Biological Sequence Clustering and Classification with a Hybrid Method and Dynamic Programming

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

Abstract

In this paper, we report a framework for biological sequence clustering and classification. The proposed framework adopts a two-phase hybrid method for clustering, and then uses the dynamic programming technique for classification. The two-phase hybrid method combines the strengths of the hierarchical and the partition clustering. Phase I of the hybrid method uses the hierarchical agglomerative clustering to pre-cluster the aligned sequences. Phase II performs the partition clustering which initiates its partition based on the result from Phase I and uses profile Hidden Markov Models (HMMs) to represent clusters. The profile HMMs are then stored in the database for unknown sequences classification, which is done by finding the best alignment of a sequence to each existing profile HMM. However, the profile HMMs and the sequence might be different in length. The dynamic programming technique proposed in our framework can efficiently find the optimal alignment for sequences of variable lengths, which enables the evaluation of the cluster membership for any unknown sequence against fixed-length HMMs. Our experiments demonstrate the effectiveness and the efficiency of the proposed framework for biological sequence clustering and classification.

Keywords:

Sequence clustering, classification, prediction, hybrid clustering, hierarchical clustering, partition clustering, profile HMM, dynamic programming, *k*-means.

1. Introduction

The knowledge of the structures and functions of biological sequences can be of great help to drug design and disease treatment. Researchers have to perform a large amount 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 an unknown 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. In this paper, we mainly discuss two commonly used methods: the hierarchical clustering methods and the partitioning methods [2, 3].

The hierarchical clustering methods group data objects into a hierarchical tree on the basis of the similarity (or distance) between data objects in either bottom-up (agglomerative) or top-down (divisive) fashion. The advantages of hierarchical clustering methods are that they can be used to cluster a wide variety of datasets and can 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 $O(n^3)$ [4, 5].

Another type of commonly used clustering methods discussed in this paper is partitioning methods. Among this category, *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 uses seeds to represent the cluster centroids. Algorithms in this category usually employ an iterative relocation strategy which moves the data objects in one cluster to another according to some similarity measure till there is no further move and finally forms *k* clusters. In the iterative process, clusters are gradually refined with the goal to maximize the similarity within a cluster and minimize the similarity between clusters, and the centroids are updated to reflect the cluster changes. To generate the new centroids/seeds, both the similarity-based methods and model-based methods are widely used. The similarity-based methods generate a new seed by examining the similarity score between each pair of objects in the cluster, while the model-based methods generate a new seed by building a model to represent the cluster [5]. The model-based methods thus 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 Profile Hidden

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

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 [6-9]. The main advantage of partition methods is the ability to build models for clusters which makes it more valuable especially for frequently updated dataset. Furthermore, partition clustering have a time complexity of $O(n^2)$ [5].

However, problems are found in partition clustering methods. Since initial seeds are selected randomly, we might get diverse results while running the partitioning methods on the same dataset for more than once. Another problem is that an empty cluster could be formed in the iterative process, when each sequence in that cluster is found to have a higher similarity score in another cluster. Thus, no sequence can be assigned to that cluster anymore.

Our goal is to develop a robust yet efficient framework for biological sequence clustering and classification. In the clustering stage, the proposed framework performs a two-phase hybrid clustering algorithm, which combines the strengths of the hierarchical agglomerative clustering and the partition clustering to build a profile HMM for each cluster. In the classification stage, the main challenge for HMM based methods is how to classify an unseen sequence, which is not aligned to the same length of the sequences used to build the models, without completely re-building the models. In this paper, we propose a dynamic programming based method to align a sequence with arbitrary length to a profile HMM.

Dynamic programming is a widely used technique in finding the optimal solutions for dynamic optimization problems. Dynamic programming algorithms, such as Needleman-Wunsch and Smith-Waterman, have been applied to solving the global and local sequence alignment of two sequences in bioinformatics [6, 10]. With dynamic programming, we can quickly find the optimal alignment of two sequences or the most probable path of a sequence against a model [6-8]. With this framework, there is no need to compute the similarity between the new sequence and every other sequence in the dataset. The time complexity of the classification process can be significantly reduced, especially when the size of the sequence database increases rapidly.

