Title

Clustering Techniques for Data Mining and Protein Design Around The Concept of Locality

Author

Hakkoymaz, Huseyin

Publication Date

2010

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Clustering Techniques for Data Mining and Protein Design Around the Concept of Locality

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Huseyin Hakkoymaz

August 2010

Dissertation Committee:

Professor Eamonn Keogh, Co-Chairperson
Professor Dimitrios Morikis, Co-Chairperson
Professor Michalis Faloutsos
Professor Vassilis Tsotras

The Dissertation of Huseyin Hakkoymaz is approved:

____

____

____
Committee Co-Chairperson

____
Committee Co-Chairperson

University of California, Riverside

ACKNOWLEDGEMENTS

I am indebted to each of the following faculties, colleagues, friends and family members who contributed to this dissertation. Without their kind support and strong encouragements, it would not have been possible for me to accomplish this doctoral thesis.

First, I would like to thank my former adviser, Prof. Dimitrios Gunopulos, for his support during all those year when I did my research under his direction. Thanks to his intellectual guidance, I found the passion of doing academic research and the way to channel my creative energy in a fruitful way in academia. He not only gave valuable advices in my research, but also helped me get on the right track whenever I faced the ups and downs and was unable to focus in the process.

I also thank my co-adviser, Dr Morikis, who guided me into wonderful world of the biology and broadened my horizon in drug design. Thanks to him, I had a great opportunity to do a research in an interdisciplinary field. His dedication and commitment to his students gave me a chance to see how advising should be done in doctoral studies and created a vision for my academic life.

I am highly indebted to Dr Eamonn Keogh, who has been, and always will be, a great mentor during the last year of my doctoral studies. Despite of his busy schedule, he has always offered me great help, both materially and psychologically. During our numerous and fruitful discussions, I have understood that running experiments and getting results was just the half of doing research, where the other half was to present the

results to the world in a neat and understandable way. Thanks to his support and guidance, writing is no longer hassle for me as it used to be.

I would like to thank Dr Vassilis Tsotras and Dr. Michalis Faloutsos for many helpful and interesting discussions during their lectures. By applying their teachings, I managed to get into one of the most prestigious companies in the world. It is a pleasure to thank them for agreeing to be on my dissertation committee and reviewing my thesis.

I would also like to thank Georgios Chatzimilioudis for sharing the burden of doing research in the DBLAB till mornings and for helping me run the experiments. He made the cold winter nights more enjoyable with his day brightening personality. I will miss our chats and all-niters in the lab, not to mention our soccer games in the field.

I would also like to extend my gratitude to all the people in the BioMoDeL Lab who showed me the ropes in drug design after the participation to the group. Especially, I want to thank Ron Gorham and Chris Kieslich for providing me the data for the experiments and for giving valuable feedbacks on the results.

A special thank goes to Amy Ricks and Terri Phonharath, as well as all other administrative staff, for their support and assistance since the start of my doctoral studies in 2005.

My deepest gratitude goes to my family, who supported me during all stages of my education. Their support has been invaluable above all and thanks to them, I have found power and courage to fight back whenever I fall into deep well of the depression and despair during my life.

ABSTRACT OF THE DISSERTATION

Clustering Techniques for Data Mining and Protein Design
Around The Concept of Locality

by

Huseyin Hakkoymaz

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, August, 2010
Professor Eamonn Keogh and Professor Dimitrios Morikis, Co-Chairpersons

Advances in technology have expedited the use of acquisition ability in computers to obtain data from diverse sources via sensors or imaging techniques with high throughput. The collected data usually tend to be extremely large, and processing a large volume of data requires computationally intensive resources. Clustering techniques simplify the data by partitioning it into meaningful groups and allow us to analyze a large volume of data in a relatively short period of time with high accuracy.

This dissertation introduces several novel approaches that improve the performance of semi-supervised and unsupervised clustering by utilizing the concept of locality. It makes two specific contributions:

1. *Magnetically Affected Paths*: A novel approach to apply the user-defined constraints through local manipulations in semi-supervised clustering. MAP refines the clustering results by increasing the weight of the edges connecting

the objects that are in the neighborhood of a cannot-link constraint, and decreasing the weight of the edges connecting the objects that are in the neighborhood of a must-link constraint. MAPClus framework introduced in this dissertation integrates the MAP concept into the clustering algorithms by applying a three-step algorithm. The efficacy of the algorithm is demonstrated through extensive experimental evaluations on several synthetic and real datasets.

2. *Wavelet-Based Similarity Measures*: A family of similarity measures which exploits the ability of wavelet transformation to analyze the spectral components of the physicochemical properties and suggests a more sensitive way of measuring the similarity of biological molecules. We demonstrate the validity of our wavelet-based similarity measures by employing them in two different protein clustering applications. In the first set of experiments, we use the measures to identify the relationships between mutant proteins that were obtained by alanine scanning. Additionally, we present how accurate our methods are in recognizing the connection between charge density and electrostatic potential in homology models.

Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

One of the many goals of artificial intelligence is to imitate the human mind as much as possible in order to solve the problems that involve vast amounts of data. In the last 30 years, advances in technology have expedited the use of acquisition ability in computers to obtain data from diverse sources via sensors or imaging techniques with high throughput. The collected data usually tend to be extremely large, and processing a large volume of data requires computationally intensive resources. For example, LSST (Large Synoptic Survey Telescope) generates 30 terabytes of astronomical data daily [1]. As for drug design, ab initio methods produce a large number of protein sequences for which no tertiary structure information is available [2]. Typically, collecting or generating the raw data is relatively easy and just the tip of the iceberg. However, a significant problem still persists after all these years: the need for efficient and novel techniques that analyze the data to explore useful information.

Data mining is the process of extracting useful information in order to discover hidden patterns and relationships which may lead to a better understanding of the data [3]. In spite of the vast amount of the data generated every day, the raw data is usually not useful for many applications. In data mining tasks, we generally assume that the dataset consists of a collection of instances. Each instance is described by a set of features, which can be numeric or categorical and can vary from a few to thousands in number, depending on the domain. Data mining transforms the raw data into useful information by analyzing and correlating the instances based on their features. It offers potential benefits for understanding the data and domain-specific decision making.

1.1 **Clustering**

Clustering is an important data mining task which partitions the data into meaningful groups without advance knowledge of the relationships between the data elements [3]. From a machine learning perspective, clustering methods search for hidden patterns and systematic relationships that reveal unknown characteristics about the data. These patterns and relationships then can be used in numerous applications ranging from information retrieval [4] [5] [6], database applications [7], market analysis[8], medical diagnosis[9], and bioengineering [10] to scientific data exploration such as satellite imagery[11]. Due to its practicality and efficiency, clustering is one of the most important analysis tools in data mining. It simplifies the data representation and helps us understand the natural grouping and structures in a dataset.

Clustering utilizes the techniques of many active research fields, such as statistics, machine learning and pattern recognition, while discovering the patterns. The primary goal of clustering is usually clustering accuracy, which is followed by efficiency, interpretability, generalizability, and ease of use [12].

Clustering methods can be roughly divided into three groups:

1. Unsupervised Clustering: Unsupervised clustering seeks to determine how the data is organized by using unlabeled examples only; i.e., no a priori knowledge about the data is available. Such methods rely on statistical data analysis and are expected to perform poorly in contrast to supervised clustering.

2. Supervised Clustering: This approach learns a clustering function from training data and uses this function to predict the value of the unlabeled data. It assumes that the examples are classified as desired. Thus, the clustering performance is relative to the quality of examples and the function learned.

3. Semi-Supervised Clustering: Semi-supervised clustering makes use of both labeled and unlabeled data for training, unlike traditional supervised and unsupervised classifiers. Labeled data is usually expensive and hard to obtain since it is gathered either from human experts or from well-studied measurements, whereas unlabeled data is relatively inexpensive and usually available in large amounts. The use of labeled data is often critical to the success of the clustering process. Semi-supervised clustering approaches improve the clustering performance considerably when the problem involves a large amount of unlabeled data and a small amount of labeled data.

In this thesis, we focus on the techniques that potentially increase the performance of unsupervised and semi-supervised clustering. Unsupervised clustering mostly depends on the quality of the statistical methods chosen. In some cases, even the state-of-the-art methods can be unsuccessful in finding a good clustering [13]. One reason for this deficiency is that the characteristics of the data may not be compatible with the objective function of the clustering algorithm. We can overcome this problem by mapping the data into another domain where we can increase the compatibility level [14][15]. In the following chapters, we shall present two methods, namely vector-to-graph transformation and wavelet transformation, which may be used for such mapping to improve the clustering quality. Semi-supervised clustering methods which use the former mapping approach in our studies suggest a significant improvement in terms of accuracy over unsupervised clustering when unlabeled data is used in conjunction with a small number of labeled data. The labeled data is considered as user knowledge that creates a positive bias on clustering function, thus improving the clustering result. An important problem is how to describe the user knowledge; we need a technique that is easy to use but also possible to generalize. Typically, one of two methods is used is to specify constraints on pairs of objects: either *must-link* (two objects must be in the same cluster) or *cannot-link* (two objects must be in different clusters).

1.2 **Locality Concept in Data Mining**

In statistics, the locality indicates the central tendency of a particular dataset[16]. This definition can be extended to the tendency of spatially close by or functionally relevant objects. In data mining, the locality is usually mentioned in the context of local feature relevance methods which attempt to select the most relevant features for distance determination as a counter-attack to the *curse of dimensionality* [17]. The curse of dimensionality refers the phenomenon in which the data become extremely sparse in high-dimensional spaces and are far apart from each other. Due to the high bias effect of the curse, distance functions lose their usefulness in high dimensionality [18]. While this may be true, recent studies reveal that not all dimensions are equally relevant to the class probability functions; i.e., some are more discriminative than others in determining the class of a given data point [19][20]. Local feature selection that analyzes the data in advance estimates different degrees of relevance for the dimensions given in feature space. This approach strengthens the relationships of relevant objects as a result of eliminating the noise in the data [21].

In this dissertation, we introduce an important term called a *local manipulation approach* for data mining applications. The main conjecture motivating this approach is that the large systems can be manipulated by local changes in order to achieve a specific objective. With this in mind, the approach involves a two-step model: first, small modifications are carried out on a large system and then their effects are observed on the whole system from a holistic perspective. While performing local manipulations, the method directly targets the raw data without any prior cluster or global information and adjusts the relationships among data elements using the insight of user-defined constraints.

In the following chapters, we shall examine several novel clustering techniques derived from two aspects of the local manipulation: *metric* and *data*. From the first point of view, the distance measures can be further improved in quality when a local manipulation method is integrated into the distance function. This function can then be employed by a clustering algorithm to measure the pair-wise distances more precisely, which in turn leads to better clustering quality. *Magnetically Affected Paths (MAP)* realizes this idea and helps the distance functions perform more sensitive measurements. The objective of the MAP is to increase the weight of the edges connecting the objects that are in the neighborhood of a cannot-link constraint, and to decrease the weight of the edges connecting the objects that are in the neighborhood of a must-link constraint. This method establishes an interesting analogy between electromagnetic field theory and graphs. In physics, charged objects produce an electric field which exerts a force on other uncharged objects in space [22]. This situation forces uncharged objects to resonate with the charged objects and to show similar electromagnetic behavior as the charged objects. The MAP concept simulates the same characteristics on user-defined constraints and data in a graph domain, where constraints correspond to the charged objects and edges correspond to the uncharged objects.

From a data perspective, we define and investigate several locality patterns in which a subset of a large dataset locally exhibits a similar behavior. These patterns are as follows:

1. *Proportionality*: This pattern indicates a change in the magnitude of data values within a local region of a large dataset due to a peculiar event. The change occurs in the same way for all local points and is measured by the proportionality constant k as $\varphi_A = k.\varphi_{B.}$ Here, φ_A and φ_B indicate the former and latter values, respectively.
2. *Displacement*: This pattern is observed when a particular region is displaced in space, while the rest of data points remain the same. In this pattern, the magnitude of data values never changes. The distortion in the pattern is measured by the displacement angle α based on a pre-defined axis origin.
3. *Scaling*: The third pattern exhibits an expanding or a shrinking behavior in the area of a particular region. The magnitude of data values within the boundary of the region remains the same. The scaling is measured by the scaling ratio S, as in the equation $r_A = S.r_B$, where r_A and r_B correspond to the former and latter radius of the region.

The investigation is carried out on an electrostatic potential distribution of biological molecules to see the effects of these patterns on molecular similarity. To this end, we present a family of similarity measures which extends the previously established Linear [23], Hodgkin [24], and Carbo [25] similarity functions with multi-resolution analysis (MRA) in order to recognize these patterns and account for them in similarity calculations. These measures apply an appropriate discrete wavelet transformation to the molecular electrostatic potential distributions to find the corresponding wavelet coefficients. Subsequently, they perform the comparison using the wavelet coefficients.

By using this approach, we can analyze the spectral components of electrostatic potential distributions at different resolutions. According to our systematic evaluations, the MRA-based approach can potentially increase the sensitivity of similarity measures for locality patterns and provide more accurate similarity values.

1.3 Dissertation Overview

The rest of this thesis is structured as follows. Chapter 2 provides a literature review for the state-of-the-art clustering methods in data mining. Moreover, we present some background information to allow the reader to understand the data mining concepts such as semi-supervised learning, graph clustering, multi-resolution analysis, and wavelet decomposition.

Chapters 3 and 4 describe the metric aspect of the locality in more detail. Chapter 3 gives a formal definition of *Magnetically Affected Paths* and discusses the implementation issues of the idea in a graph domain. Chapter 4 outlines the MAPClus framework, which integrates the MAP concept into the clustering algorithms. This section explains a three-step algorithm proposed for performing semi-supervised clustering on both vector and graph data. This chapter also serves to experimentally validate the claims of efficiency and accuracy for the MAPClus algorithm.

Chapter 5 proposes three molecular similarity measures tailored toward accounting for the locality patterns mentioned in the previous section. This chapter investigates these similarity measures with respect to their support for different patterns by conducting thorough experiments on toy data models.

In Chapter 6, we demonstrate the validity of our MRA-based similarity measures by employing them in two different protein clustering applications. In the first section of this chapter, we use the measures to identify the relationships between mutant proteins that were obtained by alanine scanning. In the following section, we present how accurate our methods are in recognizing the connection between charge density and electrostatic potential in homology models.

Finally, Chapter 7 summarizes the arguments of the dissertation and discusses future research problems related to the methods presented in this thesis.

Chapter 2

Background and Related Work

Clustering aims to organize the data into clusters since interpretation of the organized data is relatively easy in contrast with the raw data [26]. High-throughput data acquisition techniques generate raw data in large volumes every day [1][27][2]. Typically, interpreting a large volume of accumulated data for a specific application and extracting useful information with high accuracy is computationally intensive.

To cope with this challenge and balance the tradeoff between accuracy and efficiency in clustering context, several methods that exploit the local manipulation and analysis techniques will be presented in the subsequent chapters. In order to facilitate the explanation of these methods, we introduce fundamental concepts and discuss the previous work in the literature on unsupervised and semi-supervised clustering methods as well as wavelet transformation in biological applications.

2.1 **Clustering**

Clustering is the process of partitioning data into meaningful group such that each group consists of objects that are similar within themselves and dissimilar to the objects of other groups [26]. In data mining context, we consider a dataset D consisting of data points D={x_1, x_2,…, x_i,…, x_N} where x_i={a_1, a_1,…, a_j,…, a_d} and each a_i is an attribute of the data point x_i. Although the attributes may be either numerical or categorical, we will consider datasets with numerical attributes throughout this dissertation. The goal of clustering is to assign data points to a finite number of clusters C={C_1, C_2,…, C_k} such that clusters typically have the following properties [28]:

1. $C_i \neq \emptyset;\;\; i \in \{1, \dots, k\}$
2. $\bigcup_{i=1}^{k} C_i = D$
3. $C_i \cap C_j = \emptyset\;;\; i, j \in \{1, \dots, k\}\; and\; i \neq j$

A wide variety of clustering algorithms [29][30][31][32] that meet these requirements have been proposed in data mining. In this dissertation, we will focus on clustering methods based on unsupervised and semi-supervised learning.

2.1.1 Unsupervised Clustering

In unsupervised clustering, we seek to partition the datasets in which only unlabeled objects are given. No knowledge about the data is available in advance and thus unsupervised clustering algorithms thoroughly rely on mathematical and statistical methods to optimize a criterion function for the clustering [31]. In this section, we review the major unsupervised clustering approaches based on the theories behind them. To this end, we discuss the unsupervised clustering algorithms according to their clustering methodology [28], which is either partitioning or hierarchical in the context of this dissertation.

2.1.1.1 Partitioning methods

Partitioning methods usually divides the data into several groups iteratively in order to achieve a clustering objective [33][34][35]. For example, one of the widely-used clustering objectives is minimizing the distance between the objects within same cluster and maximizing the distance to the other objects residing in other clusters [36]. In data clustering tasks, searching an optimum value of an objective is clearly computationally intensive and prohibitive. Therefore, partitioning algorithms utilize heuristic algorithms in order to seek approximate solutions [31]. These partitioning algorithms typically run multiple times with different starting points and iteratively assign the set of objects into k clusters, where k is the pre-defined number of clusters.

In order to assess the validity of the clustering results, the partitioning methods evaluate the structural and statistical properties of the data via objective functions. These functions are usually evaluated at each iteration in order to optimize the final clustering. The most frequently used objective function is the squared error criterion [28]:

$$SE = \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| x_i^{(j)} - c_j \right\|^2$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is the distance measure between a data point $x_i^{(j)}$ of cluster C_j and the cluster centroid c_j.

K-means algorithm [31] is one of the simplest clustering tools, yet by far most popular, in unsupervised clustering which aims at minimizing the squared error criterion. The iteration procedure of the algorithm can be summarized as below:

1. Select *K* points in the space at random or using some heuristics. These points will be used as the centroids of initial clusters.
2. Assign each object to the cluster with the closest centroid or mean. This step is similar to the construction of Voronoi diagrams, where we focus on just discrete points rather than entire space.
3. Calculate the new means to be used as the centroids in the next iteration:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

The algorithm repeats these steps until the assignments no longer change. As you may notice, the algorithm refines the clusters in an iterative manner. The accuracy of the final clustering is thus very sensitive to the selection of initial centroids. If the initial centroids are not chosen carefully, the algorithm can get stuck in local minimal solutions which may be far away from the optimal solution. Furthermore, the stability of the clustering may be affected depending on the selection method. Therefore, many heuristic methods have been proposed to refine the quality of the start condition of the algorithm. Bradley et al. [37] suggested determining the modes of the joint probability density of the data and placing a clustering centroid at each mode. However, estimating the density in high dimensions usually gets difficult. Thus, they have used a sub-sampling approach to overcome the curse of dimensionality. Another method is to use the stochastic approach

[38] to adjust the initial centroids. To this end, a probabilistic error function integrated into the formula that is used to calculate the mean. The error function searches through all centroids found so far and determine the best solution based on a stochastic model.

Another shortcoming of the K-means algorithm is the sensitivity to the outlier objects, which are very far away from the rest of the objects. In contrast, K-Medoids algorithm [35] is very robust in the presence of outliers. In order to find the clusters, the algorithm determines a representative object for each cluster, instead of using the mean of data points:

$$c_k = min_c \left\{ \forall c \in C_k : \sum_{i=1}^{|C_k|} \left\| x_i^{(k)} - c \right\|^2 \right\}$$

Once the medoids are selected, all other objects are assigned to the cluster that has the closest medoids, likewise K-means.

K-Medoids algorithms are usually examined within a graph-theoretic framework due to the nature of how the clustering is done [30]. Thus, we compare these algorithms using the graph concepts. PAM (Partitioning Around Medoids) [39] algorithm developed by Kaufmann and Rousseeuw searches the medoids in the graph that minimizes the objection function. At each step, all nodes in the graph are examined. It is obvious that PAM is inefficient for large dataset and large values of *K*, number of clusters. Therefore, Kaufmann et al. designed CLARA (Clustering Large Applications) [39] which draws a sample of the dataset, applies the PAM on the sample to find the medoids of the sample, and then uses these medoids to approximate the medoids of the entire data set. The problem here is that the minimum medoids for the entire dataset may not be included in the sample and thus the performance of the clustering may be suffer dramatically. CLARANS (Clustering Large Application based on Randomized Search) [30] mitigates this deficiency. It does not check every neighbor to optimize the medoids. It uses the original graph but restricts maximum number of neighbors to be searched while finding the new medoids. Thus, the main difference between CLARA and CLARANS is that CLARA algorithm draws a sample of nodes while CLARANS draws a sample neighbors. CLARANS algorithm provides better quality clustering and requires a very small number of searches compared to the other methods.

*2.1.1.2 **Hierarchical methods***

Hierarchical clustering searches for a cluster hierarchy among the objects and constructs a tree structure known as a dendrogram. In the dendrogram, the root node represents the whole data set and each leaf node represents the individual data objects. Intermediate nodes describe the relationship between the objects based on their proximity. This structure allows us exploring the data at different levels of granularity. The final clustering can be obtained by cutting the tree at different levels using a cut-off criterion.

Hierarchical clustering methods are principally categorized into agglomerative (bottom-up) and divisive (top down) groups. In agglomerative approach, each data point is initially regarded as a cluster and these points are recursively merged to obtain final clusters. The latter approach starts with one cluster that contains all data points and recursively splits the most appropriate clusters. The both approaches continue forming clusters until a termination criterion is reached. Once an agglomerative method merges two clusters, the objects in the new cluster will always be in one cluster. Once a divisive method separates a cluster, the objects in the new clusters will never be grouped again. In this dissertation, we will utilize the agglomerative approach.

While merging two clusters or splitting one cluster, the hierarchical clustering algorithms need to generalize the distances between individual objects to the distances between clusters. Linkage metrics determines the distance between clusters as a function of pairwise distances between objects. Most widely used linkage metrics are complete-link, single-link and average-link:

$$link_{complete}(C_i, C_j) = max_{d(x,y)}\left\{d(x,y) = \|x - y\|^2 : x \in C_i, y \in C_j\right\}$$

$$link_{single}(C_i, C_j) = min_{d(x,y)}\left\{d(x,y) = \|x - y\|^2 : x \in C_i, y \in C_j\right\}$$

$$link_{average}(C_i, C_j) = \frac{1}{|C_i||C_j|}\sum_{x \in C_i}\sum_{y \in C_j}\|x - y\|^2$$

Like any other clustering method, hierarchical clustering needs to adapt itself in order to handle very large datasets. In recent years, many new hierarchical clustering techniques with great performance enhancements have been proposed. BIRCH [40] is the one of the most important development in hierarchical clustering. The algorithm introduces a new data structure called clustering feature tree in order to deal with large data sets and maintain robustness in the presence of outliers. Feature tree stores the summaries of the original data and the hierarchical clustering is applied to the summaries to build the global clustering tree. Additionally, Guha et al. proposed CURE algorithm [32] which can identify the clusters with non-spherical shapes. In the algorithm, each cluster is represented by a fixed number of points. These points are selected in a way that they would be well scattered inside the cluster. Selecting well-scattered points usually make the hierarchical clustering algorithms vulnerable to the outliers. In order to solve the outlier problem, CURE shrinks the points towards the mean of the cluster by a fraction α. The algorithm captures the arbitrary-shaped clusters using the representative points. It uses only these representative points to calculate the distances between clusters, which increases the efficiency of the agglomerative clustering. In order to handle very large datasets, CURE algorithm applies random sampling in advance to the actual clustering.

2.1.2 Semi-Supervised Clustering

Unsupervised clustering algorithms use only similarity information, which is usually given in the form of a distance function or a distance matrix, in order to perform the clustering. In many cases, partial background knowledge about the data is available. Such a priori knowledge can be utilized to provide a limited supervision on the clustering process. Background knowledge is usually given as instance-level constraints or class labels on some objects. In this dissertation, we will be considering the model where supervision is provided in the form of must-link and cannot-link constraints:

a. *Must-link* : Two objects should be assigned to the same cluster

b. *Cannot- link* : Two objects should be assigned to different clusters

Semi-supervised clustering methods employ the constraints in two ways. First, they modify the objective function to include the satisfaction of the constraint. For instance, COP-KMeans [41] integrates must-link and cannot-link constraints in the objection function so that it checks whether any constraint is violated after assignment of objects at each step. This approach performs hard constrained clustering and computes the transitive closure of the constraints, thus suffering greatly from noisy constraints sensitivity. The objective function of this method was re-expressed by Bilenko et al. [42] to reduce the effect of hard constraint approach as follows:

$$\lambda_{pckmeans} = \sum_{x_i \epsilon D} \left\| x_i^{(k)} - c_k \right\|^2 + \sum_{(x_i, x_j) \epsilon M} w_{ij} \mathbb{I}\left[l_i \neq l_j\right] + \sum_{(x_i, x_j) \epsilon C} \overline{w}_{ij} \mathbb{I}\left[l_i = l_j\right]$$

Second approach for applying the constraints in semi-supervised clustering is to train a distance metric to satisfy the constraints. Xing et al. [20] proposed a distance metric learning algorithm which places a distance metric over the input space with the intention of assigning small distances between similar pairs. This approach aims to satisfy the maximum number of constraints by specifying different weights for different axes using the following formula:

$$\omega(A) = \sum_{(x_i, x_j) \epsilon M} \left\| x_i - x_j \right\|_A^2 - log\left(\sum_{(x_i, x_j) \epsilon C} \left\| x_i - x_j \right\|_A \right)$$

The RCA algorithm [43] learns a Mahalanobis distance metric by using only must-link constraints. However, both approaches find one global metric which must be applied all clusters in the same way.

