmingham, Birmingham, Alabama 35294, USA
{wbc0522, czhang02}@uab.edu

Abstract

In this paper, we proposed a two-phase hybrid method for biological sequence clustering, which combines the strengths of the hierarchical agglomerative clustering methods and the partition clustering methods. In Phase I, the hybrid method uses the hierarchical agglomerative clustering algorithm to pre-cluster the aligned sequences, while in the second phase it takes the pre-clustering result as the initial partition for the profile Hidden Markov Models (HMMs) based k-means partition clustering method. Such initial partitions (generated from Phase I), as against random initial partitions, are usually more reasonable and thus can avoid the inconsistency problem in the partition clustering methods due to the randomness in initial partitions. In addition, the inaccuracy of the hierarchical agglomerative clustering methods can be compensated by the profile HMM based k-means partition clustering since the latter is model-based and can better describe the dynamic properties of the data in a cluster. Experiments on a molecular sequence dataset demonstrate the effectiveness and the efficiency of the proposed hybrid clustering algorithm.

1. Introduction

Drug design and disease treatment require the knowledge of the structures and functions of biological sequences. Researchers have to perform a great deal of time-consuming and expensive routine experiments to understand the structures and functions of unknown molecules. If we can narrow down the scope by predicting the most possible structures and functions for unknown molecules, the amount of experiments can be drastically reduced. To predict the structure and function of a sequence, we can group related sequences into categories on the basis of sequence similarity since sequence homology may indicate the common structure and function [1, 2]. This grouping process is called clustering. The major clustering methods can be classified into five categories: hierarchical methods, partitioning methods, density-based methods, grid-based methods, and model-based methods. The most commonly used clustering methods are hierarchical methods and partitioning methods [2-5].

The hierarchical methods are a similarity (or distance) based bottom-up or top-down clustering technique. These methods will finally generate a hierarchical-tree. Each object initially forms a cluster of its own as a singleton and then pairs of clusters are merged iteratively until a certain stopping criterion is met. The advantages of the hierarchical methods are that they can be used to cluster a wide variety of datasets and generate a hierarchical tree, which illustrates explicitly the distance between any two objects and how they are merged. However, there are still problems in these methods: (1) they cannot perform adjustment once a merge or split has been done; (2) they must rebuild the entire tree from the beginning when new sequences are added; (3) the time complexity of naïve implementations of the algorithms is $O(n^3)$ [3, 6].

Another commonly used clustering methods discussed in this paper are partitioning methods, which is known as *k*-way clustering. Among these partitioning methods, *k*-means and *k*-medoids are the most well-known and commonly used partitioning methods. This type of methods requires a parameter *k* which indicates the number of clusters and use seeds to represent the centroids of clusters. This algorithm uses an iterative relocation strategy which moves the sequences in one cluster to another according to some similarity measure till there is no further move, with the purpose to maximize the similarity within a cluster and minimize the similarity between clusters, and finally forms *k* clusters. In this iterative process, both the similarity-based methods and model-based methods are used to generate the new seeds. The similarity-based methods generate a new seed by computing the similarity scores between all objects in the cluster while the model-based methods generate a new seed by building a model to represent the cluster [3]. The model-based methods provide better interpretability than do the similarity-based methods since the resulting model for each cluster directly characterizes that cluster. Among various model-based methods, the

0-7803-9788-6/06/$20.00 ©2006 IEEE.

Profile Hidden Markov Model (Profile HMM) has been widely used as a probabilistic modeling technique to describe the dynamic properties of ordered observations, such as speech recognition and sequence analysis [7-10]. The main advantage of partition methods is the ability to build models for clusters which makes it more valuable especially for frequently updated dataset. Partition clustering algorithms are also faster than agglomerative clustering algorithms because its time complexity is $O(n^2)$ [3]. However, since the initial seeds are selected randomly, we might obtain different results while running the partition methods on the same dataset from time to time. Another problem is the empty cluster problem. Once an empty cluster forms, no sequence can be assigned to that empty cluster, because all sequences would have the same similarity to that cluster, which is always the smallest compared to their similarity to the other clusters. Besides, the partitioning methods also require specifying the number of clusters k in advance.