For clustering, our experimental results demonstrated that the inaccuracy of the hierarchical agglomerative clustering can be compensated by the profile HMM based 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 prevent the subsequent clustering process from getting stuck in a bad local minimum. For classification, our results show the robustness of the proposed scheme based on HMM profiles.

Section 2 discusses the details of the two-phase hybrid clustering and the classification process by dynamic programming. Section 3 presents the experimental results. Section 4 concludes this paper.

2. The proposed framework

The proposed framework includes the clustering and the classification stages. The clustering step adopts a two-phase hybrid method to cluster and build the profile HMMs for the aligned sequences in the training set. The profile HMMs are stored in the database and used for subsequent classification. The classification step applies dynamic programming to evaluate the cluster memberships of unseen sequences in the testing set.

2.1. Clustering stage: a two-phase hybrid method

The two-phase hybrid clustering algorithm we proposed combines the strengths of both the hierarchical agglomerative clustering algorithms and the partition clustering algorithms. This two-phase hybrid clustering algorithm takes the imported sequences in the training set and the parameter *k* from the user as its input. In Phase I, the hierarchical agglomerative clustering algorithm starts by placing each object in its own cluster and then gradually merges these atomic clusters into larger and larger clusters. This process continues till *k* clusters are formed. Phase II takes the pre-clustering result from Phase I as its initial partition, and then generates the pseudo seeds as cluster centroids by using the HMM profile. The algorithm involves an iterative process to minimize the inter-cluster similarity and maximize the intra-cluster similarity. This process continues till there is no further change in the cluster membership for any sequence. Figure 1 illustrates the concept of the proposed hybrid algorithm.

2.1.1. Hierarchical agglomerative clustering. In Phase I of the hybrid method, the hierarchical agglomerative clustering method initially places each sequence in the training set 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 [4, 5].

In the merging process, we need to construct a similarity matrix to measure and store the similarity between two clusters. By the single-link approach, the similarity between two clusters is measured by the similarity of the most similar pair of sequences from each of these two clusters. The similarity S between two sequences can be measured by Equation 1 [10, 11]:

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

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 in our case; 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 the two sequences.

In Phase I, the termination condition of the algorithm is "till the desired number of clusters is obtained". In other words, 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).

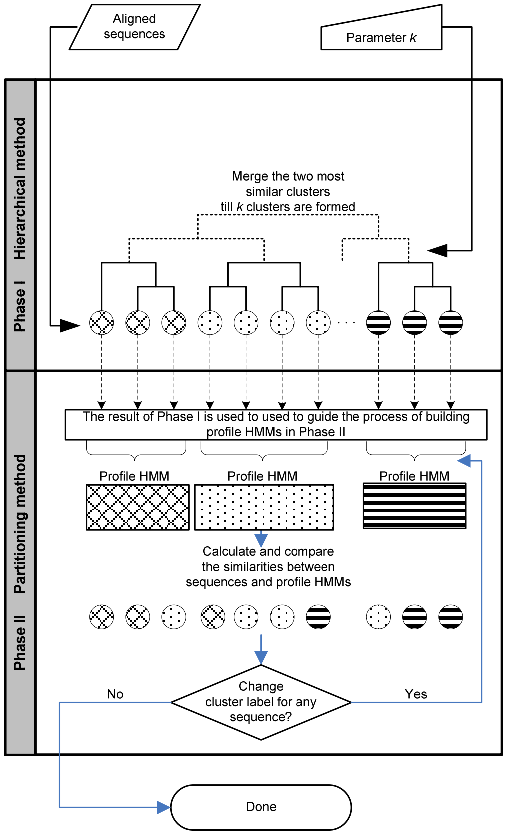

Aligned sequences
Parameter k
Phase I Hierarchical method
Merge the two most similar clusters till k clusters are formed
Phase II Partitioning method
The result of Phase I is used to used to guide the process of building profile HMMs in Phase II
Profile HMM
Profile HMM
Profile HMM
Calculate and compare the similarities between sequences and profile HMMs
Change cluster label for any sequence?
No
Yes
Done

Figure 1. The flowchart of the two-phase hybrid clustering algorithm

2.1.2. Profile HMM based partition clustering. In Phase II, instead of using random initial assignments of sequences, the partition clustering method takes the pre-clustering result from Phase I as the initial partition of sequences. A probability model, the profile Hidden Markov Model, is then built based on the initial partition. The partition clustering will use this probability model to model each cluster and reassign each sequence to a cluster according to the underlying HMMs. In addition, the model building and the sequence reassignment are iteratively refined until there is no further change in cluster assignment for any sequence in the training dataset [5, 7-9].