MPCK-KMeans [42] integrates the strengths of both metric-based and constraint-based approaches in a principal manner:

$$\lambda_{combined} = \sum_{x_i \epsilon D} \left(\|x_i - c_k\|_A^2 - \log(\det(A_k)) \right) + \sum_{(x_i, x_j) \epsilon M} w_{ij} \mathbb{I}\left[l_i \neq l_j\right] + \sum_{(x_i, x_j) \epsilon C} \overline{w}_{ij} \mathbb{I}\left[l_i = l_j\right]$$

The algorithm learns individual distance-metrics for each cluster by utilizing both unlabeled data and constraints. This allows different clusters to define their own space and have arbitrary boundaries.

HMRF-KMeans [44] is a probabilistic framework based on Hidden Markov Random Fields. Basu et al. have taken advantage of the available constraints in several ways. First, they estimate initial centroids using the constraints as the initialization step is crucial to accuracy of the KMeans algorithm. The model integrates constraint-based and distance-based approaches to maximize the joint likelihood of data and constraints while penalizing violated constraints. One weakness of the method is that it is applicable only to vector data, like all of the other mentioned so far.

Kulis et al. [45] extended HMRF-KMeans to a kernel-based clustering with constraints framework which can handle both vector-based and graph-based data. For this purpose, they have established connection between unweighted Kernel-KMeans, HMRF-KMeans and penalties for violated constraints. The proposed method, SS-Kernel-KMeans, optimizes the kernel by preprocessing the similarity matrix with must-link and cannot-link constraints. It applies the penalty or rewarding only to the constraint edges in affinity matrix. Kernel methods are sensitive to the manual selection of the kernel's parameters. Yan and Domeniconi [15] proposed an adaptive method that estimates the optimal parameters.

2.1.3 Graph Clustering

Graph $G(V,E)$ is a data structure that contains a collection of vertices and a collection of edges connecting pairs of vertices. Graph clustering is the task of partitioning the vertices of a graph into clusters using the edge information [46]. Graph clustering is an important and well-studied problem in many applications varying from image processing [47] to circuit design [48]. The spectral methods [49] have been widely used in many graph clustering algorithms [50]. These methods involve the computation of eigenvectors with the smallest Eigen values on Laplacian matrix, which can be computationally very expensive. TribeMCL[51] exploits the Markov model on connection graphs of proteins in order to cluster the protein sequences into families. The algorithm follows the idea that the random walks on a graph will infrequently go from one cluster to another based on the transitional probabilities in graphs. The transition operator used aims to strengthen intra-cluster flow and weaken inter-cluster flow. Satuluri et al. [52] applies the same stochastic flow idea in community discovery problem. Dhillon et al. extended the SS-Kernel-KMeans to multilevel GraClus algorithm [53] that generalizes the graph clustering objectives using trace maximization. In addition to obtaining high quality clusters for graph domain like these methods, MAPClus elucidates the graph clustering from a different perspective, and utilizes it as a tool for bending the space that wraps the vector data. A common denominator of the graph clustering algorithms is that they all need some multilevel partitioning schema for optimization while dealing with large graphs.

2.2 **Protein Similarity Search and Wavelet Transformation**

The protein engineering analyzes the similarity among proteins by comparing their physicochemical characteristics such as electrostatic potentials, hydropathy, charge density, sequence and tertiary structures. Protein similarity analysis is of paramount importance in many fields associated with the proteins. For instance, drug design recognizes the functionality of unknown proteins by comparing them against proteins whose functionality is well known. The idea is that the molecular structures with similar physicochemical properties tend to exhibit similar biological activity.

Sael et al. computes the similarity of proteins by using 3D Zernike descriptors, which represent a protein structure as a series of 3D functions in compact form [54]. The method serves as a index that allows fast retrieval of protein structures. Ying Zhou et al. [55] follow the same idea with using a spatial distribution function for backbone structure. The algorithm computes the distance between the Cα atoms using 3D coordinates and uses a sampling schema to extract the features. The features can assist the protein similarity searches in protein databank when sequence similarity is not enough. Daras et al. [56] have proposed another 3D shape-based approach for the efficient search of proteins which relies on the geometric 3D appearance of the proteins. After the translation and scaling of the protein, they decompose the 3D structures into planes which are then used to produce descriptor vectors. The advantage of these vectors is that even if the molecule rotates, these vectors remain the same. In these methods, only the global surface shape representation of the protein is taken into consideration while neglecting physicochemical characteristics.

Significant developments achieved in wavelet transformation recently have increased the interest in analyzing the physicochemical properties from a signal processing perspective [57]. Wavelet transformation allows us to analyze different scale information, which refers to the spectral components, of biological data. Wavelet transformation is defined as [58]:

$$W_f(\tau, s) = \frac{1}{\sqrt{s}} \int f(t)\psi(\frac{t-\tau}{s})\,dt$$

where $f(t)$ represents the biological function or data and $\psi(\frac{t-\tau}{s})$ is the wavelet function to analyze the spectral component at scale s and translation τ. By substituting s and τ in the function with different values, several spectral components are extracted out of the data and used to compare the biological structures [59]. This process is called multi-resolution analysis due to fact that each spectral component represents the same biological data but at different resolutions.

Wavelet transformation is widely used in bioinformatics and chemometrics to discriminate the molecular structures [60][61][62][63][64]. These methods attempt to understand the functionality or tertiary structure of the proteins using their sequence information only. In the analysis, they first substitute the amino acids in the sequence with a numeric value which corresponds to the physicochemical property of the amino acid. Then, wavelet transformation is applied to the vector structure to compute the corresponding wavelet coefficients at different scales. Finally, these coefficients are used in the structure analysis. In this dissertation, we will analyze the actual three dimensional distributions of physicochemical properties rather than using only sequences.

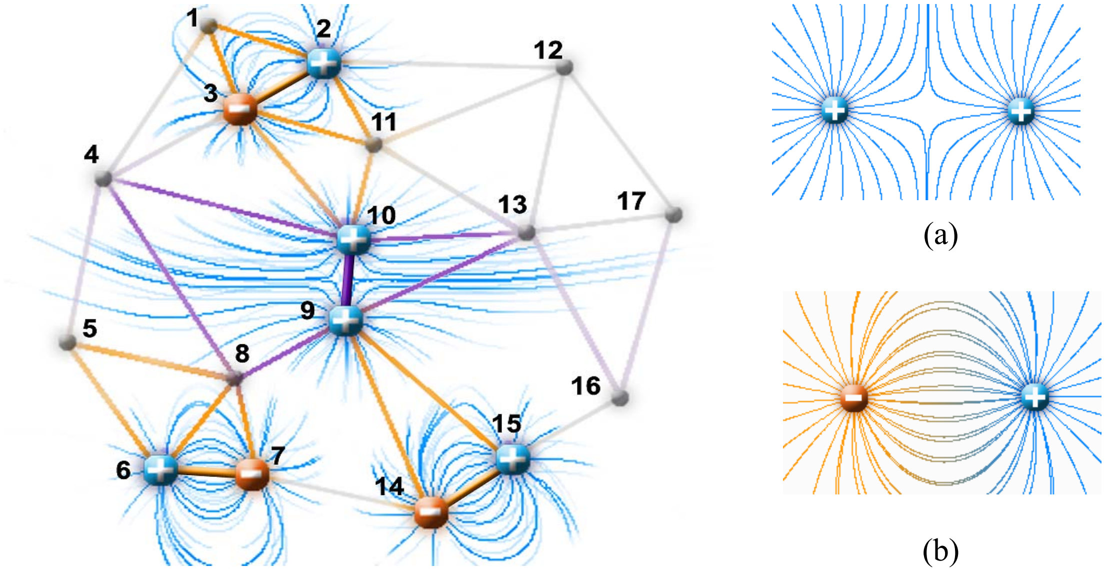
Chapter 3

Magnetically Affected Paths

In this section, we introduce the concept of Magnetically Affected Paths (MAP), a novel approach to applying user constraints in semi-supervised clustering, and describe the basic idea behind the approach in an intuitive way [13]. We assume that a function defining the distance between any two points and a set of user defined constraints are given. The main goal is to calculate more accurate distances between data instances using the supervision provided by the constraints. We make no assumptions about the distance measure, but we assume that the constraints are on pairs of points, either must-link or cannot-link, which indicate that a pair of points should be or should not be put in the same cluster, respectively. This method uses the user-defined constraints to stretch the space around the objects and helps the distance metric calculate more accurate distances.

3.1 **MAP Concept**

The objective of the MAP is to increase the weight of the paths connecting the objects that are in the neighborhood of a cannot-link constraint, and to decrease the weight of the paths connecting the objects that are in the neighborhood of a must-link constraint. We realize the idea via the interesting analogy between electromagnetic field theory and graphs. We focus on the graph representation of a given dataset and assume that this graph has the characteristics of an electromagnetic field. In physics, an electric field is the property of the space in the vicinity of electric charges or in the presence of a time-varying magnetic field. The charges produce an electric field in space. This electric field exerts a force on other charged objects [22]. Opposite-charged objects induce attractive properties, whereas like-charged objects induce repulsive properties. To



(a)

(b)

Figure 3-1. Simulation of an EMF in a graph. (a) Like charges induces repulsive properties while (b) opposite charges induces attractive properties of other objects.

simulate these characteristics, we add charges to the nodes that take part in a pair-wise constraint. For must-link constraints, we add opposite charges to the node pair. The generated magnetic field decreases the weight of the affected edges. Cannot-link constraints (like charges) increase the weights of affected edges.

The situation is illustrated in Figure 3.1, where must-link constraints are {*e(2,3), e(6,7), e(14,15)*} and the cannot-link constraints are {*e(9,10)*}. We explore imaginary magnetic fields surrounding the pair of charged nodes. We check the nearby edges in the graph and identify the graph edges that are influenced by the magnetic field. Then, we reduce the weight of the affected edge or escalate it according to the constraint type. The magnitude of readjustment depends on the distance of the edge in regard to the magnetic field and its alignment in the field. The magnitude is defined in terms of following definitions in the latter section:

- ***Constraint axis:*** The shortest path between the nodes of a constraint
- ***Reduction ratio { rRatio(u,v)*** *}:* The decrement amount in edge weight *w*(*u,v*) due to a must-link constraint.
- ***Escalation ratio { eRatio(u,v) }:*** The increment amount in edge weight *w*(*u,v*) due to a cannot-link constraint.
- ***Vertical distance { vd(u,v) }:*** The average distance of an edge *e*(*u,v*) to the constraint axis.
- ***Horizontal distance { hd(u,v)*** *}:* The distance of an edge *e*(*u,v*) to the mid-point of the *constraint axis*.

The effect of an electromagnetic field decreases as we get further away from the constraint axis (vertical distance). For must-link constraints, the horizontal distance has no effect on the reduction ratio. Cannot-link constraints utilize the horizontal distance of an edge to determine its probability of being in the separation region of two clusters. Intuitively, *the closer a regular edge e(u,v) is to the mid-point of a negative edge constraint, the higher the probability of it being an inter-cluster edge.* Based on this intuition, we apply the highest penalty to the edges overlapping with the mid-point of a constraint axis. The penalty is reduced as we go further away from the mid-point.

In summary, MAP increases or decreases the weights of regular edges based on a probabilistic approach. Even though it classifies some of the edges incorrectly, overall re-adjustment of edge weights defines better distances in the graph domain.

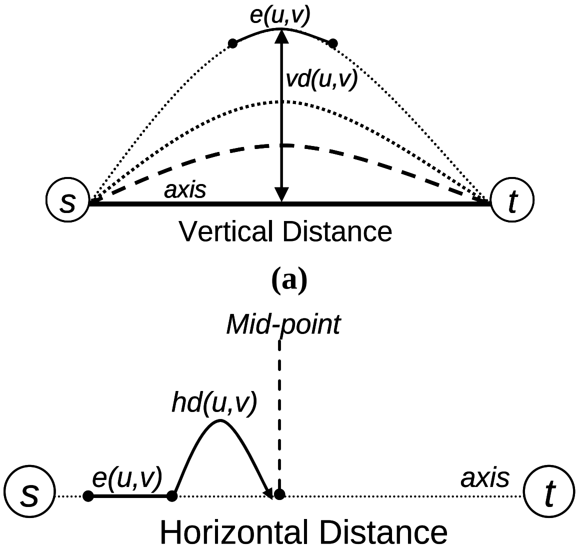


Figure 3-2. Horizontal and vertical distances relative to the constraint edge *c(s,t)*

3.2 **Weight Adjustment Process**

We describe the MAP-based weight adjustment algorithm in this section. As mentioned in previous section, the algorithm applies the MAP concept to the graph in order to increase or decrease the edge weights. For each constraint *c(s,t)* , we extract a list *L* of edges *e(u,v)* $\in$*E* which are affected by the constraint. Affected edges are recognized according to the following definition:

DEFINITION: *If a given edge e(u,v) is in between nodes s and t of constraint c(s,t) and is not orthogonal to the constraint axis, then it is affected by constraint c(s,t).*

In our model, the orthogonality and betweenness is defined based on hop-count distance to the constraint nodes *s* and *t*. We run two breadth-first search algorithms starting at node *s* and *t* separately and for each node we store entries *hc(u,s)* and *hc(u,t),* the hop-count distance to *s* and to the *t* respectively. For a given edge *e(u,v),* we check the hop-count entries of *u* and *v* to see whether the edge is affected by a constraint or not. Orthogonality and betweenness is described as the inverse behavior on *hc(u,s), hc(u,t)* and *hc(v,s), hc(v,t)* values for nodes *u* and *v*. In other words, if *hc(u,s)>hc(v,s)* and *hc(u,t)<hc(v,t)* or vice versa, then the edge is orthogonal to the constraint axis and between nodes *s* and *t*.

Once we identify the affected edges, we compute the escalation ratio for the cannot-link constraints or reduction ratio for the must-link constraints. In line with the main idea, we expect the effective escalation or reduction ratio on an affected edge to decrease as we get away from the constraint axis (vertical distance). Further for a cannot-link constraint, we expect an inversely proportional weight increase in regard to the distance of the edge to the mid-point of the constraint axis, as explained earlier. To express this, we use a method similar to the validation process. Instead of hop counts, we find the shortest path distance to all edges starting at node *s* and *t*. Shortest path *dist(u,v)* is the sum of the weights of all edges that compose the shortest path.

We compute the reduction ratio of *e(u,v)* for must-link constraints as follows:

$$rRatio(u,v) = norm(\frac{q_r}{r})$$

Here, q_r is the weight for the must-link constraint and *r*=(*dist(u,s)* + *dist(u,t)* + *dist(v,s)* + *dist(v,t)*) / *w(s,t)* which is the dispersion ratio from the constraint *c(s,t). norm()* is the normalizing function. To prevent extremely low values, normalization function maps the output to a higher interval.

Similarly, we can write the following equation to compute the escalation ratio due to a cannot-link constraint:

$$eRatio(u,v) = norm(\frac{q_e}{r \cdot \Delta})$$

where q_e is the weight of the cannot-link constraint, Δ=(|*dist(u,s)*-*dist(u,t)*|+|*dist(v,s)*-*dist(v,t)*|)/*w(s,t)*+*c,* which is the average distance approximation function to the mid-point of the constraint axis to reflect the effect of horizontal distance as seen in Figure 3.2. The value of |*dist(u,s)*-*dist(u,t)*| becomes zero if node u has equal distances to the s and t. We add a constant value *c*=1 to the Δ so that in this case, no penalty is applied to *eRatio(u,v).* If Δ increases, the effect of *eRatio(u,v)* reduces gradually.

After applying all constraints, we approximate the overall ratio of edge *e(u,v)* as follows:

$$tRatio(u,v) = \frac{\sum_{i=1}^{|C|} eRatio_i(u,v)}{|C|} - \frac{\sum_{j=1}^{|M|} rRatio_j(u,v)}{|M|}$$

In the last step, we adjust the edge weight as follows:

$$w_{new}(u,v) = w(u,v) \cdot \alpha^{tRatio(u,v)}$$

Empirically, we have observed that $1<\alpha<2$ is a good interval for the adjustment of the edge weights. It is obvious that if cannot-link constraints are dominant upon the must-link ones, then the edge weight increases. Otherwise, the base variable acts like a denominator. It is important to note that vertical and horizontal distances are used just to compute the adjustment ration and are not used in any way in the clustering process.

Chapter 4

MAPClus Clustering Framework

In this section, we describe our clustering framework that unifies MAP and clustering algorithms. The algorithm uses three steps to apply the MAP concept to a given dataset and perform clustering:

1) Graph Construction: If given a vector-based dataset, it is converted into a graph by connecting k-nearest-neighbors. Must-link and cannot-link constraints are addressed as same- or opposite-charged nodes, respectively.

2) Weight Adjustment: Identifies the edges that are affected by the given constraints and adjusts the edge weights accordingly.

3) Clustering: Runs an appropriate clustering algorithm to partition the adjusted graph.

4.1 Introduction

Clustering aims at providing useful information by organizing data into groups (referred to as clusters). The use of labeled data is often critical to the success of the clustering process and the evaluation of the clustering accuracy. Consequently, learning approaches which use both labeled and unlabeled data have recently attracted the interest of researchers [20] [42] [44] [45] [65]. These approaches incorporate user knowledge in the clustering technique, thus improving the clustering result. Typically the method used is to allow the user to specify constraints on pairs of objects, either must-link (two objects must be in the same cluster) or cannot-link (two objects must be in different clusters), and produce a clustering that satisfies these constraints as much as possible.

In this chapter, we present a novel way of applying user constraints for clustering, which is inspired by the Electromagnetic Field Theory used in physics. Our approach transforms vector data into graph data by finding and linking k-nearest neighbors for each instance in the dataset. If given graph data as an input, no transformation is needed. Must-link and cannot-link constraints are then expressed naturally as magnetic fields between the nodes that are involved in the constraint. These fields impact edge weights based on the alignment of each edge compared to the magnetic field and its distance to the constraint axis. Using graph representation yields an advantage in that we can adjust the edge weights without any limitations, in contrast to Euclidean space, where pair-wise distances need to comply with the triangle inequality. We exploit this liberty through a probabilistic model based on the nature of the constraint edges. Once we adjust the weights, we can apply any clustering algorithm compatible with graphs. In our study, we use the K-Medoids algorithm [39], since it is less sensitive to the outliers. For the distance metric to be used by the clustering algorithm, we propose *k* simple-and-distinct shortest paths. *k-SD* shortest paths leads to more accurate clustering with small number of constraints.

An important challenge is dealing with large graphs. We use the most widely used method for graph clustering, METIS [66], to partition the large graphs into equal-sized sub graphs to perform our operations. This partitioning strategy preserves 95+% quality and enables the method to scale almost linearly with the number of subgraphs. However, this approach losses its efficacy on datasets with over ten thousands instances. In order to work with very large datasets, we improve our framework with a multilevel partitioning approach which was proposed by Karypis and Kumar [67]. The multilevel approach first coarsens the given graph level by level until only a small number of nodes are left. Then, it performs the initial partitioning on this small graph. Finally, the initial partitioning is projected back to the original graph by refining the graph level by level.

4.2 **MAPClus Framework**

4.2.1 Graph Construction Phase

If the input is not graph, but a vector of data points *D*, our goal is to build a graph reflecting the data with minimal loss of information. We list *k* nearest neighbors L_i for each object $x_i \in D$, according to their Euclidean distance, and add an edge between x_i and each $v \in L_i$. We assign the Euclidean distance $||x_i-v||^2$ as the edge weight of $e(x_i,v)$. k can be easily estimated from the graph size. Experimental results show that k should be proportional to the dataset size $|D|$ for better accuracy in small data sets. However, this characteristic does not hold for large datasets and typically ten neighbors for each node suggests reasonable results in the ultimate clustering.

We take all node pairs (s_i,t_i) involved in some constraint and charge s_i and t_i so that the force between them is equal to $||s_i-t_i||^2$. Remember, opposite charged pair of nodes create an attractive force (must-link constraint), whereas same charged pair of nodes create a repulsive force (cannot-link constraint).

Even if we set k to a appropriate value, disconnected components might still exist. Thus, we identify all disconnected subgraphs, explore k nearest neighbors at subgraph level, and add an edge between the closest points connecting the disconnected components. This approach is similar to [45].

4.2.2 Weight Readjustment Phase

The weight readjustment phase applies the MAP concept, which was described in the previous chapter, to the graph in order to increase or decrease the weight of edges. Because the affected edges are identified according to their hop distances to the constraint axis, we start with running two breath-first search algorithms originated at two ends of each constraint. Once breath-first search completes, we can easily determine the affected edges using the betweenness and orthogonality criteria. Next, we compute the shortest path distances from constraints to the affected edges and measure the effect of the constraint on each edge according to the formulas given in the previous chapter. After all constraints are processed, we proceed to calculating the k-shortest path distances between each pair of nodes, which will be input into clustering algorithm as distance matrix. The weight adjustment algorithm is summarized in Table 4.1.

```
Algorithm: Weight_Adjustment_Algorithm

    Input: G(V,E): graph with constraints
    Output: G' (V,E'): graph with adjusted edge weights
            P:  proximity matrix

1. For each constraint c(s,t)
    a. Run breath-first search algorithm starting at node s and t;
       and record hop-counts for each node $v_i \in V$
    b. Run single shortest path algorithm starting at node s and t;
       and record shortest path distances for each node $v_i \in V$
    c. Identify affected edges using hop-counts and put them into list L
    d. Compute escalation/reduction ratio for each affected edge $e(u,v) \in L$
2. For each edge $e(u,v) \in E$
    a. Calculate overall ratio using following formula
       $tRatio(u,v) = \frac{\sum_{i=1}^{|C|} eRatio_i(u,v)}{|C|} - \frac{\sum_{j=1}^{|M|} rRatio_j(u,v)}{|M|}$
    b. Apply overall ratio to the edge weight $w_{new}(u,v) = w(u,v) \cdot \alpha^{tRatio(u,v)}$
3. For each node pair (v, u), calculate the distance $D(u,v) = \left( \sum_{q=1}^{k} \frac{1}{dist_q(u,v)} \right)^{-1}$ .
```

Table 4-1. Pseudo-code of Weight Adjustment Algorithm

4.2.3 K-Shortest and Distinct Paths (k-SDP) Distance

After adjusting all the edge weights, the need for a distance metric which is capable of exploiting the final affinity matrix emerges. Even though single shortest path is a widely-used metric in graphs, we can define more accurate distances between node pairs using k-shortest paths distance. Remember that k was the number of neighbors used to transform vector data into a graph representation. For vector data, we use the same k for the number of shortest paths since intuitively this is the maximum out-link number of an edge and looking for more than k shortest path that are distinct (have no common edge) would be pointless. For graph-based data, k is the average degree of the graph. Our experiments support using multiple shortest paths as a distance metric works in practice better than single shortest path. Involving more paths in the calculation of the distance involves also more constraints, which due to the homophily phenomenon, as mentioned, yields better accuracy even for a small number of constraints. The shortest path distance is expressed as:

$$dist_{all}(u,v) = \left(\sum_{i}^{k} \frac{1}{dist_i(u,v)}\right)^{-1}$$

where $dist_i(u,v)$ is the weight of the i^{th} path between node u and v such that $dist_i(u,v) < dist_j(u,v)\ if\ \forall i,j \in [1,\dots,k]\ and\ i < j.$

A naive approach for k-shortest paths is using the Dijkstra's algorithm to discover k most significant paths one by one, which has a time complexity of O(k.$|V|^2$.($|E|$+

$|V|.\log|V|$)). Obviously, finding the pair-wise paths with naive method would be cumbersome for the efficiency.