The goal of our study in this paper is to develop a robust yet efficient model for biological sequence classification. Here, we proposed a hybrid clustering algorithm which combines the strengths of the hierarchical method and the partition methods. We use the hierarchical agglomerative clustering algorithm in Phase I to pre-cluster input sequences, and in Phase II the pre-clustering result is used as the initial partition for the profile HMM based k-means partition clustering algorithm. We slightly modified the standard k-means partition clustering algorithm to avoid the empty cluster problem. We also avoid the problem of random initial partition in partition methods by pre-clustering sequence data using the hierarchical agglomerative clustering algorithms. Our experimental results demonstrated that the inaccuracy of the hierarchical agglomerative clustering can be compensated by the profile HMM based k-means partition clustering since the model-based partition clustering can better describe the dynamic properties of the data in a cluster. On the other hand, the supervised initial partition generated in Phase I can help the clustering process to escape from getting stuck at a local minimum.

Section 2 discusses the details of the proposed hybrid clustering algorithm. Section 3 presents the experimental results. Section 4 concludes this paper.

2. The proposed hybrid clustering algorithm

The hybrid clustering algorithm we proposed combines the strengths of both hierarchical agglomerative clustering algorithms and partition clustering algorithms. Figure 1 illustrates the concept of the proposed hybrid algorithm which includes both algorithms in two phases.

2.1. Hierarchical agglomerative clustering

In phase I, the hierarchical agglomerative clustering method initially places each sequence in the dataset into a cluster of its own as a singleton and then merges the most similar two clusters into one cluster iteratively till the termination condition is satisfied [3, 6].

In the merging process, we need to construct a similarity matrix to measure and to store the similarity between two clusters. Based on the single-link approach, the similarity between two clusters is measured by the similarity of the most similar pair of sequences belonging to these two clusters. The similarity S between two sequences can be measured by the following equation [11, 12]:

$$S = \Sigma\, m_i + n \times p \qquad (1)$$

where, S is the similarity score between two sequences; m_i is the score of the i^{th} character pair in the aligned sequences obtained from the substitution matrix such as BLOSUM62; n is the number of spaces which are inserted during the process of string alignment; p is the space penalty. A high similarity score suggests a good match between two sequences.

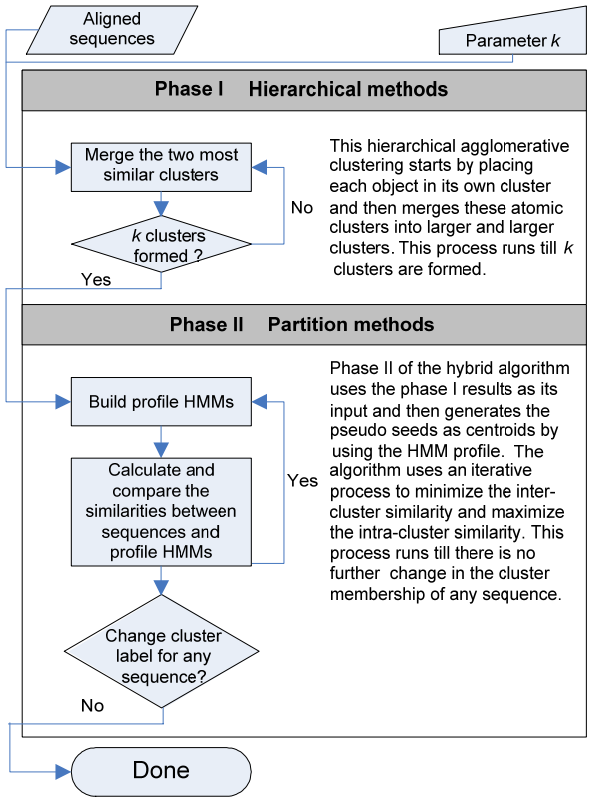

Figure 1. The flowchart of the hybrid clustering algorithm

We illustrate how to calculate the similarity scores between two clusters and between two sequences in

Figure 2. The similarity between two clusters C_1 and C_2 is measured by the similarity of the most similar pair of sequences belonging to these two clusters. Figure 2a shows that S_1 in C_1 and S_4 in C_2 have the highest similarity score 61. Thus, the similarity score of the two clusters is 61.