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.

Initialization:

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

Iteration:

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

$$M_k = (m_1, m_2, \ldots, m_n) \qquad (2)$$

where M_k consists of a connected set of states; m_j consists of the set of probabilities of alphabets 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 the 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. The denominator of the probability is the sum of the number of sequences in that cluster and the number of alphabets in σ [6].
3. Compute the probabilities for each alphabet in σ at the position j.

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

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

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

The cluster membership of the sequence S_i is then evaluated based on its 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 best-match cluster for each sequence S_i by locating its $max(P_{ik})$ value via the above formula, and reassign S_i to the k^{th} cluster. This iterative process will continue till no sequence reassignment occurs.

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.

2.2. Classification stage: a dynamic programming method

The classification is the process of finding the best-match cluster for an unknown sequence among all the profile HMMs stored in the database. The profile HMMs stored in the database are fixed in length, so that when aligning a sequence with a profile HMM, it is possible that the length of the unknown sequence does not match that of the profile HMMs. We solve this problem by finding the most probable path that a sequence may take against the model with a dynamic programming based method.

When the length of a sequence is longer than that of the profile HMMs, one or more "profile HMMs state insertions" will occur. To handle this situation, we use the following strategy. Since the size of the character set is 25 (including the gap), and the insertion of HMM states eliminates the probability of gaps in the sequence, the possible character at the position that corresponds to a new state must be one of the other twenty-four characters. Thus, the penalty for inserting a state into the model is represented as the log value of 1/24, which is (-1.380211242). This strategy allows aligning a sequence whose length is longer than the profile HMMs.

To exemplify the classification process with dynamic programming, we define U_i as the i^{th} sequence in the testing set and $C_{i,m}$ as the m^{th} character in U_i. The k^{th} profile HMM in the database is denoted by M_k, and $m_{k,n}$ is the n^{th} state in M_k. For a sequence U_i and a profile HMM M_k, we define $D(m, n)$ as the highest similarity score of the subsequence $C_{i,1}$ ... $C_{i,m}$ against the subset of the model states $m_{k,1}$… $m_{k,n}$, and let $p_{k,n}(C_{i,m})$ be the log probability of $C_{i,m}$ under the state $m_{k,n}$.

To obtain the value of $D(m, n)$, with dynamic programming, we use a matrix (D) with a size of $(s+1)\times(t+1)$, where s is the length of the sequence being tested and t is the number of states in the profile HMM. A typical dynamic programming algorithm consists: (1) initialization, (2) recurrence, and (3) traceback [6, 10].

The first step is initialization which defines the base conditions. It specifies the initial values in the first row and column of the dynamic programming matrix D. In our case, the base conditions are defined as below:

$$\begin{cases} D(0,0) = 0 \\ D(m,0) = D(m-1,0) + (-1.380211242) \\ D(0,n) = D(0,n-1) + p_{k,n}('-') \end{cases} \qquad (4)$$

where the (-1.380211242) is the penalty for inserting a state into the model.

The second step is recurrence which establishes the recursive relationship between $D(m, n)$ and its top $D(m\text{-}1, n)$, left $D(m, n\text{-}1)$, and upper-left $D(m\text{-}1, n\text{-}1)$ neighbors.

$$D(m,n) = \max\begin{cases} D(m-1,n) + (-1.380211242) \\ D(m,n-1) + p_{k,n}('-') \\ D(m-1,n-1) + p_{k,n}(C_{i,m}) \end{cases} \qquad (5)$$

where the $p_{k,n}$("–") is the log probability of a gap/space under the state $m_{k,n.}$ Thus, after the entire matrix D is filled according to the above rules, the value in $D(s, t)$ denotes the maximal similarity score of the sequence U_i aligned with the profile HMM M_k.