4.2.4 Optimizations for *k*-SDP distance

4.2.4.1 Extending Dijkstra's Algorithms for k-Shortest and Distinct Paths

We have optimized our implementation in several ways for maximum efficiency. First, we have started with restricting the definition for the *k*-shortest path. Even though there is more than one *k*-shortest paths definitions in the literature [68], we focus on *k* simple and distinct shortest paths in which no loop is allowed, i.e. all vertices on a path are distinct and no two paths share the same edge for a given source and destination pair. Exploiting this definition, we extend the Dijkstra algorithm for single-shortest path to k-shortest paths at a reasonable cost and refer to it as *k*-SDP algorithm. The new algorithm is asymptotically only *k* times slower than the Dijkstra algorithm for one path.

The algorithm works as follows: For each node, we define *k* entries to handle each l^{th} path passing through the node where $1 \le l \le k$. We initialize all entries to ∞ except the entries of source node *s* and nodes adjacent to the source. We set all *s* entries to 0. We assign monotonically increasing path labels *i* to each node *u* adjacent to *s* ensuring that each path is rooted at a different edge outgoing *s*. We update the l^{th} entry of each node u to edge weight $w(s,u)$ and the parents of l^{th} entry of the nodes to source node *s* where *l* is the path label pertaining to each adjacent node. Then, we initialize a minimum priority queue *Q* that contains all path entries. The rest of the algorithm works very similar to

Dijkstra’s shortest path algorithm except for a few additional restrictions in the relaxation routine.

When we remove the minimum entry from *Q*, we lock the parent of the entry in order to prevent more updates from this parent for other path entries, using bitmaps with

```
Algorithm: K-SD_Shortest_Path_Algorithm
    Input: G'(V,E): graph with adjusted edge weights
               s: source node
               k: number of shortest path
    Output: P: Distance matrix
1. Assign $k$ path entries for each node such as $p_i(s,u)$
2. Initialize $p_i(s,u).length \leftarrow +\infty$ for each node $u \neq s$
3. For each node $u$ adjacent to $s$
       a. Assign a monotonically increasing path id $i$ to $u$
       b. Set $p_i(s,u).dist \leftarrow w(s,u)$ and $parent_i(u) \leftarrow s$
4. Let a min priority queue $Q$ contain all path entries
5. while $Q$ is not empty do
       a. Extract path entry $pe \leftarrow Q$.removeMin()
       b. Let $i \leftarrow pe.pathID$ and $v \leftarrow pe.node$
       c. Lock $parent_i(v)$ to prevent updates for $v$
       d. for each node $u$ adjacent to $v$
               i. if $v$ is locked for $u$, then continue
              ii. if $p_i(s,v).length+w(v,u)<p_i(s,u).length$, then
                      $p_i(s,u).length \leftarrow p_i(s,v).length+w(v,u)$
                      $parent_i(u) \leftarrow v$
                      Update the value of $p_i(s,u)$ in queue $Q$
```

Table 4-2. Pseudo-code of K-SD Shortest Path Algorithm

each bit assigned to one neighbor node. In the relaxation procedure, first we check whether the current node is locked for destination node. If it is not, the algorithm allows it to relax the destination from current node. Otherwise, we simply proceed to the next available neighbor node. Another rule is that l^{th} entry of path entries at a node can be updated only by the l^{th} path entry of the parent node. This limitation is necessary in order to force all paths to follow a different set of edges to the destination so that we can trace the path lengths in the case that we want one specific path with path id of *l*. We repeat the relaxation process until all path entries at each node are updated and no more entries remain in the queue. The algorithm is outlined in Table 4.2.

THEOREM 1: *K-SD Shortest* Path *algorithm identifies k simple and distinct shortest paths from a given source node to all other nodes in* $O(k^2 * |V| * log|V|)$ *time.*

Proof: We assume the worst case scenario where we find all shortest path available, giving us $O(k*|V|)$ entries in the queue. For each entry, we check all neighbor nodes and based on *k*-shortest path definition and our graph construction algorithm, we may have at most *k* neighbors. We may update the keys of all neighbors at each relaxation step, requiring $O(k*log|V|)$ time.

4.2.4.2 Partitioning Approach for Extracting Distance Matrix

Even with the optimization in previous section, computing all pairs KSD-shortest paths still takes $O(k^{2} \cdot |V|^{2} \cdot log|V|)$, which is quite inefficient for large datasets. Many state-of-art methods deal with this problem by using a multilevel approach. Our strategy is to partition the graph into smaller pieces. Then, we compute local distance matrices and correlate these local resolutions via hubs to obtain the global solution.

Here, we have pursued the same idea in [69]: We partition the graph into *p* equally sized sub graphs using METIS with the Kernighan-Lin objective and find local K-SD shortest path distances for each partition. Let D_i be the distance matrix for partition *i*. The main problem is how to establish a mutual relation between these partitions to estimate global distances. The solution lies in hub concept discussed in [70]. The vertices that reside on the cut and bridge different clusters are considered as hubs. We identify the high quality hubs, the ones with high degree in total and balanced neighborhood to different partitions. Unlike the hubs in SCAN, we assume hubs are special vertices belonging to all clusters which it has neighborhood. Thus, we put them into the partitions as member. If needed, we add new edges to hubs to maintain the k-neighborhood. In Figure 4.1, shown hubs are an element of both partitions.

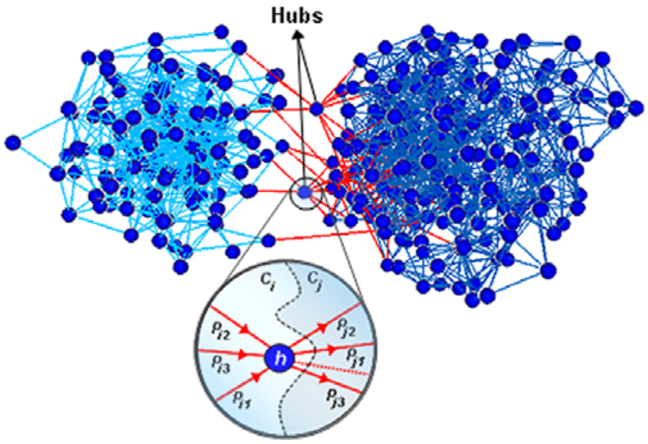


Figure 4-1. The Ionosphere graph with two partitions and hubs connecting them and the correlation of same paths for clusters C_i and C_j

Let H be the set of all hubs in the graph, H_i be the set of hubs in partition i, and m be the size of H. Interestingly, $m<<n$, which makes the fast correlation possible. We already have the distances between all H_i elements in matrix D_i. Next step is finding the pair-wise distances for the elements in H so that we can compute the distance of two elements belonging to two different partitions in an efficient way. We create an edge between all hub pairs (h_a,h_b) in H_i and assign $D_i(h_a,h_b)$ as the edge weight where $\forall h_a, h_b \in H_i$. Since H_i elements are also elements of another partition by definition, we get a fully connected graph of just hubs. By running Floyd-Warshall, we get a distance matrix S of all hubs in the graph. As we always maintain k-shortest paths for a hub in both partitions, the matrix S is a good approximation of KSD-shortest path distances and will function as a router between the nodes of different clusters.

At this point, we have performed all the preparation for extracting the global distances in an efficient way. Let s be the source node, t be the destination node, and $S(h_i,h_j)$ be the distance between hubs h_i and h_j. Then, the KSD-shortest path problem can be expressed as the following optimization problem:

$$D(s.t) = min\{D_i(s,h_a) + S(h_a,h_b) + D_j(h_b,t)\}$$
$$\text{where } \forall h_a \in H_i \text{ and} \forall h_b \in H_j\}$$

We have the distances between s and all H_i elements in matrix D_i. We compute the distances from s to the other hubs in set $(H\text{-}H_i)$ through H_i using the S matrix. Each destination node checks only the hubs of its own partition in H_j to relax the shortest path distance to node s. It takes $O(|H_i|.(|H|-|H_i|)+|V-V_i|.|H_j|)$ to find k-shortest paths from s to all other nodes. Note that, $|V_i| \approx |V|/p$ and $|H_i| \approx |H|/p$. MAPClus implementation with partitioning approach *has* an overall time complexity of $O(\frac{|V|^2k^2\log(|V|)}{p} + |H|^3 + \frac{|V|.|H|(|V|+|H|)}{p})$. If p=$|V|$, the algorithm performs like regular single shortest path algorithm.

4.2.5 Clustering Phase

The framework allows us to use any graph-compatible clustering algorithm. The success of clustering essentially depends on the compatibility of the dataset and the clustering algorithm. Thus, the choice of the algorithm must be made cautiously.

We implemented the K-Medoids algorithm, which utilizes the similarity matrix and can be applied on both vector and the graph data. Usually the initial centroids determine the final clustering, so instead of random initialization, we take advantage of the given constraints. We take the transitive closure of the must-link constraints and define groups of nodes, which have to be clustered together. At this point, each group represents a set. We merge the closest two sets until we have *K* sets remaining. Eventually, we have *K* disconnected sets formed by constraints, which we use to initialize the medoids.

The algorithm starts by assigning every point $x_i \in Ds$ to the cluster that minimizes the distance between x_i and $\boldsymbol{\mu}_k$ where $\boldsymbol{\mu}_k$ is the cluster medoid of cluster k^*. Rather than Euclidean distance, we use the distance matrix extracted in the previous step for assignment. The algorithm re-estimates medoid $\boldsymbol{\mu}_k$ using the points assigned to cluster k^*. For each point x_i, we check the total distance to all other points and we assign the point with minimum distance as the cluster medoid. Then, we repeat the steps until algorithm converges or reaches to a pre-specified number of runs. The time complexity of the clustering process is $O(t.K.N^2)$ where t is the number of iterations, K is number of clusters and N is the size of the dataset.

4.3 A Multilevel Approach

In this section, we present multi-level MAPClus framework that is based on the multilevel partitioning algorithms implemented by Karypis and Kumar [71]. The multilevel approach first coarsens the given graph level by level until only a small number of nodes are left. Then, it performs the initial partitioning on this small graph. Finally, the initial partitioning is projected back to the original graph by refining the graph level by level. Obviously, the basic structure of the multilevel partitioning is very straight-forward; however, implementing it for a specific objective, which is MAP in our case, can be quite tricky. We describe the details of each phase in order to exploit the MAP concept in a multilevel manner.

4.3.1 Coarsening Phase

Given a weighted graph $G_0=(V_0,E_0)$, the coarsening phase successively transforms G_0 into smaller graphs such that $|V_1| > |V_2| > \ldots >| V_k|$ where $|V_i|$ is the number of nodes at level *i*. A widely-used coarsening schema is combining a set of vertices into super nodes. In super node notion, we visit the each vertex at random order. For each vertex, we find the closest neighbor which is referred as candidate node for the *fold* operation. If

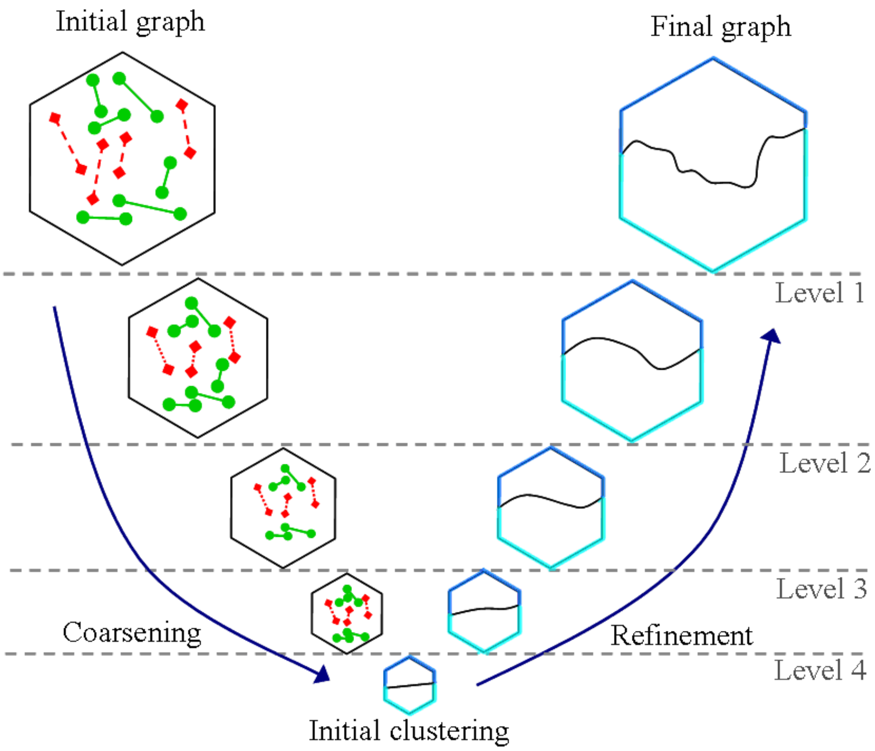


Figure 4-2. Multilevel Clustering with four levels. During the coarsening, some constraints are removed out of the graph due to the homophily criterion

there is a negative edge between current node and candidate node, then we look for the second closest and so on. We check which one, between current node and candidate node, has a greater degree. The smaller one then gets collapsed onto the larger one. The operation uses a similar approach as the union-by-rank procedure in graph theory in order to maintain the efficiency.

The main issue in the *fold* operation is how to treat the edges. So as to preserve the connectivity information, the super node typically contains the union of the edges from current and the candidate nodes. If there is more than one edge between same node

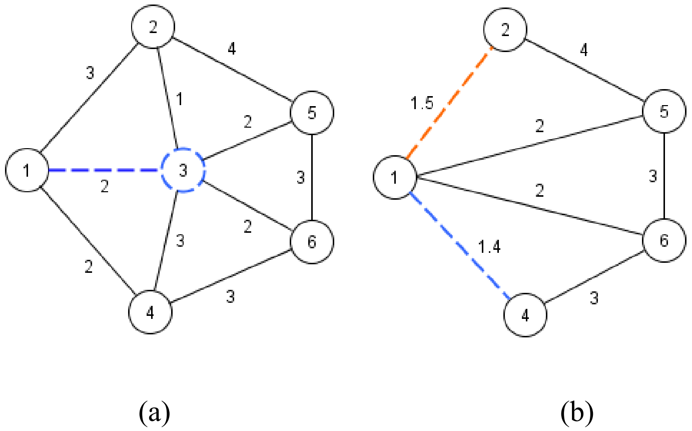


Figure 4-3. Folding operation of node v_3 onto the node v_1

pairs, these edges are merged into one and the weight of the new edge is determined as the sum of the weights of these edges. This procedure works well with methods using the Kernighan-Lin objective. However, the methods relying on a distance-based objective such as MAP benefits very little. To change the course of the situation, we need a new interpretation for the merged edges while conserving the connectivity information.

After the edge weights are adjusted in the next phase, MAP employs the *k*-SDP algorithm to compute the distance between two objects. The *k*-SDP performs a distance relaxation procedure by finding and merging *k*-shortest paths. When we collapse two nodes during *fold* operation, the path information between two nodes gets lost and this affects the *k*-SDP algorithm's performance dramatically. We perform a relaxation procedure to reduce this effect as follows: Consider the sub-graph example in Figure 7. Let v_3 be the current node to be merged, v_1 be the candidate node to which we merge and $e(v_1, v_3)$ be the collapsing edge. The merging of edge $e(v_2, v_3)$ to node v_1 fits to the problem definition since we will have two $e(v_1, v_2)$ nodes in the super node at the very end. To solve the problem, we find the path from $v_{2\ to}\ v_1$ that passes through the node v_3. Then, we relax the edge between v_1 and v_2 using the equation:

$$w_{new}(v_1, v_2) = \frac{w_{path}(v_1, v_2)^{v_3} . w(v_1, v_2)}{w_{path}(v_1, v_2)^{v_3} + w(v_1, v_2)}$$

where $w_{path}(v_1, v_2)^{v_3} = w(v_1, v_3) + w(v_2, v_3)$. This equation simulates the *k*-SDP metric in a local scope. We can use the relaxed edges directly in the initial clustering phase. In Figure 4.3, the relaxation procedure is applied to $e(v_1, v_4)$ as well while all other edges are connected to the super node v_1 without any change.

During the coarsening phase, the folding operation may eliminate the homophily criteria for some of the constraints. If a node of a constraint edge gets carried away from the original location too much, the probabilistic model no longer holds for the constraint. Before coarsening phase starts, the algorithm identifies the 2-hop neighbors for each node and they are marked as safe neighbors for *fold* operation. As long as the node collapses on any of these neighbors, we assume the homophily criteria still holds. If the *fold* operation involves neighbors out of this range, then the constraint edge is removed from the graph.

The algorithm stops coarsening phase when the average degree of the current-level graph gets larger than 2 times the average degree of the original graph. Considering each fold operation removes one node and minimum one edge out of the graph, this condition is typically more than enough for obtaining a good sub-graph.

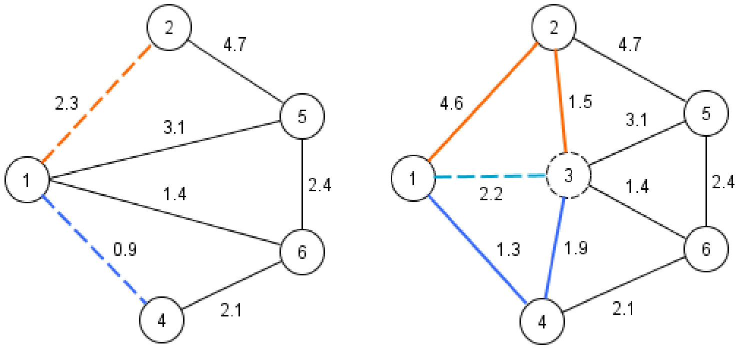
4.3.2 Initial Clustering Phase

After the coarsening phase, we have a small graph which is a projection of the original graph with limited yet enough number of constraints. The framework allows us to use any graph-compatible clustering algorithm. The success of clustering essentially depends on the compatibility of the dataset and the clustering algorithm. Thus, the choice of the algorithm must be made cautiously. In the multilevel approach, the selection of the algorithm gets more of an issue.

4.3.3 Refinement Phase

In the final phase, the initial partitioning is repeatedly projected back to the original graph. The initial clustering can be improved using refinement algorithms during the projection of G_i to G_{i-1}. Many multilevel methods implement the Kernighan-Lin refinement algorithm [71], which attempts to minimize the cut while maintaining equal-sized clusters. The Kerninghan-Lin algorithm engages a quantity called "*gain*" which is the benefit of swapping nodes between clusters relative to the objective function. Kerninghan-Lin algorithm computes the gain for all nodes in the graph. However, most of the swap operations happen along the boundary of the cut and Kerninghan-Lin wastes too much time while visiting other nodes. The boundary refinement algorithm computes the gain for only boundary vertices and performs swap operation accordingly. In our study, we employ the *k*-SDP based refinement algorithm which is also a type of boundary refinement. It aims to minimize distance-based objective without the restriction of equal-sized clusters. Our multilevel algorithm uses MAPClus algorithm to refine the clustering from previous level. Before running the algorithm, the collapsed nodes for the current level should be unfolded so that they can contribute to the refinement step.

We assume that the edges are already readjusted by the MAP algorithm in the previous level. We consider the same setting as in the fold operation. Figure 4.4a illustrates the sub-graph before the node v_1 and v_3 are unfolded. The edge weights are



(a) (b)

Figure 4-4. Unfolding operation on node v_3 after the readjustment procedure

increased or decreased in the sub-graph. The data structure in the algorithm keeps the sub-graph state before the readjustment procedure. We need to compute $tRatio_{i-1}(v_1, v_2)$, $tRatio_{i-1}$ (v_1, v_3) , $tRatio_{i-1}$ (v_2, v_3), $tRatio_{i-1}$ (v_1, v_4) , $tRatio_{i-1}$ (v_3, v_4) in order to readjust the unfolded edges. Running the readjustment algorithm just for these edges is a waste of computation. Instead, we can use the *tRatio*s of the merges edges, – $tRatio_i(v_1, v_2)$ and $tRatio_i(v_1, v_4)$ –, to estimate these values. We compute $tRatio_i(v_1, v_2)$ as

$$tRatio_{i-1}(v_1,v_2) = \log_{\alpha}\left(\frac{w_i(v_1,v_2)}{w_{i-1}(v_1,v_2)}\right)$$

We apply *tRatio*$_{i\text{-}1}$(v_1, v_2) to the edges *e*(v_1, v_2) and *e* (v_2, v_3). In a similar way, we compute *tRatio*$_i$(v_1, v_4) and apply it to the edges *e*(v_1, v_4) and *e*(v_3, v_4). Finally, we compute the average of *tRatio*$_{i\text{-}1}$(v_1, v_2) and *tRatio*$_{i\text{-}1}$(v_1, v_4) and apply it to the e(v_1, v_3). This method achieves very good estimations for the readjustment of unfolded edges while avoiding heavy constraint computations recurrently at each level. The multilevel algorithm terminates after the refinement of the original graph. Because of the optimizations such as boundary nodes and efficient readjustment, the refinement step usually converges very quickly, -just like coarsening phase. The experimental results demonstrate that this refinement algorithm projects the initial clustering to the final graph very rapidly and accurately.

4.4 **Experiments**

We have conducted experiments on two synthetic datasets and six real datasets from UCI Machine Learning Repository [72]: *Soybean, Iris, Wine, Ionosphere, Balance, Breast Cancer and Satellite.* The properties of these dataset are summarized in Table 1. *N* is the number of instances, d is the number of dimensions, and K is the number of clusters in each dataset. We have measured the clustering accuracy as:

$$Accuracy = \sum_{i>j} \frac{1\{1\{c_i = c_j\} = 1\{\hat{c}_i = \hat{c}_j\}\}}{0.5N(N-1)}$$

where 1{·} returns 1 if any pair of instances x_i and x_j are assigned correctly by the algorithm [20]. In each experimental setup, we have run the clustering algorithms for 50 times and reported the average accuracy ratios.

Must-link and cannot-link constraints are generated randomly at equal amounts and the total amount is varied proportional to the dataset size. For each run we have used the same constraint set for all clustering algorithms. For the MAPClus configuration, we used α=1.2 and increased the number of nearest neighbors and number of shortest paths proportional to the dataset size. The ratio between positive and negative edge constraint weights, q_r and q_e, are set to 1.6.

4.4.1 Synthetic Datasets

To visualize how our method works, we generated two synthetic datasets:

Gaussian: A set of 180 two-dimensional instances generated by Gaussian number generator, as shown in Figure 4.5. 120 instances in vertical and lower horizontal sets are labeled as class one. Upper horizontal set is label as class two.

ThreeCircles: Similar to *TwoCircles* data in [45], we have generated three layered circular data with 300 instances in 2 dimensions. Each circle represents one class with 100 data points in it.

We generated small amount of must-link and cannot link constraints (18 for *Gaussian* and 30 for *ThreeCircles*). Figure 4.5 shows final clustering of these datasets with high accuracy. For the circular data, Graph Construction process had a pre-clustering effect. As we have selected k-nearest neighbors for each point, there were not too many inter-cluster edges in the graph. In addition, the re-adjustment phase successfully wiped out the effects of these inter-cluster edges for the clustering phase.

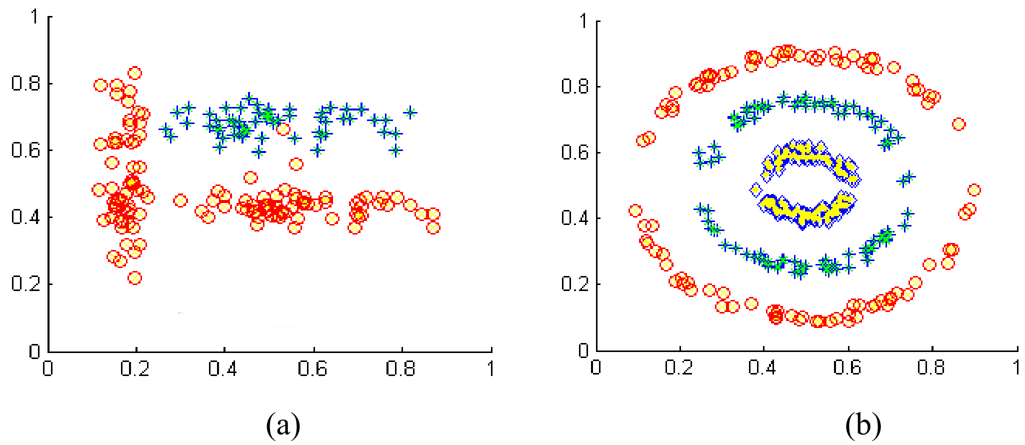


Figure 4-5. MAPCLUS clusters (a) Gaussian and (b) ThreeCircles datasets with small set of constraints

4.4.2 Choice of Tuning Parameters

4.4.2.1 Effect of Parameter k and p

We have analyzed the effect of parameters over clustering results. As increasing number of constraints and increasing number of nearest neighbors *k* both increase accuracy rate, they show different effects in the results. Second, we have varied both *k* and number of constraints in the first set of experiments. The number of nearest neighbors, *k,* is one of the essential parameters for the graph construction algorithm and it must be increased proportionally to the dataset size to ensure the best result. However, as seen in Figure 4.7, taking a value for *k* outside the optimal interval causes reduction or instability in the accuracy trend. For small values of *k*, we cannot fully take advantage of the k-shortest paths distance metric. If the value is set too high, all edges are labeled as an affected edge and consequently, false escalation or reduction occurs for too many edges.