Figure 2b shows the similarity score of two sequences. The similarity of two aligned sequences is computed as the sum of the substitution matrix score of the i^{th} character pair plus the number of spaces × space penalty [6, 7, 13]. A typical value of the space penalty is -9 and thus the similarity score of S_1 and S_4 is 61.

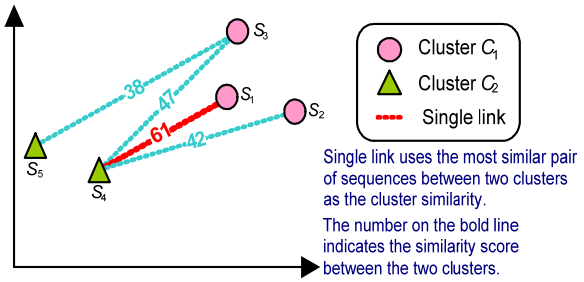

Figure 2a. The similarity score of two clusters

S_1	H	C	H	D	D	H	N	B	C	E	W	M	H	C	N	W	Similarity
S_4	H	C	H	D	A	-	-	D	C	E	-	M	H	C	N	W	
Score	8	9	8	6	-2	-9	-9	6	9	5	-9	5	8	9	6	11	61

Figure 2b. The similarity score of two sequences

Typically, a substitution matrix scores for each possible match of the alphabets in sequence character set. The most widely used matrices for proteins are categorized into two families, the PAM family and the BLOSUM family. The PAM matrices are derived by extrapolation from mutation rates while the BLOSUM matrices are derived by calculation from the most highly conserved regions in proteins which usually represents the function domains of proteins. This suggests that BLOSUM matrices are more appropriate for searches and alignments than PAM matrices [7]. As shown in Table 1, we choose BLOSUM62 (*http://www.people.virginia.edu/~wrp/cshl05/blosum62.html*), the matrix calculated from comparisons of sequences with no less than sixty-two percent divergence, as the substitution matrix to compute the similarity of sequences.

In Phase I, the termination condition of the algorithm is "till the desired number of clusters is reached", meaning the entire dataset will be clustered into *k* clusters by applying the hierarchical agglomerative clustering algorithm. We then pass the clustering result to Phase II as its input (the initial partition).

2.2. Partition clustering by k-means based on HMM profile

In Phase II, instead of using random initial assignments of sequences, the partition clustering algorithm takes the pre-clustering result generated by Phase I as the initial partition of sequences. A probability model, the profile Hidden Markov Model, is then built based on the clustering result of Phase I. The partition clustering algorithm will take this probability model to model each cluster and reassign each sequence to a

Table 1. BLOSUM62 Substitution Matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	**4**	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	**4**	1	-1	1	0	1	0	0	0	-1	0	-1	-2	-2	-2	-2	-2	-3
P	-3	-1	1	**7**	-1	-2	-1	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	-1	-1	**4**	0	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-2	-3
G	-3	0	1	-2	0	**6**	-2	-1	-2	-2	-2	-2	-2	-3	-4	-4	0	-3	-3	-2
N	-3	1	0	-2	-2	0	**6**	1	0	0	-1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	1	-1	-2	-1	1	**6**	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	0	-1	-1	-2	0	2	**5**	2	0	0	1	-2	-3	-3	-3	-3	-2	-3
Q	-3	0	0	-1	-1	-2	0	0	2	**5**	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	0	-2	-2	-2	1	1	0	0	**8**	0	-1	-2	-3	-3	-2	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	0	-1	-1	-2	0	-1	1	1	-1	2	**5**	-1	-3	-2	-3	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**	1	2	-2	0	-1	-1
I	-1	-2	-2	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**	2	1	0	-1	-3
L	-1	-2	-2	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**	3	0	-1	-2
V	-1	-2	-2	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	2
W	-2	-3	-3	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**

cluster. In addition, the model building and the sequence assignment are iteratively refined until there is no further change in cluster assignment for each sequence [3, 8-10].