The last step in dynamic programming is traceback. It extracts how the two sequences are aligned or how the sequence is aligned to the model by tracing the pointers, which are obtained in the recurrence step, starting from the pointer of $D(s, t)$.

In our implementation, we do not include the traceback step because only the maximal similarity score is needed for classification purposes, while the specific alignment of the sequence with the HMM models is not of our concerns. As aforementioned, the final similarity score is located at the bottom-right corner of the dynamic programming matrix D, which represents the similarity value between the sequence and the model. Thus, in this case, the space complexity is $O(s)$ because there is no need to keep all the similarity scores in the memory.

3. Experimental results and discussion

The sequence dataset used in our experiments contains 429 protein sequences from 65 families of cytochrome P450. The aligned sequences are obtained from a public website (http://drnelson.utmem.edu/CytochromeP450.html) which is maintained by David Nelson at The

University of Tennessee, Memphis, College of Medicine, Department of Biochemistry.

3.1. The evaluation of the hybrid clustering

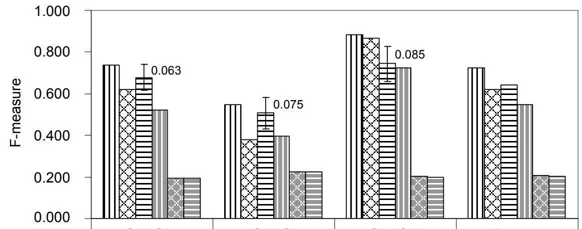
The performance of the proposed two-phase hybrid clustering method is evaluated by the F-measure [12]. The higher the *F*-measure value is, the better the clustering result is. Three subsets of sequences are randomly selected from the sequence dataset. The three subsets are named by their content. For example, the dataset S135C17 contains 135 sequences from 17 families.

In our experiment, the proposed hybrid clustering method is compared with five other methods – the hierarchical agglomerative clustering, the standard *k*-means partition clustering, and three other similarity based clustering, including: direct *k*-way clustering, repeated bisections (R.B.), and global optimized repeated bisections (R.B.R.), which are implemented in the "CLUTO" which is a software for clustering high-dimensional datasets [13, 14]. Figure 2 demonstrates the performance of the six clustering methods. The average F-measure of the proposed hybrid method is 0.723, which is significantly better than that of the hierarchical agglomerative clustering (0.622), the standard *k*-means partition clustering (0.643), the direct *k*-way clustering (0.547), the repeated bisections (0.205), and the global optimized repeated bisections (0.204).

In addition, the experimental results also show the instability of the standard *k*-means. Since the standard *k*-means has its initial seeds selected randomly, the results vary according to different initial conditions and are thus unpredictable. We ran the standard *k*-means twenty times to obtain an average result. However, the average result generated by standard *k*-means is still significantly worse than that of the proposed hybrid method.

Furthermore, we also tested the performance of the profile HMMs based partition clustering algorithm (with a random initial partition) which performs only the second phase of the hybrid method. The experimental result shows that the average values of F-measure of the profile HMMs based partition algorithm on the three testing datasets are 0.163 ± 0.500, 0.403 ± 0.157, and 0.306 ± 0.152, respectively.

In our experiment, the proposed two-phase hybrid clustering method produces the best result among all the methods used in the comparison, which is a direct result of combining the strength of the two types of clustering algorithms: the hierarchical agglomerative clustering and the partition clustering methods.



	S135C17	S217C9	S149C13	Average
◫ Hybrid	0.738	0.549	0.883	0.723
☒ Hierarchical	0.622	0.378	0.865	0.622
☰ K-means	0.678±0.063	0.508±0.075	0.744±0.085	0.643
▥ Direct	0.522	0.395	0.725	0.547
▩ R.B.	0.192	0.222	0.201	0.205
▤ R.B.R.	0.192	0.222	0.197	0.204

Figure 2. Comparison results

3.2. The accuracy of the classification

Two datasets used in our experiments are randomly selected from the sequence dataset containing 429 protein sequences from 65 families of cytochrome P450. The first dataset contains 109 sequences from 8 protein family and the second dataset includes 113 sequences from 12 protein families. We applied the proposed method on six pairs of training set and testing set with a training-testing ratio of 2:1. The sequences in the training sets and the testing sets of the pairs 01~03 and the pairs 04~06 are randomly selected from Dataset 1 and Dataset 2, respectively. In addition, the sequences in the testing datasets may not have the same length as that of the sequences in the training sets. Table 1 shows the accuracy of the proposed classification algorithm on the six training/testing sets.