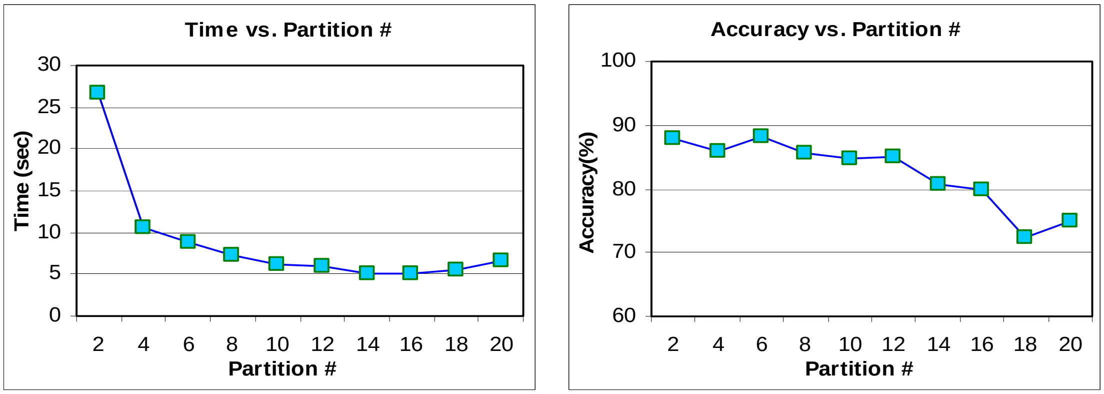


Figure 4-6. Effect of number of partitions on clustering Breast dataset in terms of (a) running time and (b) accuracy

The number of partitions has a dramatic effect upon the running time while preserving the accuracy. The algorithm runs up to 24x faster for the Breast dataset without significant loss of accuracy. Given that this is a small dataset, we have more gain in performance for larger datasets. After p=12, the accuracy starts to decline because the K-SD shortest path distance approximation does not keep up with very small-sized sub graphs. On the flip side, running time starts increasing after p=16.

The reason for this situation is the high number of hubs. The computation of matrix S starts to dominate the running time as it requires O($|H|^3$) time. One advantage of partitioning is that we no longer need to increase the value of parameter k and about five shortest paths are quite enough to compute distances accurately.

4.4.2.2 Effects of Only Must-link or Cannot-link Constraints

Next, we have checked the effect of constraint types individually (see Figure 4.6). When we use the must-link constraints alone, it reduces the weights of both inter-cluster and intra-cluster edges. When the reduction ratio on weights of intra-cluster edges is larger than inter-cluster edges, there is a small gain. Similarly, when cannot-link constraints are used alone, the weights of inter-cluster edges increase more than weights of intra-cluster edges. However, gain in accuracy is greater than when using only positive edges. This situation contradicts with [73] and proves that the informativeness of a constraint type depends on the method how it is applied. Furthermore, the accuracy ratio trend is not steady as number of constraints is augmented. On the other hand, when used together, we get optimal results for the algorithm. On incorrectly validated edges, as seen in Figure 4.8, they cancel the effect of each other. We observed the same phenomena for other algorithms as well.

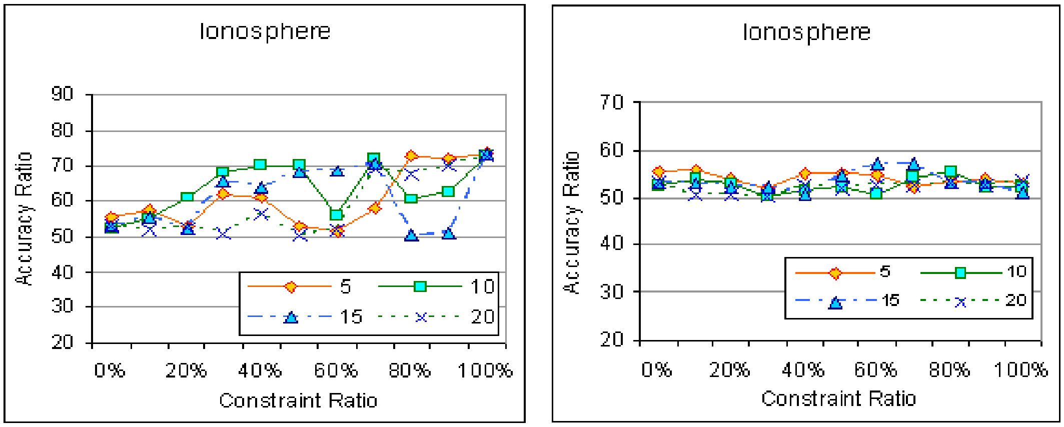


Figure 4-7. The effect of only (a) negative edges and (b) only positive edges on Ionosphere dataset

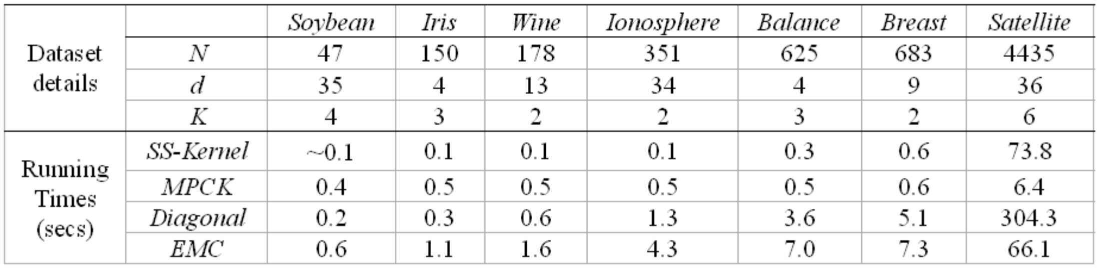
4.4.3 Real Datasets

4.4.3.1 Methods Compared

On real datasets, we compared our MAPCLUS algorithm with the MPCK-Means, SS-Kernel-KMeans and KMeans+Diagonal Metric algorithms, which are publicly available online. We used the same parameters for MAPCLUS: k=5 and p=$|V|$/80 (each subgraph has approximately 80 nodes. MAPClus outperforms MPCK-Means, SS-Kernel-Means and KMeans+Diagonal Metric algorithms on all datasets, except *Breast* and *Wine (Fig. 11).* Also, it runs better than SS-Kernel-KMeans and Kmeans+Diagonal Metric on *Breast* dataset and quite reasonable compared to the MPCK-Means. For *Wine* dataset, graph-based methods such as SS-Kernel-KMeans and MAPClus do not improve the performance significantly and overall accuracy is very low compared to metric-based methods.

SS-Kernel-KMeans concentrates on min-cut objective while MAPClus tries to minimize overall pairwise distance. Furthermore, the same way MPCK-Means and Kmeans+ Diagonal Metric algorithms could not improve the clustering performance for Balance regardless of constraint amount, MAPClus failed to increase the accuracy for Wine dataset. In some of our experiments, we detected the phenomena that the accuracy of the algorithm goes up and down slightly as we increase the constraint amount. As shown in [6], this is a general problem of randomly-chosen constraint sets, where some constraints reduce the clustering performance. Thus, the experiments show that a learning metric or edge weight re-adjustment, using a small amount of constraints, is not always reliable. Compared to other methods, MAPClus is typically more trustworthy even when using few constraints.

Table 4-3. Datasets used in experiments and running time of the algorithms (in secs)



		Soybean	*Iris*	*Wine*	*Ionosphere*	*Balance*	*Breast*	*Satellite*
Dataset details	*N*	47	150	178	351	625	683	4435
	d	35	4	13	34	4	9	36
	K	4	3	2	2	3	2	6
Running Times (secs)	*SS-Kernel*	~0.1	0.1	0.1	0.1	0.3	0.6	73.8
	MPCK	0.4	0.5	0.5	0.5	0.5	0.6	6.4
	Diagonal	0.2	0.3	0.6	1.3	3.6	5.1	304.3
	EMC	0.6	1.1	1.6	4.3	7.0	7.3	66.1

*4.4.3.2 **Running Time Experiments***

We have performed experiments on the running time of the algorithms. All experiments were carried out on 1.7 GHz Pentium IV machine with 512 MB memory. We have performed 10 experiments for each algorithm as we increase the constraint amount by 10%·|*D*|*,* where |*D*| is the dataset size, for each experiment and reported the average running time of these experiments. The running time of MAPClus implementation with multilevel approach is clearly better than other methods, even better than standard K-means algorithm. Thanks to the small graph size at the lowest level, the convergence time is much smaller than other iterative approaches. MAPClus algorithm managed to partition Forest dataset with approximately 62000 instances in about 45 seconds where K-means took 78 seconds to converge. It took 220 seconds for MPCK-Means to partition the same dataset. Most of the time was spend during the swap operation which moves the objects between the clusters.

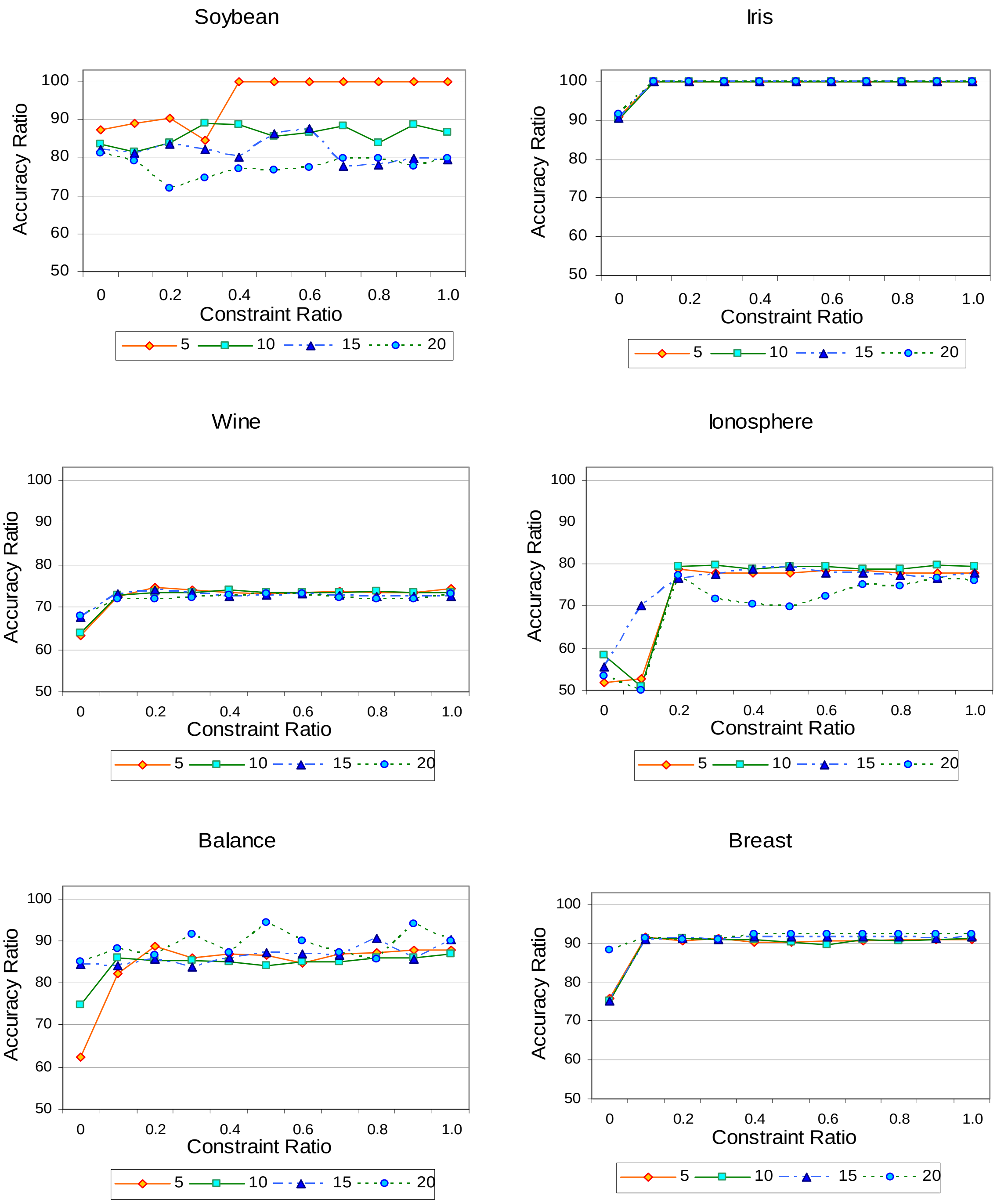


Figure 4-8. Accuracy results of MAPClus, as we vary the number of nearest neighbors, *k* from 5 to 20. Constraints amounts are constraint ratio times |*D*|.

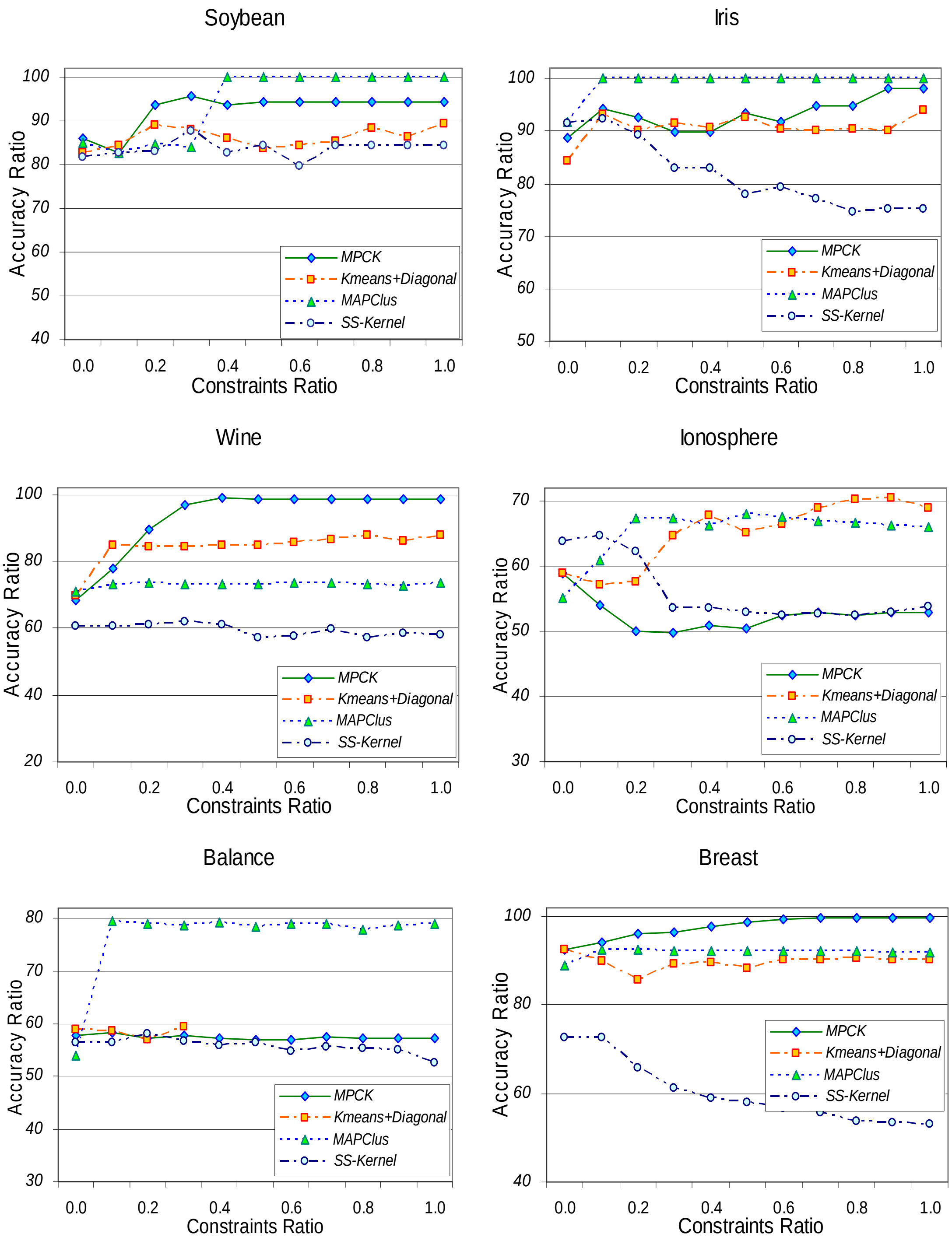


Figure 4-9. Comparison of MAPClus, MPCK-Means, KMeans+Diagonal and SS-Kernel-KMeans algorithms.

4.5 **Summary**

We have presented a framework that, when given a dataset of instances and user constraints, transforms vector data into a graph and improves the clustering algorithm distance metric by adjusting the edge weights based on user constraints. The most important contribution lies in the way the weights are adjusted based on Electro-Magnetic field theory. Instead of modifying the distance metric, it alters the distances between objects in the graph domain. MAPCLUS algorithm allows us to cluster both vector-based and graph-based datasets and it works with distances only as well. Rather than a standard variation of K-Medoids, we can integrate other clustering algorithms into the framework. We have shown than even when using a small amount of constraints, the algorithm improves clustering accuracy significantly.

Chapter 5

MRA-Based Similarity Measures

Molecular similarity is an important tool in drug design and protein engineering for analyzing the quantitative relationships between physicochemical properties of two molecules. We present a family of similarity measures which exploits the ability of wavelet transformation to analyze the spectral components of the physicochemical properties and suggests a more sensitive way of measuring the similarity of biological molecules. In order to investigate how effective wavelet-based similarity measures are against conventional measures, we defined several patterns which indicate a scalar or topological change in the distribution of the properties. The proposed methods were more successful in recognizing patterns in contrast to the state-of-the-art similarity measures.

5.1 **Introduction**

The notion of molecular similarity is widely used in protein engineering and drug design to detect the structural and functional patterns based on molecular properties such as electrostatic potentials, hydropathy, and charge density [74] [75]. Similarity analysis helps classifying molecules according to their physicochemical properties. Many molecular similarity determination approaches [76] [77] [78] compare the molecules of interest according to their electrostatic potentials quantitatively and attempt to reveal the correlations between physicochemical properties and biological activities through similarity analysis. The basic idea is that if electrostatistics is an important driving force for bioactivicity, then molecules with similar electrostatic potential distribution exhibit similar biological functionality. With the advent of Adaptive Poisson-Boltzmann Solver (ABPS) [79], the evaluation of electrostatic distributions and interactions has become more efficient and more accurate. Thus, computational methods for analyzing electrostatic properties has been of great interest.

Molecular similarity determination usually involves evaluating a distance function which compares the relevant properties of two superimposed molecules and returns a numeric value within well-defined limits. A variety of similarity indices have been introduced in the past. Carbo et al. [25] introduced a similarity measure for comparing the molecular density functions which were established in quantum mechanics:

$$CB_{AB} = \frac{\sum_{i,j,k} \phi_A(i,j,k) \cdot \phi_B(i,j,k)}{\sqrt{\sum_{i,j,k} \phi_A(i,j,k)^2} \cdot \sqrt{\sum_{i,j,k} \phi_B(i,j,k)^2}}$$

Although Carbo similarity measure was first proposed to compare continuous functions using integration, the current function is usually evaluated in the discrete space. The value of the CB_{AB} is bounded by the interval $-1.0 \leq CB_{AB} \leq 1.0$ where 1.0 indicates two potentials are identical and -1.0 indicates they are completely different. The measure is still used by many applications due to its sensitivity to the spatial behaviour and the sign of the potentials. However, it comes with a particular drawback referred as *proportionality problem* [23]: if $\varphi_A = k.\varphi_B$, the similarity gets equal to the identity. Consequently, Carbo measure does not take magnitude of electrostatic potentials into account in similarity calculations. Yet higher electrostatic potential magnitude typically indicates higher functionality in biological molecules.

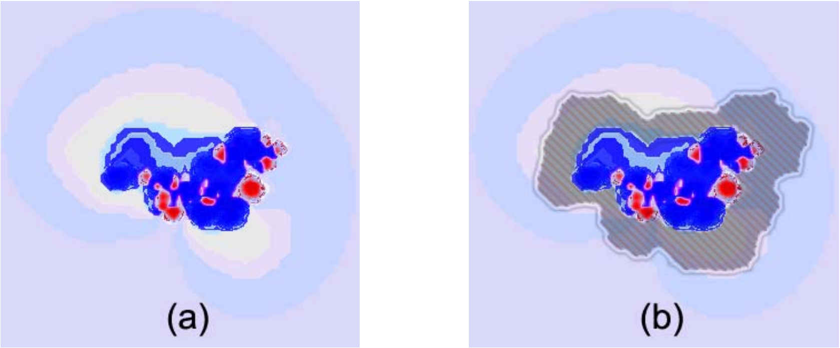


Figure 5-1. An illustration of electrostatic potential distribution of a protein. (a) Darker region indicates larger values residing in the molecule interior. (b) These large values are usually filtered out by using the skin concept in case Hodgkin or Carbo are used.

Hodgkin et al. [24] proposed another measure which mitigates the proportionality problem to some extent by using summation in the normalization factor:

$$HD_{AB} = \frac{\sum_{i,j,k} \phi_A(i,j,k) \cdot \phi_B(i,j,k)}{\sum_{i,j,k} \phi_A(i,j,k)^2 + \sum_{i,j,k} \phi_B(i,j,k)^2}$$

The measure is sensitive to the magnitudes of the potentials in addition to the other characteristics. Like Carbo measure, the similarity value falls within the range of -1.0 $\leq HD_{AB} \leq$ 1.0.

Hodgkin and Carbo measures both can be generalized as cumulative similarity measures considering the fact that they are computed by accumulating the products of potentials and dividing by the normalization factor. The use of products leads to the domination of large values upon small values in similarity calculation. The situation may cause inaccurate comparison of proteins, especially when extreme variance is observed in the spatial distribution of electrostatic potentials. For instance, electrostatic potential distribution in the protein interior may potentially dominate the one on the protein surface and in the protein exterior in similarity calculations due to the relatively large values [76]. If either Hodgkin or Carbo measure is used to compare the proteins, the similarity will be under the influence of mostly the interior distribution, instead of overall distribution. Wade et al. [80] introduced the "skin" concept which focuses on the thin region surrounding the protein. The skin responses to the ion accessibility surface and the debye, which are dependent on ionic screening of electrostatic interaction. However, it ignores the electrostatic potentials in the interior and further away from the debye screening length, which is typically 3^o-7^o from the surface. Another shortcoming peculiar to these measures is that the similarity is overestimated at extreme similarity values due to the again use of products.

Reynolds [23] proposed a non-cumulative measure which focuses on the local differences of the potentials:

$$LN_{AB} = \frac{1}{N}\sum_{i,j,k}\frac{|\phi_A(i,j,k) - \phi_B(i,j,k)|}{\max(|\phi_A(i,j,k)|, |\phi_B(i,j,k)|)}$$

The linear similarity measure has the distinctive property of providing a linear relationship with respect to the proportionality of the compared electrostatic potentials. Unlike other two measures, the similarity is bounded by the interval $0 \leq LN_{AB} \leq 2.0$.

In this study, we present a family of similarity measures which employ the previously established CB [25], HD [24], and LN [23] similarity measures in conjuction with multi-resolution analysis (MRA) [81]. The MRA-based similarity measures transform the protein characteristics into real numbers and apply an appropriate discrete wavelet decomposition to find the corresponding wavelet coefficients. Subsequently, they perform the comparison on the wavelet coefficients. This approach exploits the ability of wavelet transformation to analyze the spectal components of the physicochemical characteristics and suggests a more sensitive way of measuring the similarity of biological molecules.