To further illustrate the process, assume we want to partition a dataset D into k clusters by using HMMs-based partition clustering. The dataset D contains d aligned sequences of length n, and the character set of the sequences is σ.

Definitions:

- S_i denotes the i^{th} sequence in dataset. ($S_i \in D$, $1 \le i \le d$, $i \in N^+$).
- R_k denotes the set of sequences in cluster k. ($R_k = \{S_i\}_k$).
- M_k denotes the profile HMM for cluster k.
- P_{ik} ($S_i \mid M_k$) denotes the probability of S_i under model M_k.

For each model M_k:

- m_j denotes the j^{th} state in the HMM profile. ($1 \le j \le n$, $j \in N^+$).
- C_{ij} denotes the j^{th} character in i^{th} sequence. ($C_i \in \sigma$, $1 \le i \le d$, $1 \le j \le n$, $i, j \in N^+$).
- p_{ij} denotes the probability of C_{ij} in the m_j state.

Initiation:

Each sequence S_i is assigned to a cluster k according to the clustering result from Phase I.

Iteration:

A profile hidden Markov model M_k is built for cluster k with the probabilities of each alphabet in R_k.

$$M_k = (m_1, m_2, \ldots, m_n) \tag{2}$$

Where, M_k consists of a connected set of states; m_j consists of the probability set at position j. For each state m_j, the probability set is obtained by the following steps:

1. Count the occurrence of each alphabet in σ at position j over all the sequences in R_k.
2. To avoid zero probabilities, we add one to each alphabet count based on the Laplace's rule which is the simplest pseudo count method. Therefore, the denominator of the probability is the sum of the number of sequences in that cluster and the number of alphabets in σ [7].

Compute the probabilities with alphabet counts for each alphabet in σ at position j.

In Figure 3, we demonstrate how to build the profile HMM (M_1) for a cluster C_1 as an example. Assuming that there are three sequences (S_1, S_2 and S_3) in cluster C_1 and the alphabets at the first position (m_1) of the three sequences are H, H, E (H for histidine and E for glutamic acid). For each character (amino acid), we count the number of its occurrences at the first position over all three sequences and calculate its probability at the first position. In this case, we have the number of occurrences of character H equal to 2, the number of occurrences of E equal to 1 and that of all other characters equal to zeros. However, the probabilities of the occurrence of other amino acids cannot be 0s because there might be sequences that fall into this cluster but do not have H or E as the starting characters. To avoid zero probabilities, we use pseudo count, which adds one to each alphabet count. Since the number of alphabets is 20 and the number of sequences is 3, the denominator of the probability is 20+3=23. Therefore, the probability of histidine (H) is 3/23, the probability of glutamic acid (E) is 2/23 and the probabilities of all other amino acids are 1/23 (see row m_1 under column M_1 in Figure 3). After calculating the probabilities for all characters at all positions for the cluster k, the profile Hidden Markov Model M_k is formed as shown in Figure 3.

Cluster	C_1			C_2		M_1							M_2						
Sequences	S_1	S_2	S_3	S_4	S_5	C	T	G	D	E	H	others	C	T	G	D	E	H	others
Position 1 (m_1)	H	H	E	H	E	1/23	1/23	1/23	1/23	2/23	3/23	1/23	1/22	1/22	1/22	1/22	2/22	2/22	1/22
Position 2 (m_2)	C	D	D	C	E	2/23	1/23	1/23	3/23	1/23	1/23	1/23	2/22	1/22	1/22	1/22	2/22	1/22	1/22
Position 3 (m_3)	H	G	G	H	G	1/23	1/23	3/23	1/23	1/23	2/23	1/23	1/22	1/22	2/22	1/22	1/22	2/22	1/22
Position 4 (m_4)	D	T	T	D	T	1/23	3/23	1/23	2/23	1/23	1/23	1/23	1/22	2/22	1/22	2/22	1/22	1/22	1/22
M_1 Score	-4.1	-3.5	-3.7	-4.1	-4.2	Note:													
M_2 Score	-4.2	-4.5	-4.5	-4.2	-4.1	For computation convenience, we take log value of all probability values in M_1 and M_2.													
New cluster	C_1	C_1	C_1	C_1	C_2														