Table 1. The accuracy of the classification algorithm

		Number of sequence			
		Training set	Testing set	Correctly classified	Accuracy
Dataset 1	Pair 01	72	37	36	97.30%
	Pair 02	73	36	35	97.22%
	Pair 03	73	36	36	100.00%
Dataset 2	Pair 04	73	38	31	81.58%
	Pair 05	74	38	34	89.47%
	Pair 06	76	37	37	100.00%
Average					94.26%

As shown in Table 1, the average accuracy of the proposed classification algorithm over the six pairs of training/testing sets is 94.26%. The classification results evidence that the proposed dynamic programming is effective in classifying unseen sequences for HMM profile based approaches.

In addition, it is worth pointing out that the proposed framework is time efficient. It stores the profile HMMs in a database and retrieves the profile HMMs from the database when needed, which is fast and guarantees the data persistency. The dynamic programming method used in the proposed framework provides a way for aligning a sequence with an arbitrary to the profile HMMs stored in the database. The time complexity of the dynamic programming is $O(a \times b)$, where a is the length of the unknown sequence and b is the length of the profile HMM. If the lengths of both sequence and profile HMM is about 'n', then the time complexity is $O(n^2)$, which is more efficient than that of the hierarchical agglomerative clustering algorithm $O(n^3)$. In addition, there is no need to reconstruct the entire tree as the hierarchical agglomerative clustering algorithm does.

4. Conclusions and future work

The proposed framework for biological sequence clustering and classification is not only effective but efficient. Its effectiveness is demonstrated by the more robust results produced by the proposed two-phase hybrid clustering method. The two-phase hybrid method for sequence clustering takes the advantages of the hierarchical agglomerative clustering and the profile HMMs based partition clustering and avoids the problems arisen when using any of them alone.

To further improve clustering results, we believe that increasing the weight for the functional domains might help bring us closer to the goal 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 structures. Based on this assumption, if we can discover the functional domains of a protein from its primary sequence, then we can increase the weight in a profile HMM for the regions containing the functional domains.

In addition, when a new sequence is inserted into a cluster, we can dynamically update the profile HMM of that cluster without recalculating the similarity between the newly inserted sequence and every other sequence in the entire dataset. With the proposed framework, we can simply update the model based solely on its member sequences. Since the number of sequences to be updated in the cluster is way less than the number of sequences in the entire dataset, the profile HMM of that cluster can be updated with little cost. However, the new sequence might be aligned to the profile HMMs in several different ways since more than one path might be found in the trace back step in dynamic programming. Therefore, how to trace back all paths efficiently and to identify which path to use for model update is another issue we need to address in our future work.

5. Acknowledgement

The work of Chengcui Zhang was supported in part by her IBM UIMA Award.

6. References

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

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

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

[4] F. Corpet, "Multiple sequence alignment with hierarchical clustering", Nucleic Acids Res., Oxford University Press, vol. 16, Nov. 1988, pp. 10881-90.

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

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

[7] S.R. Eddy, "Hidden Markov models", Curr. Opin. Struct. Biol., Current Biology, vol. 6, Jun. 1996, pp. 361-5.

[8] S.R. Eddy, "Profile hidden Markov models", Bioinformatics, Oxford University Press, vol. 14, Oct. 1998, pp. 755-63.

[9] I.S. Mian and I. Dubchak, "Representing and reasoning about protein families using generative and discriminative methods", J. Comput. Biol., Mary Ann Liebert, Inc., vol. 7, Dec. 2000, pp. 849-62.

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

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

[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, Oxford University Press, vol. 20, Nov. 2004, pp. 2534-44.

[13] Y. Zhao and G. Karypis, "Criterion function for document clustering", University of Minnesota, Department of Computer Science / Army HPC Research Center Technical Report #01-40, Apr. 2003.

[14] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets", Data Mining and Knowledge Discovery, Vol. 10, 2005, pp. 141-68.