We also propose a generalized testbed for comparing the characteristics of each measure. To this end, we analyzed electrostatic potential distributions generated by APBS software and identified several patterns indicating a scalar or topologocial change in the distributions. In addition to proportionality pattern [23], we introduce the locality and scaling patterns. The definitions for all three patterns are as follows:

(i) The *proportionality pattern* indicates the change in the ratio of molecular electrostatic potential magnitudes and does not involve any topological changes. In Figure 5.2, an electrostatic potential distribution of a protein is illustrated. In the figure, region A exhibits positive electrostatic properties due to an amino acid with positive side chain. After replacing it with an amino acid with negative side chain, the same region exhibits negative electrostatic properties. The difference between the electrostatic potentials of region A and region B can be expressed as proportionality pattern.

(ii) The *scaling pattern* is commonly encountered in electrostatic potential distributions when an expansion or shrinkage is observed in the area of a particular region with no other significant changes. As seen in Figure 5.3, the area of a positive region expands due to a mutation in the molecule or a change in ionic strength of the medium.

(iii) The *locality pattern* reveals the particular regions which exist in both molecules with similar properties but at different locations. The locality is of paramount importance in homology modelling where we compare the proteins derived from a common ancestor. In homology models, amino acid sequences are more or less conserved, thus possibly leading them to have similar tertiary sub regions but at different locations. Figure 5.4 depicts a simple example where locality pattern is observed. The positive and negative regions illustrate two sub regions of a protein. The positive region is displaced by an angle of α in Figure 5.4b while its distance ℓ to the center point C is being preserved.

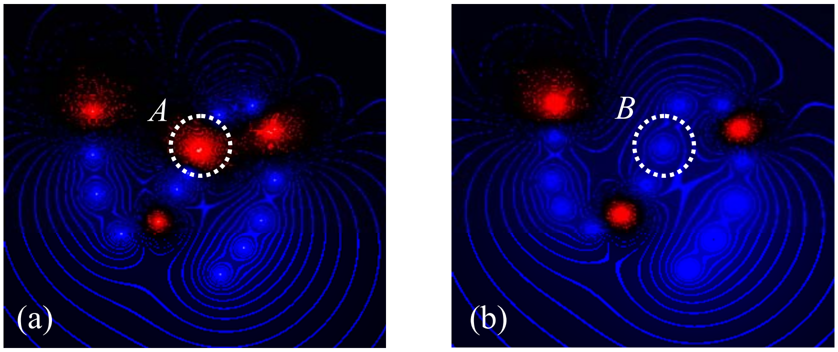


Figure 5-2. Proportionality pattern observed in electrostatic potential distributions. The positive potentials in the region A are replaced by negative potentials such that $\varphi_A = k.\varphi_B$

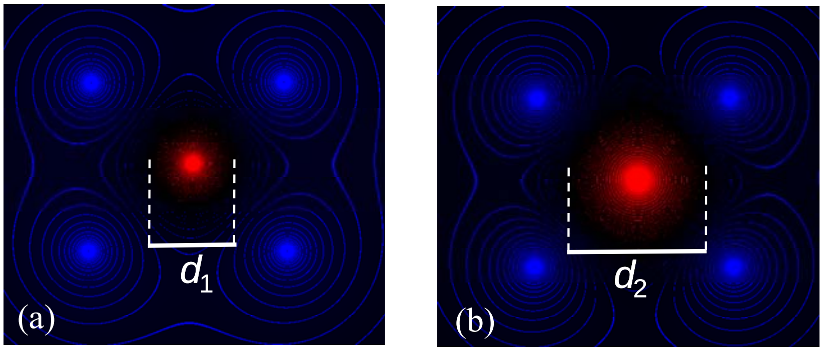


Figure 5-3. Scaling pattern observed in electrostatic potential distributions. The electrostatic potential at a specific region can expand or shrink while rest of the molecule remains the same. Here, $d_2 > d_1$ indicating an expansion in positive potentials (red).

Finally, we have performed a systematic study using the testbed to investigate how sensitive the proposed MRA-based similarity measures to the proportionality, locality and scaling patterns. We have conducted our investigation on the charge distributions of molecules on which the electrostatic potential distribution is grounded.

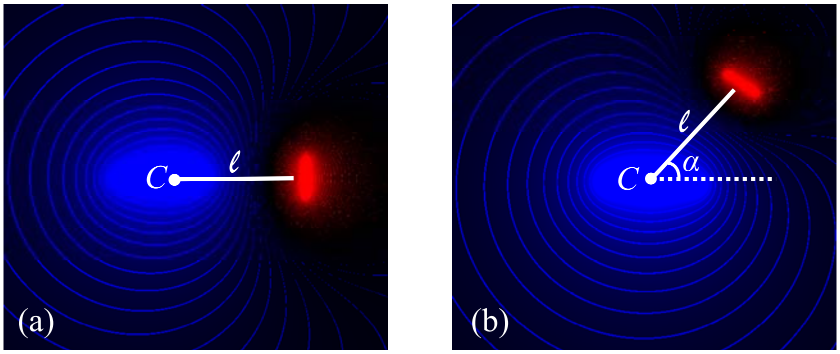


Figure 5-4. Locality pattern observed in electrostatic potential distributions. A specific region may get displaced while the electrostatic potential values in the system remain the same. Here, positive region (red) is displaced by an angle of α. Note that its distance ℓ to the center point C doesn't change.

Empirical evaluations demonstrate that MRA-based methods are more successful in recognizing these patterns in contrast to the state-of-the-art similarity measures.

5.2 Discrete Wavelet Transformation

The wavelet signal transformation is used to divide a raw non-stationary signal into its spectral components at different scales. It helps obtaining further information about the local features of a signal. The process makes use of the concept called multi-resolution analysis. Signals are usually represented in two main domains: time and frequency domains. Time domain is obtained by plotting the amplitude of a signal as a function of time. The change of amplitude is mostly in the form of oscillations similar to the cosine or sine waves. In frequency domain, we study the components of a signal at different spectra. The frequency is used to observe the change in the rate of an oscillating variable. It is measured as the number of oscillations per unit time. If a variable changes rapidly, it is of high frequency. Likewise, if a variable changes smoothly, it is of low frequency [58]. The multi-resolution analysis decomposes the signal into its frequency components and observes the changes in the signal at different frequency bands.

In wavelet analysis, the frequency bands are disintegrated by performing a convolution operation on the original signal and the wavelet function. The convolution operation involves shifting the wavelet function, multiplying the original signal by the wavelet and summing up the results. The continuous wavelet transform is defined as follows:

$$W_f(\tau, s) = \frac{1}{\sqrt{s}} \int f(t) \psi(\frac{t-\tau}{s}) dt$$

where *ψ(t)* is the mother wavelet function, and the variables τ and s are the translation and scale parameters, respectively. The mother wavelet function is a small wave function used as a prototype to generate the window functions for each τ and s values. The scale parameter defines the length of the wavelet and makes the window function react to a specific frequency band. When the window function is multiplied by the original signal, the output is the spectral component that resonates with the frequency band defined by the scale. If the scale is low, the wavelet function extracts the high frequency components. When we increase the value of the scale, we can extract the lower frequency elements which may span even the entire signal. The translation parameter defines the location of the window. During the extraction process, the window function is shifted through the signal. At each location, we check whether the window function and the signal resonate. If they have a perfect match, the resulting signal will be identical to the wavelet function. Otherwise, it will be zero. The operation is applied to all time locations in the original signal. When we sum up the resulting signals, we obtain the frequency component or wavelet coefficient whose frequency is same as the wavelet. The high frequency elements represent the short-range changes in the signal and thus are referred as *detail coefficients* whereas the low frequency elements represent the long-range changes and are referred as *approximation coefficients.*

The continuous wavelet transformation scans through all spectrum and computes all coefficients. It is computationally very expensive. In addition, analyzing all wavelet coefficients is highly redundant. The discrete wavelet transformation is a special kind of wavelet transformation that achieves significant computational improvement over continuous wavelet transformation while providing sufficiently non-redundant coefficients. The DWT takes a discrete function as input. It requires the continuous signal function to be transformed into the discrete form. The discrete function is usually created by sampling the continuous function at discrete time values. In DWT, the selection of scale parameters is s=2 and τ=1, i.e. the wavelet length is dilated by 2 at each level and shifted by 1 at each step. The discrete wavelet transform is defined as:

$$DW_f(\tau, s) = \frac{1}{\sqrt{2}} \sum_t f(t)\psi(2t - \tau)$$

The DWT consists of two filter banks that are derived from wavelet functions: high-pass filter and low-pass filter. The same signal is passed through the filters separately. In case s=2 and τ=1, the low-pass filter removes all frequency elements above the half of the highest frequency in the signal. The high-pass filter, on the contrary, filters out all the frequencies below the half of the highest frequency. For example, if the highest frequency in the signal is 500 Hz, we obtain the 0-250 Hz components via low-pass filter and 250-500 Hz components via high-pass filter. The low-pass filter and high-pass filters decompose the signal into its approximation and the detail coefficients by using equations:

$$c_f^k = \frac{1}{\sqrt{2}} \sum_t f(t) \psi_{low}(2k - t)$$

$$d_f^k = \frac{1}{\sqrt{2}} \sum_t f(t) \psi_{high}(2k - t)$$

respectively.

The DWT explores a range of frequency components instead of all frequencies individually. Multi-resolution analysis takes advantage of this property. It repetitively applies the decomposition to the approximation coefficients to produce multi-level coefficients. Suppose the original signal has a maximum frequency of 200 Hz. At the first decomposition level, the signal is passed through the high-pass and low-pass filters. The output of the low-pass filter is 0-100 Hz components whereas the output of the high-pass filter is 100-200Hz components. We take the output of the low-pass filter and input it into the second level decomposition. The output is 0-50 and 50-100 Hz components. The procedure is repeated until no more spectral elements left. The decomposition is illustrated in Figure 5.5.

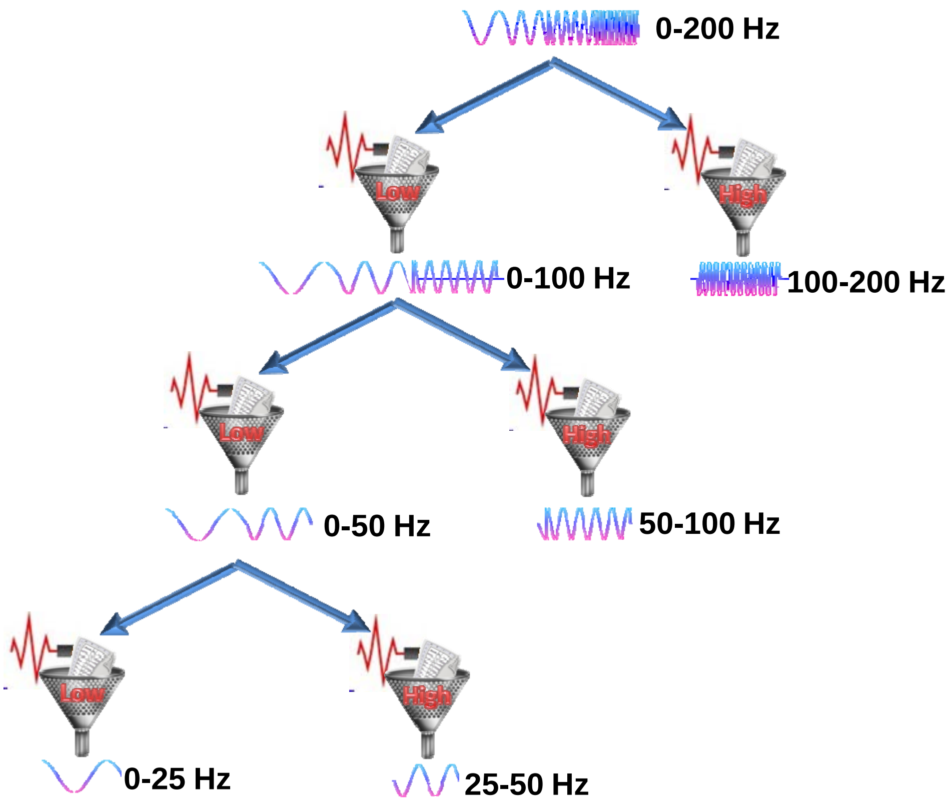


Figure 5-5. The illustration of a wavelet transform on a signal with 0-200 Hz spectrum

5.3 **Wavelet-Based Similarity Measures**

The intuition behind this study is that almost all biological data can be expressed in some form of time series and this property allows us to apply the time-series techniques to many biological applications. We hypothesize that the similarity information can be captured with higher sensitivity in time-series domain than the original domain. We exploit the similarity measures through a concept called multi-resolution analysis (MRA) which is practically relevant to the discrete wavelet decomposition. MRA produces a series φ =(φ_1, φ_2,..., φ_m) of coefficients by decomposing the signal such that their corresponding frequencies are ordered as $f_1>f_2>...>f_m$. Each coefficient represents the same signal at different resolutions. Here, the resolution defines the amount of detail information in the signal. Depending on the application, each resolution may be of different importance. For example, MRA is widely-used in medical domain to analyze the medical images. In a cancer image, early stages of the cancer can be determined by examining fine-resolution coefficients, while the late stages are more likely to appear at coarser resolutions.

The key insight of our algorithm is that the low-frequency coefficients correspond to the more global changes and thus, more important than the high-frequency coefficients in similarity calculations. The high-frequency coefficients represent the local changes which may vary significantly within a very short range. They are usually considered as noise in many applications and filtered out of the signals to obtain more smooth representation. However, in biological domain, each piece of information is usually valuable and should be taken into consideration carefully.

Note that each coefficient is a spectral component of the original signal and contributes to the similarity value proportional to its importance. The importance may differ depending on the domain. Therefore, we calculate the similarities on each coefficient separately and then take the weighted sum of the calculated similarities. We define the following wavelet-based similarity measures in this context.

Wavelet Similarity Index with Carbo Distance:

$$W_{Carbo}(A,B)=\left(\sum_{l=1}^{m}w_l\right)^{-1}\cdot\sum_{l=1}^{m}w_l\frac{1}{N_l}\sqrt{\frac{1}{2}-\frac{\sum_{i,j,k}\varphi_A^l(i,j,k)\cdot\varphi_B^l(i,j,k)}{2\cdot\sqrt{\sum_{i,j,k}\varphi_A^l(i,j,k)^2}\cdot\sqrt{\sum_{i,j,k}\varphi_B^l(i,j,k)^2}}}$$

Wavelet Similarity Index with Hodgkin Distance:

$$W_{Hodgkin}(A,B)=\left(\sum_{l=1}^{m}w_l\right)^{-1}\cdot\sum_{l=1}^{m}\left(w_l\frac{1}{N_l}\sqrt{\frac{1}{2}-\frac{\sum_{i,j,k}\varphi_A^l(i,j,k)\cdot\varphi_B^l(i,j,k)}{\sum_{i,j,k}\varphi_A^l(i,j,k)^2+\sum_{i,j,k}\varphi_B^l(i,j,k)^2}}\right)$$

Wavelet Similarity Index with Linear Distance:

$$W_{Linear}(A,B)=\left(\sum_{l=1}^{m}w_l\right)^{-1}\sum_{l=1}^{m}w_l\frac{1}{N_l}\sum_{i,j,k}\frac{\left|\varphi_A^l(i,j,k)-\varphi_B^l(i,j,k)\right|}{\max(\left|\varphi_A^l(i,j,k)\right|,\left|\varphi_B^l(i,j,k)\right|)}$$

In the formulas above, φ is a series of wavelet coefficients and w is the weight function for the coefficient levels. We typically choose the weights as k^{th} power of 2 for the k^{th} coefficient level. According to our empirical evaluations, the selection of weights works fine with many biological applications. Wen et al. [59] also used a similar weight function in detecting protein sequences, with the motivation of normalizing the energy levels of wavelet coefficients. However, different weights can always be used for specific applications.

5.4 Empirical Evaluations and Discussion

To quantitately assess the pattern recognition capability of MRA-based similarity measures, we generated several toy data, where each dataset projects an isolated pattern. As proof of concept, we used simple spherical representation for molecule structures. We assumed that the charge distributions within the molecules were uniform. The wavelet coefficients were obtained by applying discrete wavelet transformation on the three-dimensional spatial distribution of molecular charges. In the experiments, we investigated similarity behavior of the measures with respect to the proportionality, locality, and scaling patterns. In our evaluations, “0„ indicates that two molecules are identical. All other values indicate intermediate dissimilarity levels between two molecules.

5.4.1 Proportionality

The proportionality pattern implies a change in the ratio of charge magnitudes. After a single amino acid mutation, charge magnitudes close to the mutated residue may change significantly and as a result, electrostatic potential values may increase or reduce dramatically. To simulate this pattern, we generated two equal-sized spherical molecules and varied the ratio between their charge magnitudes from -1.0x to 1.0x as shown in Figure 5.6.

Proportionality experiments suggested no change in similarity behavior whether or not the wavelet coefficients were used. CB and WCB measures both behave like a step function for positive and negative proportionality constants due to the fact that they are based on cosine similarity and cosine similarity does not take charge magnitudes into

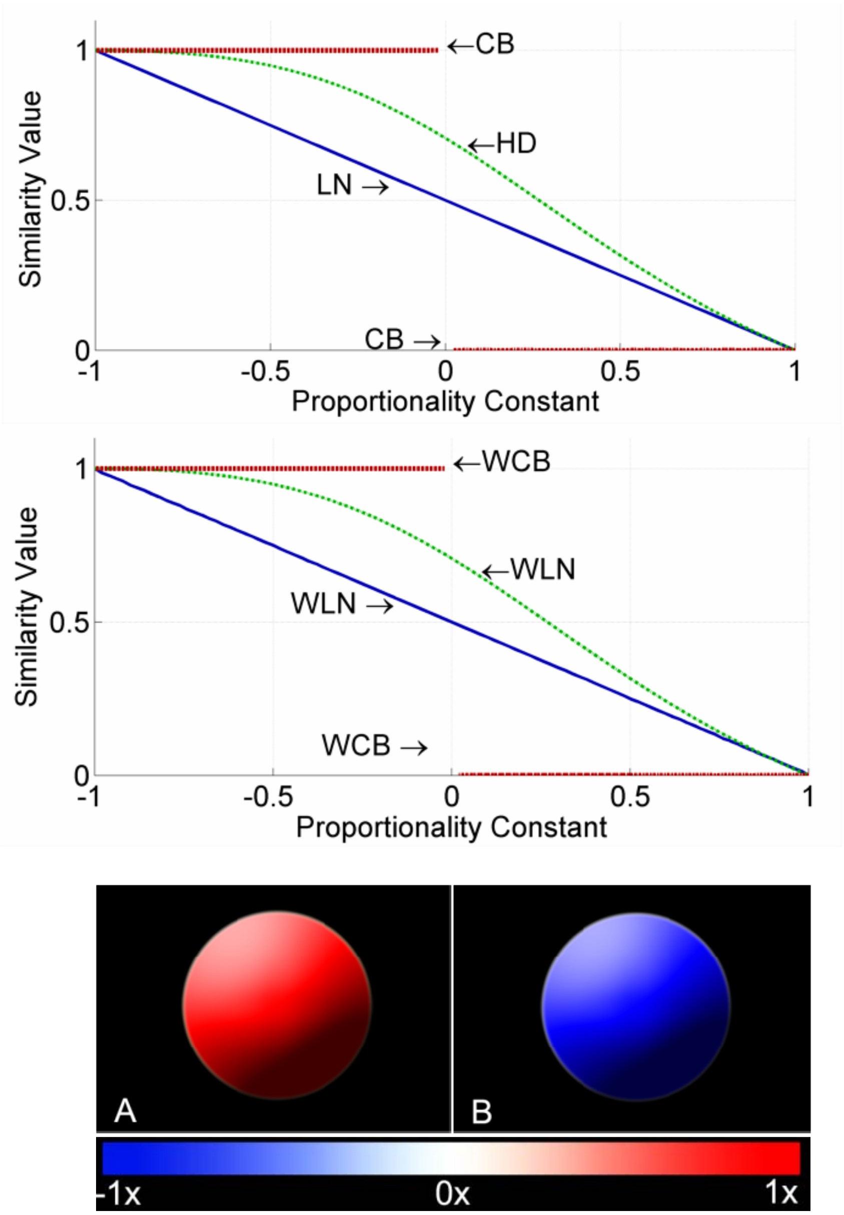


Figure 5-6. The similarity between molecules A and B whose charge distributions are maintained by a proportionality variable. Non-MRA and MRA-based measures show similar behavior for the same proportionality constants.

account. HD and WHD similarity measures both overestimated the dissimilarity at extreme values. While it may be true, they managed to restrain the overestimation for the extreme similarity values, i.e. HD and WHD similarities scale almost linearly when the objects get more similar. On the contrary, LN and WLN similarity measures both displayed a linear relationship between similarity and proportionality constant. The reason why non-MRA and MRA measures exhibit similar behaviour lies in the wavelet transformation: The wavelet decomposition analyzes the charge distributions according to their spectral or frequency characteristics. In frequency domain, the magnitudes are not taken into consideration. Instead, changes in the magnitudes are considered in the decomposition. Therefore, when the wavelet coefficients were examined for different proportionality constants, the ratio between the charges were also preserved in wavelet domain for the coefficients.

For completeness, we also analyzed the behaviours of similarity functions when proportionality constant k<-1.0 and k>1.0. In our analysis, we observed that out of interval [-1, 1], all similarity functions except for CB and WCB change direction. When the proportionality constant k gets closer to the -∞ and +∞, HD and WHD similarity functions converge to $1/\sqrt{2}$ whereas LN and WLN converge to $1/2$. A similar behaviour was also reported by Petke et al. [75] in their analyis of proportional datasets

5.4.2 Scale

The scaling pattern indicates a change in the area of a specific region with no significant change in other characteristics of the region. One scenario for the pattern is that there might be two amino acids right next to each other, one with a negative side chain and the other with a positive side chain. If the residue with negative side chain is replaced with an amino acid with positive side chain, the area of positive charges, as well as electrostatic potentials, in the neighborhood will expand proportional to the net charge of new amino acid. Similarly, substitution of a hydrophobic amino acid residue with a hydrophilic amino acid may cause the same effect. Such mutations affect the hydrophobic cores within the protein, which may result in expansion or shrinkage in the size of whole molecule or just a particular sub region.

In order to investigate the effect of scaling in similarity analysis, we generated several spherical molecules with varied radius. We have used three charge distributions in scaling experiments. The first experiment assumes $\varphi_A = 1$ and $\varphi_B = 1$, where φ_A is the charge per unit for the molecule which is used as reference in comparisons and φ_B is the charge per unit for the molecule whose radius is varied. Remember that charges are uniformly distributed inside the molecules. Figure 5.7 depicts the behavior of similarity measures as we increase the scale from 0x to 3x. Even though all measures demonstrated similar behavior, WCB measure underestimated the similarity between two molecules when the scale is less than 1x.

In the second experiment, we adopted the case where $\varphi_A = 1$ and $\varphi_B = 4$. CB measure could not handle the experiments with different charge values at all and presented the same plot as in Figure 5.8. WHD, WLN, HD and LN measures managed to discriminate the objects reasonably well when their charge distributions different.