Figure 3. Profile HMM & similarity measurement

The model M_k is then used to compute the P_{ik} ($S_i \mid M_k$) for the cluster k by using the following equation:

$$P_{ik} = \sum_{j=1}^{n} \log p_{ij} \tag{3}$$

The cluster membership of the sequence S_i is then evaluated based on P_{ik} values. The higher the P_{ik} value is, the more likely the sequence S_i belongs to the cluster k. Therefore, we can find the closest cluster for each sequence S_i by finding the $max(P_{ik})$ value via the above formula, and reassign S_i to the k^{th} cluster. This iterative process will continue till no sequence reassignment occurs.

This evaluation process is also illustrated in Figure 3. We assume that there are five sequences S_1, S_2, S_3, S_4 and S_5 and two clusters (C_1 and C_2). In the first iteration, sequences S_1, S_2 and S_3 are clustered into C_1, and sequences S_4 and S_5 are clustered into C_2. The profile HMMs M_1 and M_2 are generated for C_1 and C_2, respectively. In this example, the first character of S_4 is H

and the probabilities of H at m_1 position in M_1 and M_2 are 3/23 and 2/22, respectively. The second character of S_4 is C, the probabilities of which at m_2 position in M_1 and M_2 are 2/23 and 2/22, respectively. For efficiency purpose, we calculate the log values of all probability values in M_1 and M_2. Therefore, the probabilities of S_4 under M_1 and M_2 are calculated as follows:

$$P(S_4|M_1) = \log(3/23)+\log(2/23)+\log(2/23)+\log(2/23) = -4.1 \quad (4)$$

$$P(S_4|M_2) = \log(2/22)+\log(2/22)+\log(2/22)+\log(2/22) = -4.2 \quad (5)$$

S_4 is clustered into C_1 since $P(S_4|M_1) > P(S_4|M_2)$. The same process is applied to all sequences. This iterative process will continue till there is no further change made to clusters.

To deal with the empty cluster problem in *k*-means, the new clustering result at the end of each iteration will be examined for emptiness. Once a cluster becomes empty, we will move the one sequence which has the lowest probability of belonging to its original cluster, to that empty cluster.

3. Experimental results

In this paper, we use a dataset containing 217 protein sequences from 9 families of cytochrome P450. We tested the proposed hybrid clustering algorithm over the dataset and generated nine clusters. The performance of the proposed clustering method was evaluated by the F-measure as defined in [14]. In the definition detailed below, C_i is the right cluster according to its actual protein family and C_j is the cluster obtained from the clustering results. The higher the F-measure value is, the better the clustering result is.

$$F = \sum_{i=1}^{n} \frac{|C_i|}{N} \times F(i), F(i) = \max_{i=1}^{m} F(i,j), F(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)}$$

$$P(i,j) = \frac{|C_i \cap HC_j|}{|HC_j|}, R(i,j) = \frac{|C_i \cap HC_j|}{|C_i|}$$

Based on our experiments on 217 sequences, as shown in Table 2, the F-measure of the hierarchical clustering result is 0.53, which is the worst among all methods. We then compare the proposed hybrid algorithm with a modified *k*-means algorithm which uses random initial partitions. The standard *k*-means algorithm is modified to ensure that empty clusters are avoided by using the algorithm presented in Section 2.2. We ran the modified *k*-means partition clustering on the same dataset for twenty times. The best and the worst cases are both shown in Table 2. It can be observed that under the modified *k*-means clustering, different clustering results were produced since the initial partition is random – the algorithm randomly assigns each sequence to one of the nine clusters initially. The F-measure of the worst case is 0.56 and the average F-measure is about 0.60.