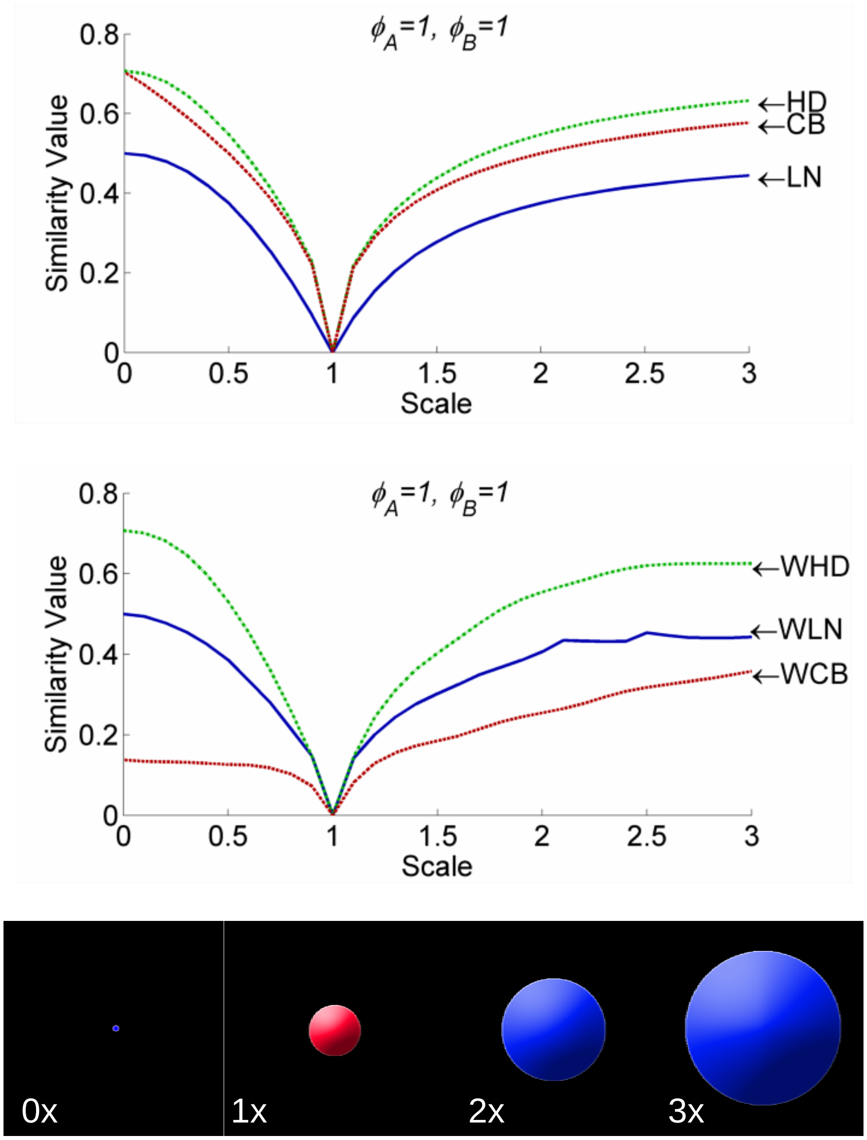


Figure 5-7. The similarity between molecules A and B whose radii differ by a scale constant. The molecular charges are $\boldsymbol{\varphi_A = 1}$ and $\boldsymbol{\varphi_B = 1}$. Although non-MRA (a) and MRA-based measures (b) show similar behaviour, the slope of WCB similarity measure indicates an underestimation of the similarity when the scale is less than 1x.

However, in wavelet domain, the point where the minimum similarity value is observed is shifted to the left by a ratio of $r_A\sqrt{\varphi_A/\varphi_B}$. Interestingly, both molecules have the same sum of charges at this scale value. When molecular charges were $\varphi_A = 4$ and $\varphi_B = 1$, we observed that the minimum similarity value is shifted to the right by a ratio of $r_A\sqrt{\varphi_A/\varphi_B}$. The behavior indicates a clear relationship between the minimum similarity value and the sum of charges in the system. The approximation function in the wavelet transformation performs like a mean function and takes the average of the charges at each level. At the lowest level of the decomposition, the approximation and detail coefficients become equal for two molecules due to the overall distribution of the charges. In the weight function we use, the lowest level has the highest weight because of the importance of the information it presents. Consequently, the equality at lower levels dominates the overall similarity value. This property of MRA-based similarity measures is quite attractive for similarity analysis of homology models, whose biological functionality is usually compared according to the net charge in the molecule.

When the molecules have opposite charges, it has a negative effect on the similarity. We have the similar behaviours for similarity measures, however the similarity value was relative small compared to the configuration where identical charges were used.

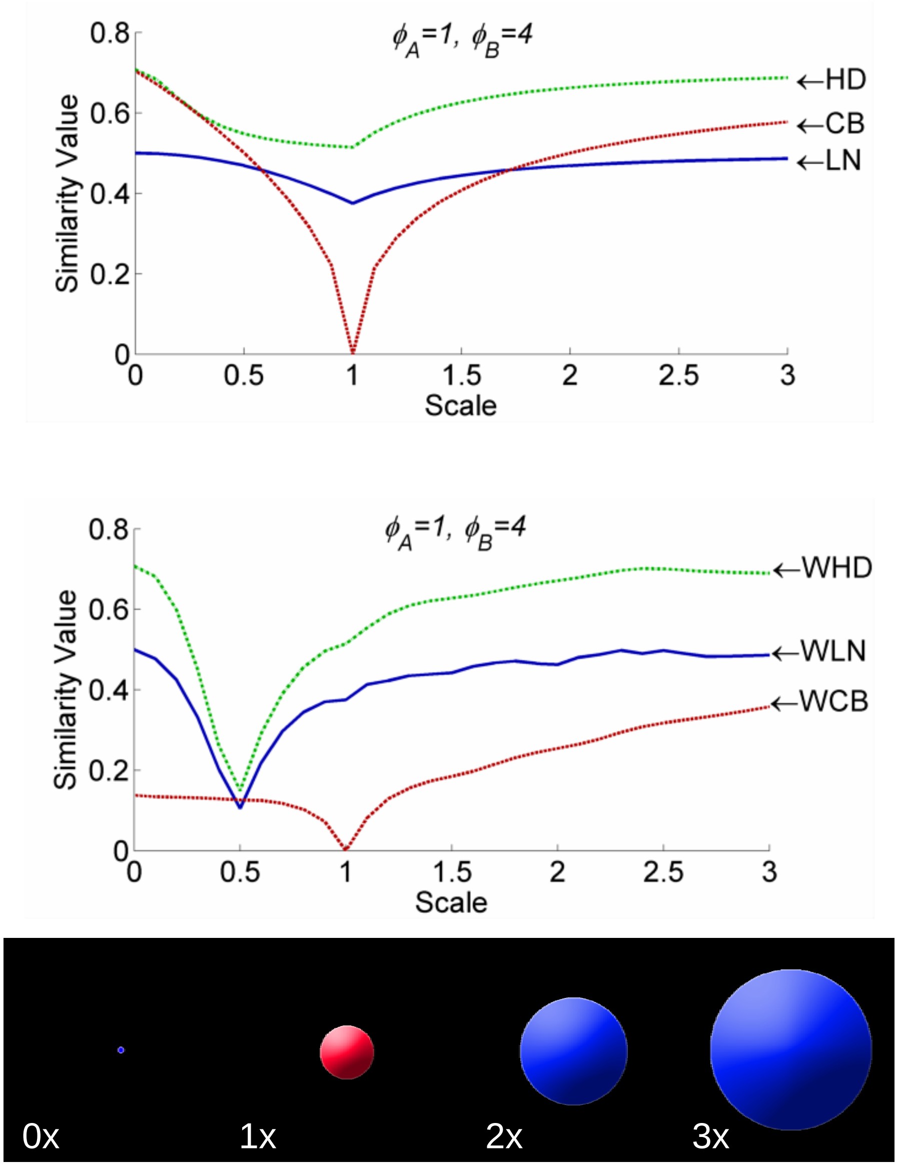


Figure 5-8. The similarity between molecules A and B whose radii differ by a scale constant. The molecular charges are $\boldsymbol{\varphi_A} = \mathbf{1}$ and $\boldsymbol{\varphi_B} = \mathbf{4}$. The point where the minimum similarity value is observed is shifted to the left by $\boldsymbol{r_A\sqrt{\varphi_A/\varphi_B}}$ in wavelet domain. Interestingly, the sum of charges has same value at this scale.

5.4.3 Locality

The locality criteria measures how much a particular region is displaced from its original position as a result of a mutation or an external singularity. The mutation at a single residue may affect the position of a binding interface of two complementary proteins that are involved in a complex structure. Also, we observe the same pattern in homology modeling. The homology models usually have similar amino acid sequences, and this situation causes homologous proteins to have similar secondary structures. Due to the substitutions in the sequence, these sub regions may be found at different locations in space [27]. Considering C^{α} atoms forming the backbone structure of the homologous proteins are more or less conserved, the displacement can be expressed in terms of a displacement angle. In order to simulate these scenarios, we generated two spherical molecules that were attached to each other as illustrated in Figure 5.9. The relationship between the molecular radii was r_A=2r_B. The charges were equal and uniform inside the spheres as in the previous experiments. At each step, molecule B was displaced counter-clockwise by an angle of α on the surface of molecule A. We used the first charge distribution (α=0^{o}) as reference and compare it against all other configurations.

Figure 5.9 compares the behavior of conventional and MRA-based similarity measures as we displaced the molecule B by an angle of 15^{o} at each step. LN, HD and CB measures cannot discriminate the displacements greater than 60^{o}, when molecule B in two different configurations no longer overlapped. CB and HD functions calculated the same similarity values for all displacement angles. Conversely, MRA-based measures

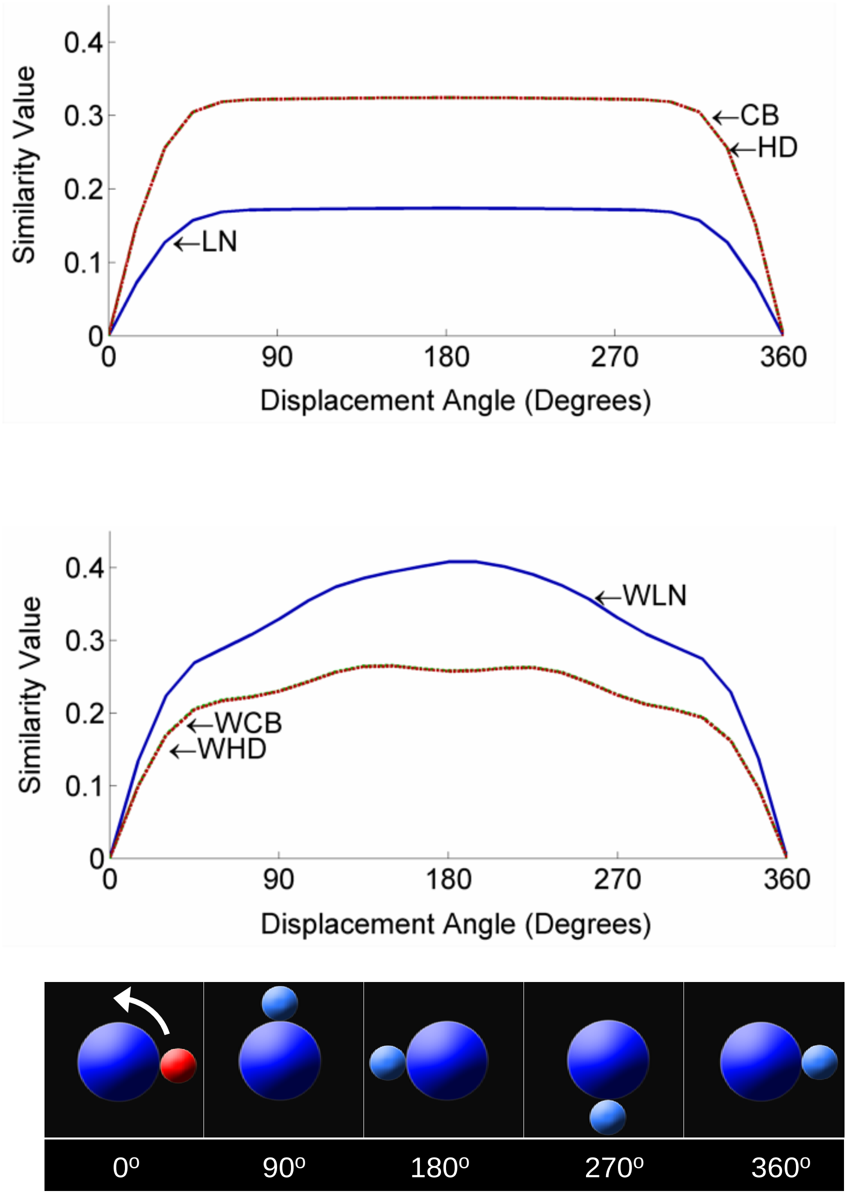


Figure 5-9. The similarity between molecules A and B where the molecule B is displaced counter-clockwise by a displacement angle of α on the surface of the molecule A. The conventional similarity measures (a) cannot differentiate the similarity values once α>60. The MRA-based similarity measures (b), on the contrary, managed to discriminate different configuration much better.

managed to discriminate the displacements even beyond this angle. Similar to HD and CB, WHD and WCB measures showed the same behavior in the experiments and overlapped for all displacement configurations. Even though they were more sensitive to the locality than CB and HD measures, they couldn't discriminate the displacements greater than 120^{o}. In contrast, WLN outperformed all other measures.

During our experiments, we have observed that the steepness of the MRA-based similarity functions reduced dramatically when the displacement angle was a multiple of $\pi/2$. In other words, the similarity was underestimated for these configurations. Such behavior was a natural result of wavelet transformation. When performing wavelet decomposition in multiple dimensions, the original wavelet function was applied in turn to each of the dimensions in an orthogonal fashion. Once again, the wavelet function decomposes the data according to their spectral information. When the displacement angle gets closer to the multiple of $\pi/2$, the molecules shows similar characteristics in terms of spectral information.

When we use different or opposite charges, the similarity measures have similar behavior as we see in Figure 5.8. While this may be true, likewise scaling experiments, the similarity was again relatively small in contrast with the default configuration.

Table 5-1. Comparison of Similarity Indices with respect to their support for locality, proportionality and scaling patterns

Pattern	CB	HD	LN	WCB	WHD	WLN
Proportionality		✓	✓		✓	✓
Locality				✓	✓	✓
Scaling	✓	✓	✓	✓	✓	✓

5.5 **Conclusions**

In summary, we have proposed a novel approach for molecular similarity determination, which integrates the established similarity measures and the multi-resolution analysis. In order to establish a test bed for molecular similarity measures, we have studied several patterns which are commonly observed in charge and electrostatic potential distribution of molecules, especially biological molecules and proteins. Using the test bed, we have reported a systematic study of pattern recognition supported by each similarity measure. The comparisons of similarity measures with respect to their support for different patterns are given in Table 5.1. The results of empirical evaluation suggest that the pattern recognition ability of the state-of-art similarity measures can potentially be enhanced when they are used in conjunction with the multi-resolution analysis. Even though we have studied only electrostatics in this paper, the proposed measures can potentially be applied to other physicochemical characteristics as well.

Chapter 6

Similarity Analysis of Complement Proteins using Wavelet Transform

Advances in computational biology allow us to determine 3-D protein structures and evaluate their properties, such as electrostatic potential and hydropathy, rapidly. However, electrostatic potential is more distinguished than any other characteristic because it drives the intermolecular interactions and keeps the protein stable within itself by cooperating with the hydrophobicity. Due to its strenuousness in bioactivity, we need to perform comprehensive electrostatic potential analysis on proteins in order to understand their biological functionality and consequently design novel drugs [82] [83]. In this chapter, we shall present the validity of wavelet-based similarity measures through the hierarchical clustering of several complement protein datasets. The datasets were obtained either via alanine scanning or homology modeling, two widely-used methods in drug design to analyze the mutations and unknown protein structures, respectively.

Alanine scanning is a powerful tool for protein analysis which replaces every single ionizable amino acid by alanine which is neutral in charge [84]. In molecular biology, the alanine scanning is used to select residues in a protein sequence for mutation in order to improve the functionality and molecular properties. The idea is to determine whether a specific side chain group of a specific residue plays an important role in bioactivity. The residues without any important role are considered as sites for substitutions for the generation of mutant alleles. Such residues usually present a high probability of getting a mutation that allows folding while giving a phenotype. In alanine scanning, the sequence of a protein is scanned using an overlapping window of five residues as looking for charged residues. All the charged residues are substituted with alanine using vitro mutagenesis or computational methods. The mutant allele is then examined for phenotype. In this study, we have focused on the C3d and Efb-c mutants that form the C3d/Efb-c complex [82]. The complex prevents the activation of immune response and causes infections on the host. In order to understand the interactions between C3d/ Efb-c, we performed a alanine scanning and quantitative analysis on C3d and Efb-c mutants. We have followed a computational approach for modeling the mutants with the *Whatif* program widely used for alanine scanning in the research community [85]. Each mutant corresponds to the substitution of a single residue on the original protein. The goal of the analysis is to determine which mutation causes enhancement or inhibition in the C3d/Efb-c association.

Alternatively, the wavelet-based similarity measures have been employed to analyze the proteins that were computationally generated by homology modeling. In homology modeling, a sequence of a protein whose topology is unknown is compared against a set of template proteins whose sequence and topology is available, by using an alignment algorithm [27]. After finding the template with the best alignment, a sequence-to-structure alignment is applied to the sequence of unknown protein and the tertiary structure of the template in order to predict the tertiary structure of the unknown protein. The sequence-to-structure alignment should be performed with high precision. The protein structure is so sensitive that single residue misplacement may cause unrealistic models. This is to say that the computationally discovered protein becomes useless for an application like drug design. The homology modeling was applied to generate the unknown structures of CCP modules of Factor H in the second set of experiments. The Factor H in particular regulates the complement system by binding to the C3b protein and prevents the complement components attacking the host cells. It is believed that the electrostatic diversity of CCP modules drives the interactions for distinguishing self from non-self and the key information to understand the bioactivity of the Factor H lies in understanding the electrostatic characteristics of the CPP modules [86]. As we shall see in the experiments, we have computed the tertiary structures of unknown CCP modules using the known modules as template and performed a comparative analysis on the CPP using the wavelet-based similarity measures. Next, we will give an overview of the complement system and describe the functionality of each protein mentioned so far in details.

6.1 The Complement System Overview

Every living organism is equipped with an immune system that protects the organism from invading pathogenic microorganisms. The immune system is a remarkably versatile defense system that initiates and maintains protective responses against a vast variety of foreign invaders called antigens [87]. The elimination or neutralization of the antigens is accomplished by complex interactions between components of the adaptive and innate immune system. The adaptive immune system is composed of highly specialized, systemic cells that can virtually recognize any foreign microorganisms and eliminate them with tailored responses. The lymphocytes generated by adaptive immunity have a single type of receptor protein with an unlimited repertoire of variants. By courtesy of this receptor, the lymphocytes can bind to and recognize any type of antigens. The recognition ability of the lymphocytes is so sensitive that it can distinguish between two proteins that differ in only a single amino acid. The adaptive immunity can remember specific pathogens after an initial encounter. It can adapt itself accordingly to mount stronger attacks next time the same pathogen is encountered. Unlike adaptive immune system, the innate immune system is not adaptable and does not change over the course of an individual's lifetime. It provides a network of antigen-nonspecific defense mechanisms that the organism activates immediately or within several hours after the invasion of an antigen.

The complement system is an integral part of the innate immune system that can mediate a variety of immune reactions such as triggering the downstream inflammatory, directly attacking the membrane of the intruding micro organisms and simulating the antibody production which is essential for the proper elimination of the antigens [88][89]. To this end, it promotes and regulates the phagocythosis or lysis of foreign cells, macromolecules and host tissue breakdown products [90].

The complement system consists of more than 30 proteins, both soluble and membrane bound. These plasma and membrane proteins interact with each other when the complement system is activated via three pathways: classical, alternative and lectin. The Complements component 3 (C3) functions as the central protein of the complement system and provides amplification of immune response. It serves as a link between innate and adaptive immune components [82]. Thus, it is required for both classical and alternative complement activation pathways. During the activation pathways, C3 protein is cleaved into two active fragments, C3a and C3b, by C3-convertase enzyme. The cleavage of the C3 enhances the clearance of the foreign cells by promoting binding of the antibodies to the infection site.

C3a peptide mediates histamine release from several immune cells such as mast cells, basophiles, neutrophils, and eosinophils. The histamine release triggered from mast cells and basophiles increases the vascular permeability to leukocytes and other proteins so as to allow them to engage foreign cells, whereas the histamines released from eosinophils and neutrophils cause inflammatory response[91]. C3b fragment usually binds to the foreign cell membrane directly to make the cell more attractive to the phagocytic host cells which have receptors for C3b. This fragment, thus, acts as binding enhancer for the process of phagocythosis. C3b can be further be cleaved to produce C3d protein which activates the B cells or lymphocytes. The B lymphocytes possess C3d receptor called complement receptor 2 (CR2). The interaction between C3d and CR2 plays a significant role in lymphocyte activation and maturation. During the infection, C3d binds to the CR2 receptor on B cells and enhances their response to the antigens greatly.

The complement activation has a tremendous potential for self-amplification and destruction. The continuous activation of C3 causes damage to the both host cells and microbes since the complement system is non-specific and does not discriminate between host and foreign cells. In addition, uncontrolled amplification accelerates the depletion of complement proteins in the bodily fluid. In order for the complement system to function properly and not cause the oponization of host cells, the complement should be regulated by specific inhibitors such as factor H, factor I, decay accelerating factor, C1 inhibitor. Many of these inhibitors are expressed on the surface of the host cells but not foreign cells [92]. Therefore, the complement activation causes limited damage to the host cells compared to the pathogens. Among all inhibitors, factor H plays a distinguished role in complement activation and regulation.

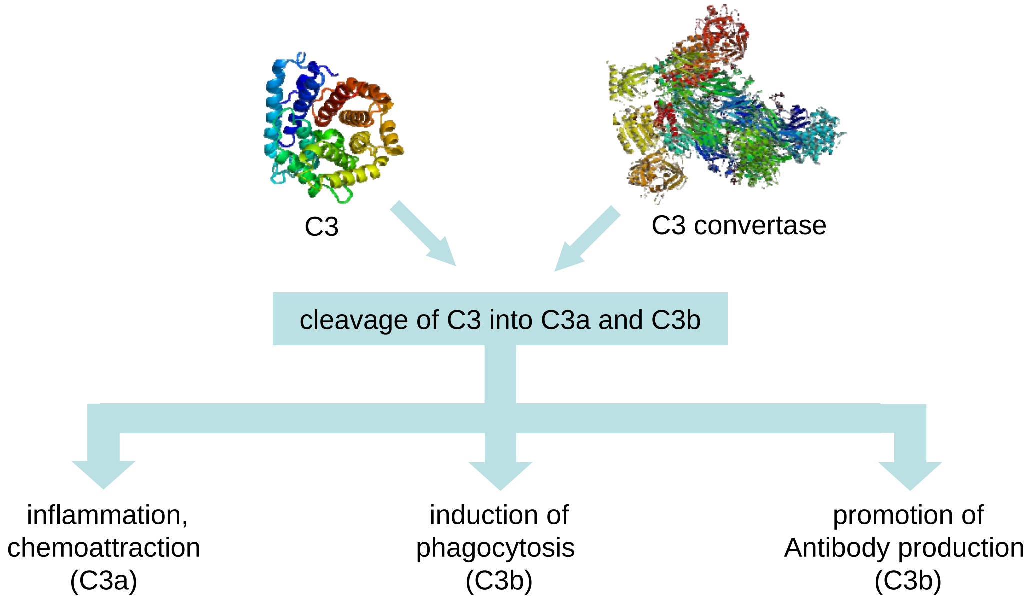

C3
C3 convertase
cleavage of C3 into C3a and C3b
inflammation, chemoattraction (C3a)
induction of phagocytosis (C3b)
promotion of Antibody production (C3b)

Figure 6-1. The functionality of C3a and C3b fragments produced by the cleavage of C3.

Amplification of complement activity is achieved by association of C3b and factor B enzyme; the association results in C3bBb complex, also known as C3 convertase. The presence of C3b protein in the medium increases the convertase formation, which in turn further activates the complement system. However, if C3b density decreases in the serum, the complement activation stops. To this end, factor H competes with factor B for binding to C3b and prevents further pathway activation. Another favorable characteristic of factor H is its ability to protect the host cells. The type of cell surface to which C3b binds affects which factor binds to the C3b. The host cell surfaces possess sciatic acid, which favors the binding of C3b with factor H. However, the microbial cells lack the siatic acid, which favors the binding of C3b with factor B. In the host cells, the C3 convertase is inhibited by factor H and complement activation is not carried on any further. Thus, the host cells are protected against the complement proteins. The microbial cells, on the contrary, still remain as targets for further complement activation.

6.2 **Methods**

6.2.1 Similarity Analysis

The electrostatic interactions drive the biological function of complement proteins and similar electrostatic characteristics possibly indicates similar physicochemical properties as well as biological role. Our study aims to perform a comparative electrostatic analysis on immune-related protein datasets generated by either alanine scanning or homology modeling and to identify protein families in which the proteins typically have similar tertiary structures and functions.

Our systematic analysis consists of three steps:

i. Acquisition of Molecular Structures

ii. Electrostatic Potential Calculation

iii. Similarity Calculation

Our similarity analysis assumes the three-dimensional structures of all proteins in each dataset are available at atomic resolution. In our experiments, C3d and Efb-c datasets consists of mutant proteins that were generated via alanine scanning. *Whatif* [85] program was used to generate C3d and Efb-c mutants by substituting each ionizable amino acid with alanine one at a time. Once all alanine mutants were obtained, the structures were superimposed in order to assure that rotation- and displacement-invariant comparison could be performed. The superimposition was obtained by overlapping the backbone C atoms of mutant proteins as close as possible to each other. In Factor H experiments, we shall investigate the functional similarity of 20 modules which have the complement control protein (CCP) architecture. Factor H consists of 20 homologous proteins out of which only 11 modules were derived from NMR or X-Ray crystallography experiments. The rest of the CCP modules were computationally obtained through protein threading method. Threading attempts to solve the tertiary structure of an unknown protein by looking for a template protein through the set of currently known structures and using the most appropriate template to calculate the ideal coordinates for the backbone C- atoms.

Following the acquisition of molecular structures, the method calculates the electrostatic potentials by using the atomic charges composing the electric fields inside the protein. The problem here is that protein structure files only define the coordinates of the atoms and lack the atomic charge values. PDB2PQR [93] software is utilized to integrate missing hydrogen atoms into the structure and assign the charge and radius values to the coordinates of the atoms. The charges are determined according to the PARSE [94] force field. Adaptive Poisson-Boltzmann Solver (ABPS) [95] was subsequently employed to calculate the electrostatic potentials resulting from the protein charge distribution. ABPS actually solves the linearized Poisson-Boltzmann equation given below:

$$-\nabla \cdot \varepsilon(r)\nabla \varphi(r) + \varepsilon_0 \varepsilon(r)\kappa^2(r)\varphi(r) = \frac{4\pi e^2}{\varepsilon_0 k_B T}\sum_{i=1}^{F} z_i \delta(r - r_i)$$

For each electrostatic calculation, the protein is embedded into a three-dimensional grid structure with 129x129x129 grid points. Even though ABPS supports several grid dimensions, our preliminary calculations demonstrated that selecting the grid value as 129 provides better results for both small and large proteins. ABPS calculates the spatial distribution of electrostatic potentials at discrete grid points rather than over the continuous space. In our calculation, we assume the solvent dielectric coefficient is set to 1, i.e. vacuum environment. Another feature of the program is ability to generate isopotential surfaces for visualization of electrostatic potentials surrounding the protein. At this point, the spatial distributions of electrostatic potentials are represented by a 128x128x128 matrix. We perform three-dimensional discrete wavelet transformation on the matrix and decompose it into corresponding detail and approximation wavelet coefficients. The transformation is iteratively applied on the approximation coefficients to extract the wavelet decomposition tree. At each iteration, the coefficients are sub-sampled by 2 and we have thus seven levels in the tree due to the initial size of the matrix.

Once the wavelet coefficients for electrostatic potentials are calculated, similarity analysis is performed on each protein datasets by calculating the distances between each pair with different MRA-based distance measures. Three MRA-based distance measures mentioned in the previous chapter are compared against their corresponding regular form. Remember that the regular measures computes the distances using the raw electrostatic potential distribution.

The MRA-based measures include:

$$WCB_{AB} = \left(\sum_{l=1}^{m} w_l\right)^{-1} \cdot \sum_{l=1}^{m} w_l \frac{1}{N_l} \sqrt{\frac{1}{2} - \frac{\sum\limits_{i,j,k} \varphi_A^l(i,j,k) \cdot \varphi_B^l(i,j,k)}{2 \cdot \sqrt{\sum\limits_{i,j,k} \varphi_A^l(i,j,k)^2} \cdot \sqrt{\sum\limits_{i,j,k} \varphi_B^l(i,j,k)^2}}}$$

$$WHD_{AB} = \left(\sum_{l=1}^{m} w_l\right)^{-1} \cdot \sum_{l=1}^{m} \left(w_l \frac{1}{N_l} \sqrt{\frac{1}{2} - \frac{\sum\limits_{i,j,k} \varphi_A^l(i,j,k) \cdot \varphi_B^l(i,j,k)}{\sum\limits_{i,j,k} \varphi_A^l(i,j,k)^2 + \sum\limits_{i,j,k} \varphi_B^l(i,j,k)^2}} \right)$$

$$WLN_{AB} = \left(\sum_{l=1}^{m} w_l\right)^{-1} \sum_{l=1}^{m} w_l \frac{1}{N_l} \sum_{i,j,k} \frac{\left|\varphi_A^l(i,j,k) - \varphi_B^l(i,j,k)\right|}{\max(\left|\varphi_A^l(i,j,k)\right|, \left|\varphi_B^l(i,j,k)\right|)}$$

The original distance measures are defined as following:

$$CB_{AB} = \sqrt{\frac{1}{2} - \frac{\int \phi_A \phi_B dt}{2 \cdot (\int \phi_A^2 dt)^{\frac{1}{2}} \cdot (\int \phi_B^2 dt)^{\frac{1}{2}}}}$$

$$WHD_{AB} = \sqrt{\frac{1}{2} - \frac{\int \phi_A \phi_B dt}{\int \phi_A^2 dt + \int \phi_B^2 dt}}$$

$$WLN_{AB} = \frac{1}{2N} \sum_{i,j,k} \frac{\left|\phi_A(i,j,k) - \phi_B(i,j,k)\right|}{\max(\left|\phi_A(i,j,k)\right|, \left|\phi_B(i,j,k)\right|)}$$

The wavelet-based measures use the wavelet coefficients (φ) to compute the similarity value, whereas the regular distance measures use the original electrostatic potentials (ϕ) for the same purpose. In wavelet-based similarity measures, the weights were chosen as *k*th power of 2 for the *k*th decomposition level. As you may have noticed, the regular distance measures here are normalized so that the resulting similarity ranges from 0 meaning exactly the same to 1 meaning exactly different. Custom Python and Matlab scripts were implemented to generate separate distance matrix for each protein dataset and distance measure combination.

6.2.2 Clustering Analysis

Using the distance matrix extracted during similarity analysis as input, the agglomerative hierarchical clustering algorithm with average linkage was applied to cluster similar proteins together. The algorithm creates a hierarchy of proteins which is represented as a dendrogram. Therefore, it is necessary to apply a tree cutting procedure to find the final clusters. The criterion for tree cutting in our experiments was determined by cluster function in Matlab. The function supports two cutoff settings to construct clustering from the dendrogram, which are height and inconsistency coefficient. The height criterion uses the distance value indicated as x-axis in the dendrogram. All nodes below than a height threshold are grouped into a cluster. The second criterion, the inconsistency coefficient, indicates how consistent a link is compared to the other links at the same level in the tree. The value statistically compares the height of a link in a cluster hierarchy with the average height of links below it. The algorithm finds an inconsistency value for each link present in the tree. By changing the height or inconsistency coefficient at each step, it scans through the dendrogram and cuts the tree into clusters. For each cutoff criterion, it analyzes the clustering using ground truth provided for each dataset and calculates an accuracy score. The clustering with the best score was then reported back for accuracy analysis.

Almost all clustering analysis measures assume that the class labels for each protein should be known prior to the analysis. Therefore, first step is to obtain class labels using the known similarities in physicochemical properties of proteins. C3d/Efb-c dataset consists of C3d and Efb-c proteins that are mutated via alanine scanning. The main goal of the alanine scanning is to identify the importance of an ionizable amino acid in the protein. Since alanine is a neutral amino acid, it usually doesn't affect the tertiary structure and replacing it with an ionizable amino acid cause either enhancement or disruption in functionality. The enhancement or disruption in a protein-protein interaction can be measured in terms of association free energy. Note that this measure is usually applicable when the structures of all proteins participating in a complex are known in advance. Electrostatic free energies were calculated by APBS according to a thermodynamic cycle. The change in electrostatic free energy of C3d/Efb-c association is calculated according to

$$\Delta\Delta G_{assoc}^{vacuum} = \Delta G_{C3d/Efb-c}^{vacuum} - \Delta G_{C3d}^{vacuum} - \Delta G_{Efb-c}^{vacuum}$$

where $\Delta G_{protein}^{vacuum}$ is the measure of unstableness, which is higher in protein's unfolded state. In mutant comparisons, the free energies higher than the parent protein indicate enhancement of functionality. If the free energy is less than the parent, the mutant is less stable than the parent implying a disruption in the association. In that respect, the Efb-c and C3d mutants are categorized into two classes based on their affect on the interaction: enhancing and inhibitory. Factor H dataset, on the contrary, does not assume any interaction among the proteins. In absence of interactions, we can use the charge composition of the protein surface as reference to form ground truth for the clustering analysis. Even though this approach is not as accurate as the free energy, the method will still aid in understanding the functionality of modules since the surface charges are one of the significant forces that drive complement interactions. The method partitions the CCP modules into three groups: positive, negative and grey zone.

Expressing the clustering performance quantitatively is usually a critical step in clustering analysis. Our method employs a scoring function that is originated in classification problem of data mining. The classification assigns a specific class label to each instance while clustering identifies a set of correlated instances which may form a class. As a matter of course, we have the actual class labels for each instance in classification while having only group labels in clustering. The classification compares the class labels estimated by the classification algorithm to the original ones and measures the number of the correct assignments for accuracy calculation. As mentioned earlier, each protein in a dataset is associated with a class label which was determined based on its physicochemical properties. However, the clustering algorithm returns only group labels to which proteins are assigned, -which may not be necessarily same as the class label. Since the relationship between actual class labels and estimated cluster labels is unknown, we define a measure that is capable of extracting the accuracy from only the group information.

The measure of clustering accuracy in our clustering analysis is the *pair-wise relationship conservation*. The measure calculates the percentage of the relationships conserved by the clustering algorithm to the number of all relationships in the dataset. Let l_i be the class label for instance i. Assume we know the original classes to which each instance belongs. From the clustering perspective, we have two types of relationship for each pair $\rho(i, j)$ of instances:

1. They both belong to the same cluster ($l_i = l_j$)
2. They belong to the different clusters ($l_i \neq l_j$)

The clustering algorithm assigns a clustering label to each instance, let's say l_i', which may be different from the original one. However, we may safely assume that if the estimated clusters are correct in respect to the original grouping, the relationships should be still the same for the pairs:

1. if $l_i = l_j$, then $l_i' = l_j$
2. if $l_i \neq l_j$, then $l_i' \neq l_j$

The measure is expressed as follows:

$$ClusterAccuracy = \frac{\sum_{i>j}^{N} S(i,j) + \sum_{i>j}^{N} D(i,j)}{\frac{1}{2} \cdot N(N-1)}$$

where

$$S(i,j) = (l_i = l_j) \text{ AND } (l_i' = l_j')$$

$$D(i,j) = (l_i \neq l_j) \text{ AND } (l_i' \neq l_j')$$

If the estimated clustering is the same as the clusters in the original dataset, the metric will return 1. Otherwise, each lost relationship degrades the clustering accuracy. The measure computes the accuracy as a numeric value whose range is [0,1].

6.3 **Experimental Results**

6.3.1 Efb-c and C3d Mutants Dataset

6.3.1.1 Overview

In order to design therapeutic drugs to eliminate infections, we must analyze the molecular interactions between the complement and pathogenic proteins. In the first set of experiments, we examine the effectiveness of our wavelet-based similarity measures in comparing mutant proteins that are generated via alanine scanning and understanding the functional similarity between each mutant pair. C3d/Efb-c interaction is an excellent candidate for such analysis since the association provides means of intrusion into the host body for the bacteria. Efb is a surface protein that is produced and released by the *staphylococcus aureus* bacteria. The Efb protein binds to the C3d complement protein and prevents the activation of immune response by interrupting CR2/C3d interaction. CR2/C3d interaction plays an important role in activation and maturation of B-cells of immune system. Due to the interruption, the bacteria pass through the surveillance of the immune system without being caught and cause infection and inflammatory response.

Recent studies drew attention to importance of electrostatics in the C3d/Efb-c association. Electrostatic analysis in conjunction with alanine scanning may help identify the significant ionizable residues that influence the protein complex formation. The drugs targeting the important residues necessary for binding may interfere the C3d/Efb-c association and eliminate the virulence mechanism of the Efb-c pathogen protein. We will analyze the electrostatic properties of the Efb-c and C3d mutants using six different similarity measures and seek which method can correlate the electrostatic properties of the mutant to its physicochemical properties to the best.

6.3.1.2 Efb-c Clustering

Figure 6.2 and 6.3, respectively, compares the performance of the non-wavelet and wavelet based similarity measures in clustering Efb dataset. The clustering of the Efb-c mutants is based on their similarity in the spatial distribution of their electrostatic potentials. In each experiment, we seek to cluster inhibitor and enhancer mutants into two separate groups using only electrostatic potential information. The class labels for each mutant were determined in advance based on the association free energies. The mutations that release less free energy than the parent protein are classified as inhibitory. In contrast, the mutations which increase the stability of the association by releasing more

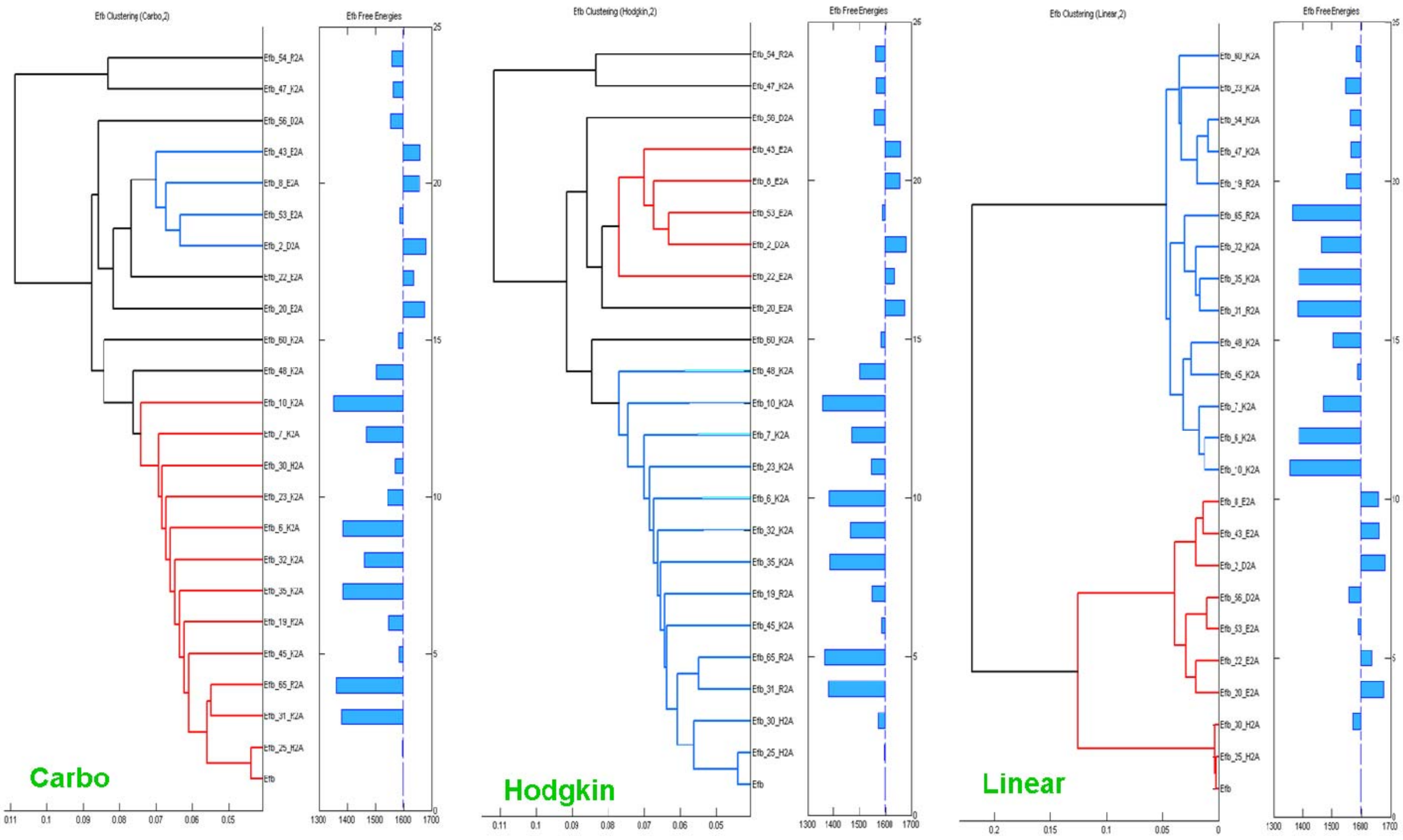


Figure 6-2. Clustering diagram of the 24 mutants of Efb-c protein. Electrostatic potentials were calculated using ionic strengths corres-ponding to 0 mM counterion concentration. The similarity matrix was calculated using CB, HD and LN measures.

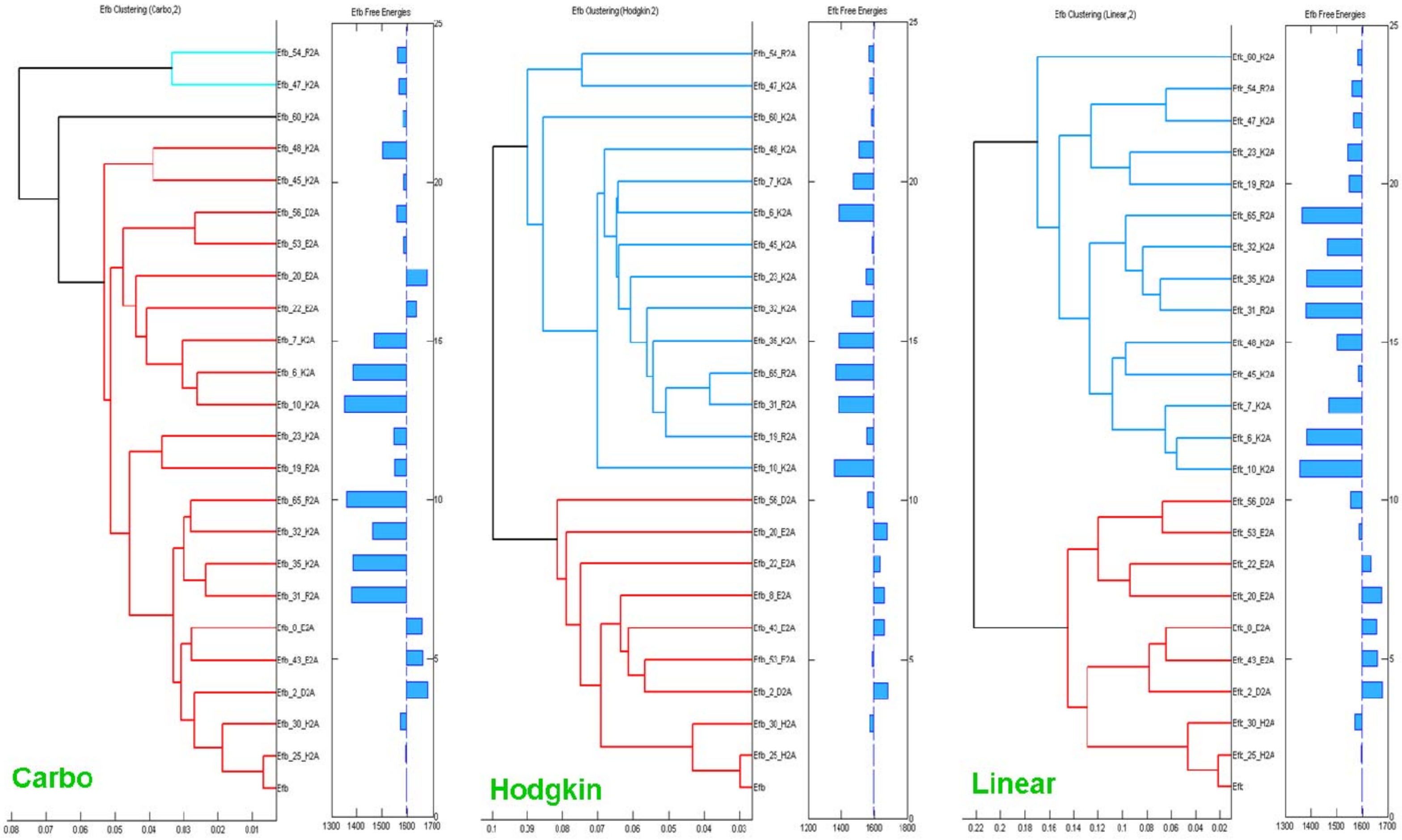


Figure 6-3. Clustering diagram of the 24 mutants of Efb-c protein.Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counterion concentration. The similarity matrix was calculated using WCB, WHD and WLN measures.

energy are classified as enhancing mutation. Based on this classification, we have 6 enhancing and 18 inhibitory proteins in the Efb-c dataset.

Since the Efb-c is excessively positive charged (+7e), we predict the positive mutants to reduce the association free energy and stability while the negative mutants to increase them. The positive mutants in the dataset are arginine-to-alanine (R2A) and lysine-to-alanine (K2A). The negative mutants are glutamic acid-to-alanine (E2A) and aspartic acid-to-alanine (D2A). In addition, we have histidine mutations (H2A) or neutral mutations whose contribution to the association is negligibly small. Thus, we expect the histidine mutations to cluster with the parent protein and to exhibit almost identical physiological characteristics.

The relationships between the electrostatic potentials, as well as the corresponding free energies, of 24 Efb-c mutants were depicted in Figure 25 and 26. Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counter ion concentration. The dendrograms in Figure 6.2 were calculated by using the agglomerative clustering algorithm in conjunction with Carbo (CB), Hodgkin (HD) and Linear (LN) distance measures, respectively. Figure 6.3 demonstrates the dendrograms generated by using the wavelet-based Carbo (WCB), wavelet-based Hodgkin (WHD) and wavelet-based Linear (WLN) distance measures. Figure 6.8 provides the accuracy ratios for final clusters, calculated by applying six methods and cutting the trees with height criterion. Figure 6.9 shows the corresponding accuracy ratios when inconsistency coefficient was used to generate the final clusters. Glancing at both tables, we observe that the inconsistency coefficient is a better criterion for clustering the Efb-c mutants. Thus, the discussions will refer to the results in Figure 6.8.

The dendrograms generated by using the Carbo and Hodgkin measure were quite comparable, producing almost identical clusters. Although these measures managed to cluster the inhibitory and enhancing mutations partially, they are not very promising in finding the correlation between the electrostatic potentials and the complex stability with high accuracy. The Linear similarity measure outperformed them by achieving 72% accuracy, while Hodgkin and Carbo are both stuck at an accuracy of only 61%.

The clustering results based on wavelet-based similarity measures suggests that the wavelet domain is more favorable in comparing the electrostatic potentials and finding their connection with the physicochemical properties, as seen in Figure 6.3. While WCB and WLN were about the same as CB and LN respectively, WHD measure managed to increase the cluster quality by 10% over HD measure. In our analysis, LN, WHD and WLN measures outperformed the other three with the maximum accuracy of 71%. The reason why the accuracy does not exceed 71% is that D56A, E53A, H30A and H25A mutations tend to decrease the association free energy even though they are classified as neutral or negative mutations and physiologically supposed to maintain or increase the free energy.

In free energy analysis of C3d/Efb-c complex, we observe that not all mutations equally affect the association. The mutations whose free energies change less or more than 50kJ/mol are considered as important residues affecting the association stability.

These residues are known to be on the association interface with C3d. Based on free energy analysis, K10A, K6A, K7A, R31A, K35A, K32A, R65A represents the most inhibitory mutations for the formation. In addition, D2A, E8A, E20A and E43A residues are observed as the most important mutations that enhance the stability of the protein complex. In the dendrograms, LN and W-LN successfully clustered the most inhibitory and enhancing mutations together while other methods have failed to identify such patterns.

6.3.1.3 C3d Clustering

We have performed electrostatic analysis on C3d fragment of C3d/Efb-c complex for completeness. Similar to the Efb dataset, C3d mutants are categorized into enhancing and inhibitory classes according to the a priori knowledge on the association free energies. Efb-c and C3d have opposite excess charges and Efb-c protein is believed to bind to the acidic pocket of C3d. An increase in the magnitude of the negative excess charge on C3d component can potentially increase the interaction between two components. As a result, we expect the positive mutations to enhance the binding ability and the negative mutations to disrupt the binding ability, which is the opposite of Efb mutants. Within this context, 33 out of 66 mutants in the dataset exhibit inhibitory characteristics while the rest cause enhancement in the binding.