In Table 2, we also show the F-measure of the proposed hybrid algorithm. The F-measure of the hybrid method is 0.65. Obviously, the result of the hybrid algorithm is better than that of the hierarchical methods, and more robust than that of the modified *k*-means methods. In addition, compared with the standard *k*-means algorithm, the proposed approach can avoid the problem of empty clusters.

To summarize the experimental results, we observe that the performance of *k*-means partition clustering is varied since the initial partition is random. In addition, pre-clustering sequence data by using hierarchical methods can greatly improve the accuracy and the robustness of partition clustering algorithms. Furthermore, the inaccuracy of the hierarchical agglomerative clustering can be compensated by the profile HMM based *k*-means partition clustering since the model-based partition clustering can better describe the dynamic properties of the data in a cluster.

It is worth pointing out that after the HMM profiles for clusters are built and saved in the database, we can predict the category for an unseen protein sequence by finding the maximum similarity score between the new sequence and the existing HMM profiles. There is no need to compute the similarity between the new sequence and every other sequence in the dataset. Each time a new sequence is inserted, only the HMM profile of the cluster in which the new sequence was added will be updated. Therefore, with this hybrid clustering method, a lot of time can be saved, especially when the size of the sequence database increases rapidly.

Table 2. The F-measure of different clustering methods

Family	Hybrid method	Hierarchical method	*k*-means (worst case)
1	0.11	0.03	0.12
2	0.23	0.28	0.30
3	0.08	0.02	0.08
4	0.17	0.17	0.17
5	0.01	0.00	0.01
6	0.04	0.01	0.04
7	0.01	0.02	0.02
8	0.00	0.00	0.00
9	0.00	0.00	0.00
Overall	0.65	0.53	0.56

4. Conclusions and future work

The proposed hybrid method for biological sequence clustering takes the advantages and overcomes the shortcomings of the hierarchical clustering algorithms and the partition clustering algorithms. This is evidenced by

the fact that the proposed method produced a more robust result, when compared with standard hierarchical methods and partition clustering methods.

Currently, our algorithm can only deal with the aligned sequences. However, in many cases, sequences for clustering are in different lengths. This will cause a problem to the HMM profiling method. More work need to be done in order to accommodate this need for flexibility.

To further improve clustering results, we believe that increasing the weight for the functional domains might help since it is a known fact that protein function is determined by its domains, the three-dimensional structures. The three-dimensional structures are constructed based on the primary sequence. Hence, proteins that have similar functions and are classified into the same family usually have similar sequence. Based on this assumption, if we can discover the functional domains of a protein from its primary sequence, then we can increase the weight for the regions that contain the functional domains when building the profile HMM to represent a protein family.

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

[4] M. Linial, N. Linial, N. Tishby, and G. Yona, "Global self-organization of all known protein sequences reveals inherent biological signatures," *J Mol Biol*, vol. 268, pp. 539-56, 1997.

[5] Y. Zhao and G. Karypis, "Clustering in life sciences," *Methods Mol Biol*, vol. 224, pp. 183-218, 2003.

[6] F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic Acids Res*, vol. 16, pp. 10881-90, 1988.

[7] 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.

[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. Mian and I. Dubchak, "Representing and reasoning about protein families using generative and discriminative methods," *J Comput Biol*, vol. 7, pp. 849-62, 2000.

[11] D. Gusfield, *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge, UK: Cambridge University Press, 1997.

[12] N.C. Jones and P.A. Pevzner, *An Introduction to Bioinformatics Algorithms*. Cambridge, Mass: MIT Press, 2004.

[13] R. Spang and M. Vingron, "Limits of homology detection by pairwise sequence comparison," *Bioinformatics*, vol. 17, pp. 338-42, 2001.

[14] 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.