In Figure 6.4, which shows the clustering on C3d mutants using the non-wavelet similarity measure, we see that clustering results with CB and HD measures were not optimal. Notice that the dendrograms display a cascading structure. The situation is due

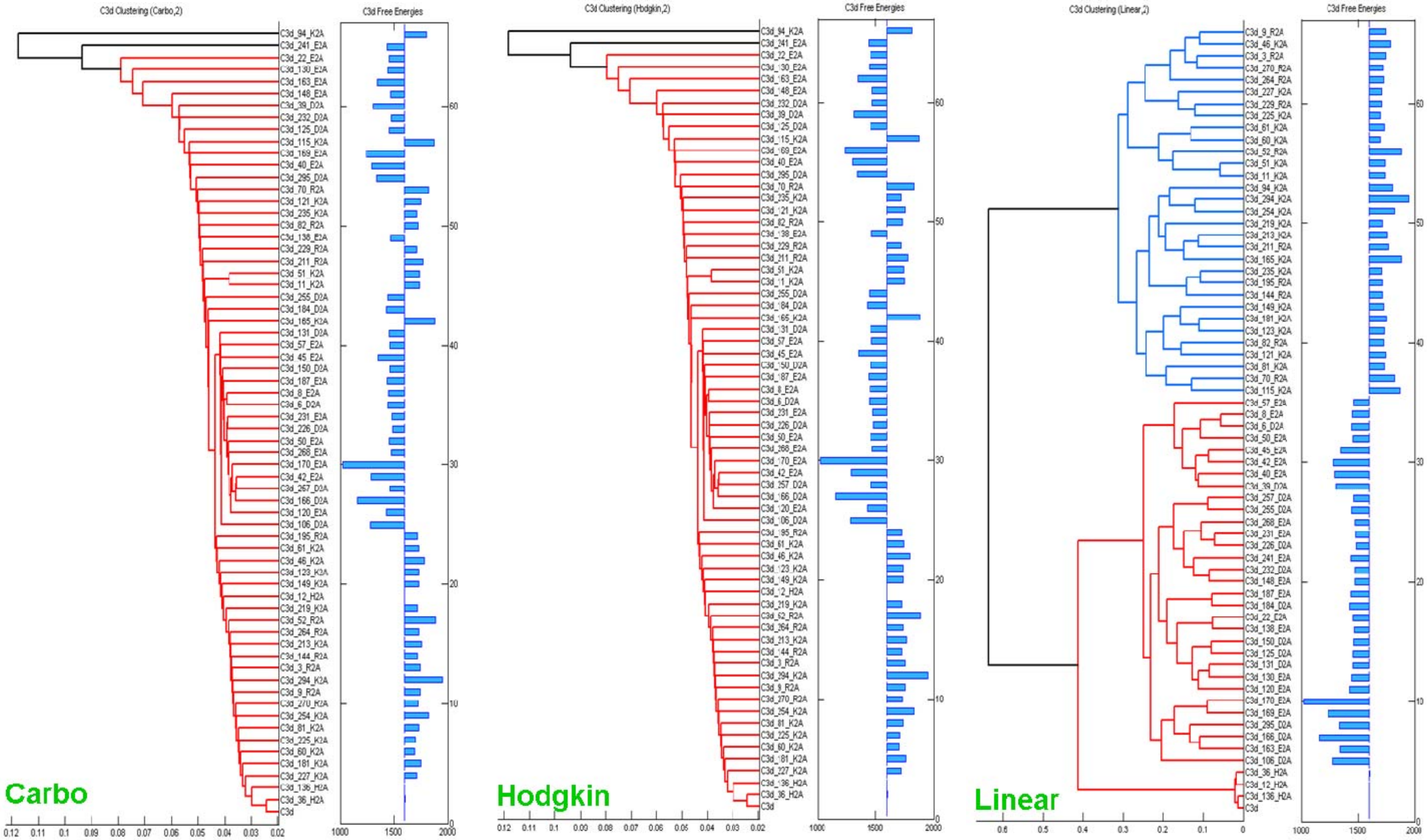


Figure 6-4. Clustering diagram of the 66 mutants of C3d protein.Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counterion concentration. The similarity matrix was calculated using CB, HD and LN measures.

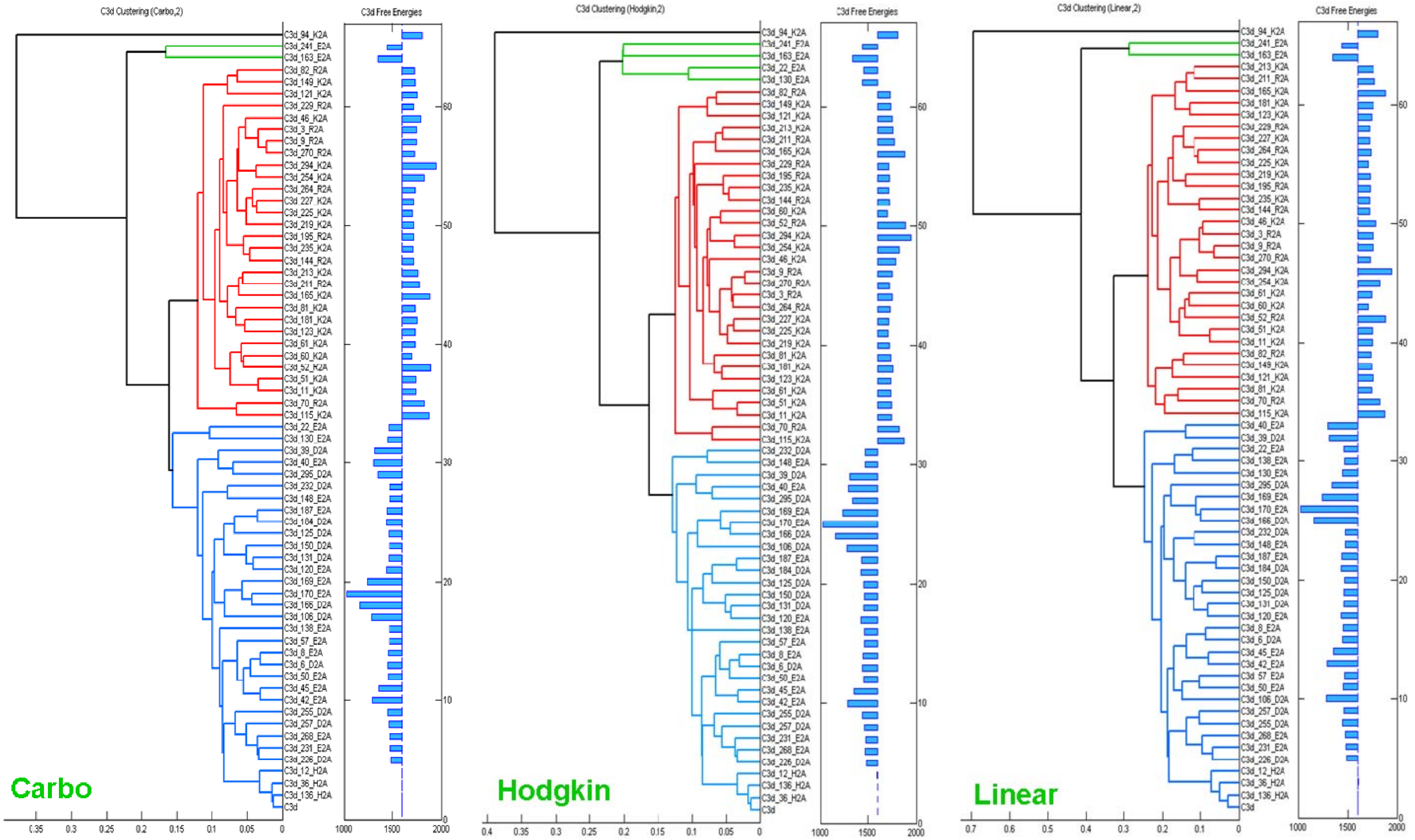


Figure 6-5. Clustering diagram of the 66 mutants of C3d protein.Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counterion concentration. The similarity matrix was calculated using WCB, WHD and WLN measures.

to the large electrostatic potentials observed inside the mutants. The quadratic nature of these measures favors the large values in distance calculation. The negative effect of large values is considerably obvious in large proteins like C3d. Clustering algorithm joined with LN measure, on the contrary, managed to recognize all C3d clusters except for the histidine mutations. In quantitative analysis of the non-wavelet similarity measures on C3d mutants, choosing inconsistency coefficient as tree cutoff criterion demonstrated 16% more accurate results in HD and CB clustering. With 94% overall accuracy, LN clustering outperformed the others.

The wavelet-based clustering has shown in Figure 6.5 that the quality of CB and HD clustering could be significantly increased in the wavelet domain. We observed 27% and 24% better clustering accuracy for WCB and WHD clustering respectively. In the dendrograms, negative and positive mutations are all clustered together properly, except for the E241A, E163A, K94A, E22A and E130A mutations. The situation implies that using wavelet-based similarity measures lessens the negative effect of large values on CB and HD clustering. In contrast, there was 2% drop in WLN clustering accuracy compared to the LN clustering. Considering the increase in WCB and WHD clustering, this amount is relatively negligible. In the tree cutting routine here, the height criterion dominated the inconsistency coefficient and increased both the precision and accuracy in clustering results.

Besides achieving high quality clusters, we anticipate that the wavelet-based clustering algorithm may assist to identify the important ionizable C3d residues which favorably contribute to the C3d/Efb-c binding. So long as it does not interfere with associations of other immune proteins with C3d, these residues are possible targets for drug design. In the C3d analysis, we only consider those residues that have a considerable effect on the association free energy. The threshold for the effect is defined as >250 kJ/mol because of the large molecular size of C3d component. Based on our free energy threshold, there were nine noteworthy inhibitory mutations: D106A, E163A, D166A, E169A, E170A, D295A, D39A, E40A, and E42A. In the dendrograms generated by using WCB, WHD, WLN and LN, these inhibitory mutants were observed to be very close to each other, which were expected. We have also identified four important enhancing mutations which are K165A, K115A, K294A, and R52A. Even though these mutants had similar free energy values, they were typically located away from each other in the dendrograms. The reason is believed to be due to their distance to each other in the protein structure.

6.3.2 Factor H Modules Dataset

6.3.2.1 Overview

Factor H has a chain-like structure consisting of 20 complement control protein modules. The charge diversity of the CCP modules permits a variety of interactions with immune system proteins. In absence of this essential regulator, the immune system cannot distinguish self from non-self and complement proteins attack to the self-tissues aside from pathogen cells. In addition, mutations at functional sites of Factor H result in a variety of diseases such as Age-related Macular Degeneration (AMD) and Atypical

Hemolytic Uremic Syndrome (aHUS). In this section, we present a comparative electrostatic analysis of CCP modules which may guide future studies on Factor H.

Currently, the structures of only 11 CCP modules are available through experimentation, against 9 CCP modules still lacking structures. The experimentally determined structures are: CCP1-3, 5-8, 15-16, and 19-20. In this study, we have obtained the structures of the rest by using homology models. If an amino acid sequence of an unknown structure has more than 30% identity to the sequence of a known structure, it is highly probably that they have a similar tertiary structure. The rationale of this statement is that less than 15% of structures deposited into the protein databases in recent years are considered as new folds[]. Because of the structural and sequential

CCP Module	**CCP Template**	**Sequence identity**
CCP4	CCP5 (Online)	32%
CCP9	CCP7 (2UWN)	35%
CCP10	CCP16 (1HCC)	27%
CCP11	CCP19 (2BZM)	34%
CCP12	CCP16 (1HCC)	33%
CCP13	CCP15 (1HFI)	17%
CCP14	CCP15 (1HFI)	37%
CCP17	CCP16 (1HCC)	31%
CCP18	CCP19 (2BZM)	40%

Table 6-1. Template CCP modules and corresponding coordinate (PDB) files used for homology modeling of the individual Factor H modules.

similarity among CCP modules, the homology modeling is a reasonable method for obtaining the unknown structures computationally. The template modules used for homology modeling are described in Table 2. After all CCP modules were obtained, the structures were superimposed based on C-atom using the CCP16 module as reference and their electrostatic potentials were calculated using APBS. Then, we have performed a comparative analysis to classify similarities and dissimilarities of the spatial distributions of elect static potentials of the CCP modules. In the following section, we shall see the results of this analysis.

6.3.2.2 ***Factor H Clustering***

Figure 6.6 and 6.7 presents the clustering of the spatial distribution of electrostatic potentials of 20 CCP modules with non-wavelet and wavelet-based clustering algorithms, respectively. To quantitatively assess the performance of the methods, the ground truth for the Factor H dataset was to be obtained. The excess charge of each module is responsible for driving the Factor H interactions and thus can be used to determine the family of the module. In this respect, the individual CCP modules were divided into three small families prior to the clustering:

1. *Negative Modules:* The family consists of nine CCP modules whose excess charge varies between -6 and -2: CCP2, 3, 6, 9, 11,12, 14, 15, and 16
2. *Neutral Modules:* These modules are assumed to be in the gray zone since their excess charge ranges from -1 to +1: CCP4, 8, 10, 17, 18, and 19

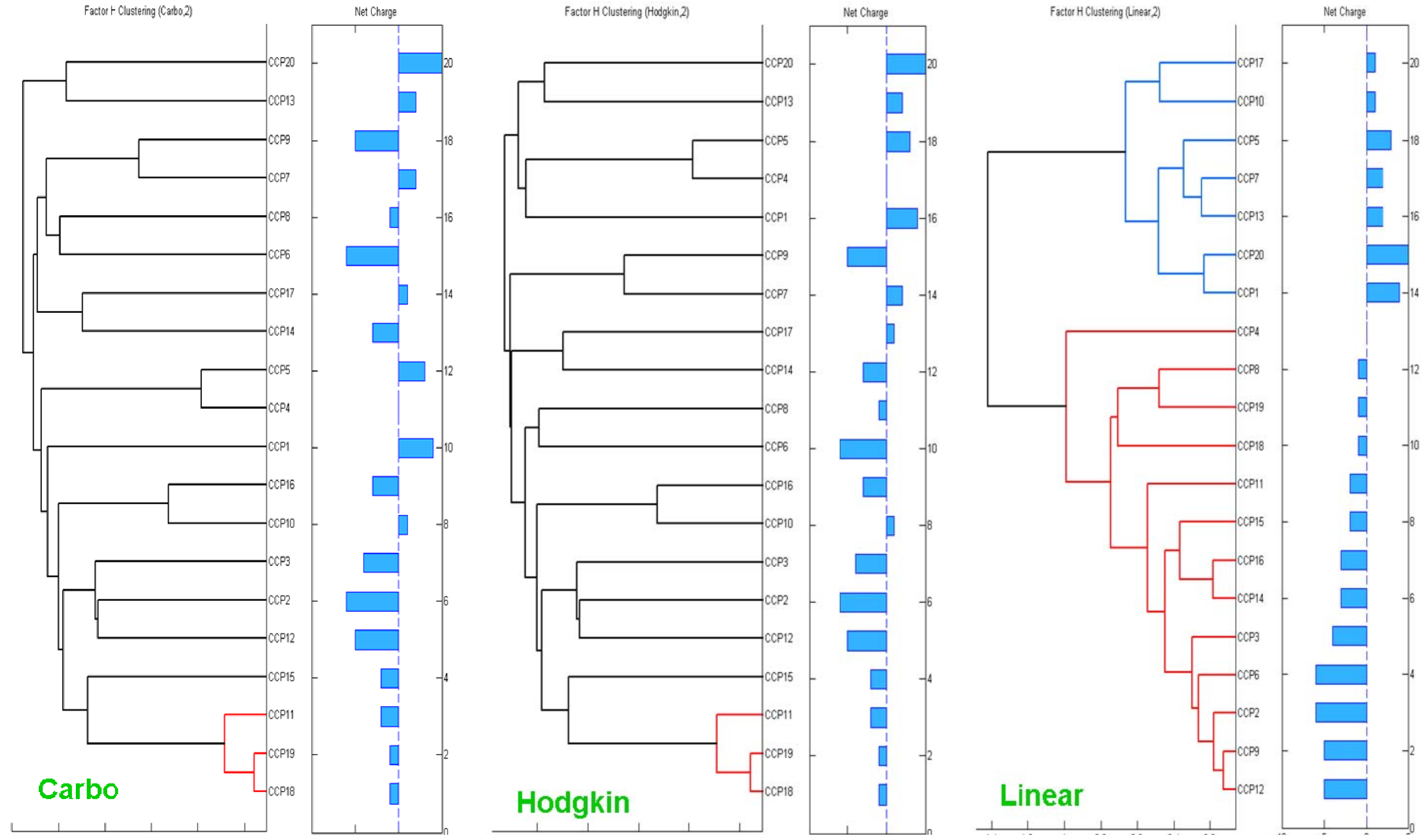


Figure 6-6. Clustering diagram of the 20 CCP modules of FH. Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counterion concentration. The similarity matrix was calculated using CB, HD and LN measures.

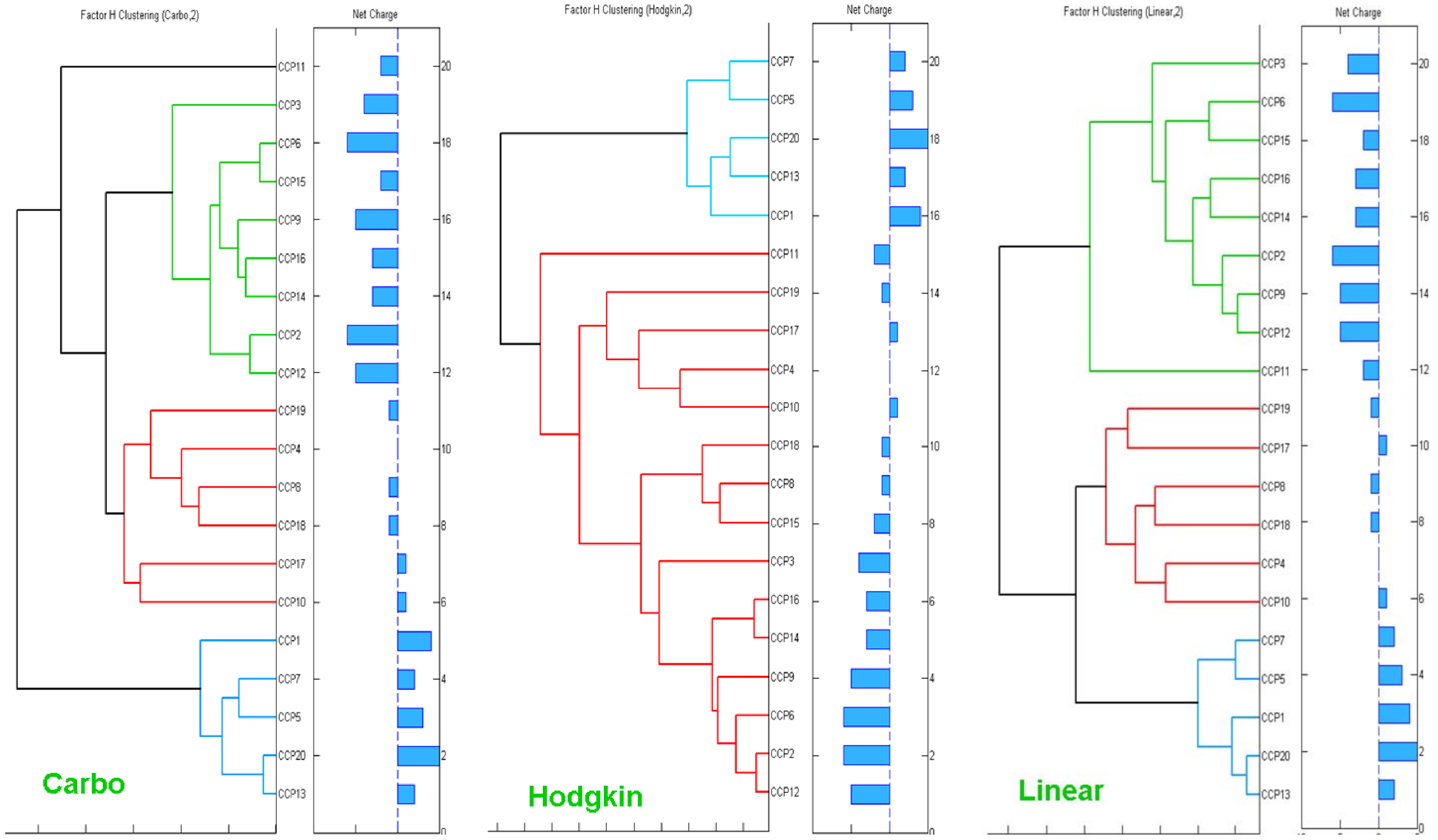


Figure 6-7. : Clustering diagram of the 20 CCP modules of Factor H.Electrostatic potentials were calculated using ionic strengths corresponding to 0 mM counterion concentration. The similarity matrix was calculated using WCB, WHD and WLN **measures.**

3. *Positive Modules:* This family consists of five modules whose excess charge varies between +2 and +5: CCP1, 5, 7, 13 and 20.

Figure 6.6 show that clustering algorithm with CB and HD measure cannot cluster the CCP modules properly as seen in the fuzzy structure of the dendrogram. The fuzzy structure is believed to be due to the quadratic nature of the CB and HD measures. In the Figure 6.9, it is evident that the HD measure does not suffer as much as the CB measure in distinguishing the modules owing to its normalization factor. In comparison, the clustering algorithm cooperating with LN measure performed much better in identifying the clusters. While the clustering results may be acceptable, the neutral modules were not recognized as a separate cluster; rather they were clustered with the closest positive or negative modules.

The clusters generated by using wavelet-based similarity measures depict more encouraging results. From first glance, it is apparent that all CCP families were identified more clearly in the cluster tree. The quantitative analysis also supports the validity of the clustering results based on the wavelet approach. Although CB clustering had the lowest quality for the clusters among all methods, WCB presented the best results with 92% accuracy when the tree was cut by using the inconsistency coefficient as seen in Figure 6.8. In addition, the WLN clustering has succeeded to find the optimal clustering of the modules when the height criterion was used. While WHD clustering performed poorly against other wavelet-based methods, it was still superior to the non-wavelet based clustering methods.

6.4 **Conclusion**

In this chapter, we have performed theoretical calculations for a set of computationally obtained complement-related proteins to investigate the effectiveness of wavelet-based similarity measure in practice, specifically on C3d, Efb-c and Factor H proteins. We have carried out a comparative and quantitative analyses of wavelet and non-wavelet similarity measures by applying a hierarchical clustering method. The proteins in each dataset were partitioned into small families in advance to generate a ground truth which was later used to the measure the clustering quality of each method. In generation of families, we have used the association free energies and excess charges, two characteristics chiefly affecting the binding ability of the proteins. In our quantitative analysis, the wavelet-based similarity measures outperformed the non-wavelet similarity measures by presenting up to 45% better clustering quality in some experiments. Additionally, the proposed methods managed to identify the significant mutations in Efb and C3d datasets and cluster them together more precisely in contrast to the non-wavelet similarity measures. These results suggest that wavelet-based methods are more effective and promising in electrostatic potential similarity calculations, and thus identifying the physicochemical similarities among proteins.

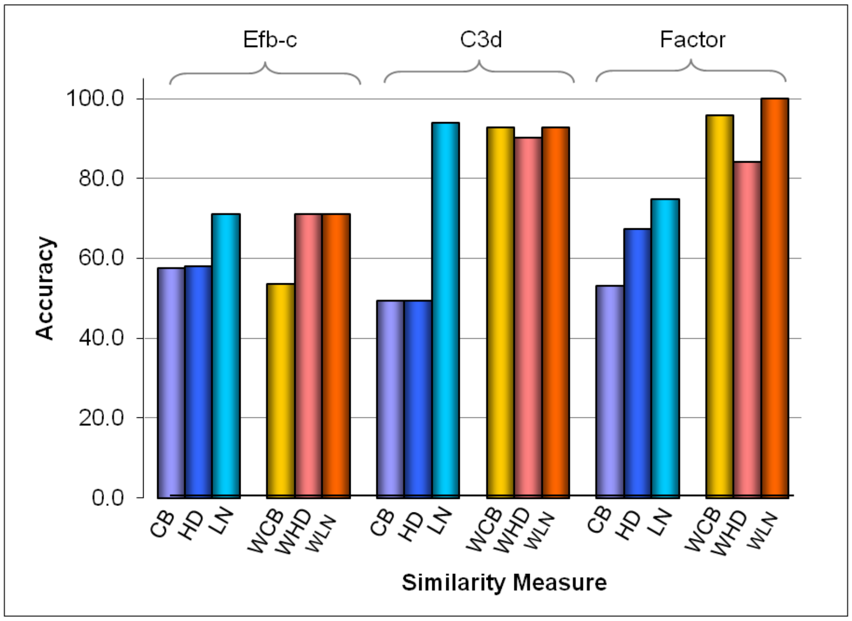


Figure 6-8. The performance comparison of wavelet and non-wavelet clustering methods using the height criterion.

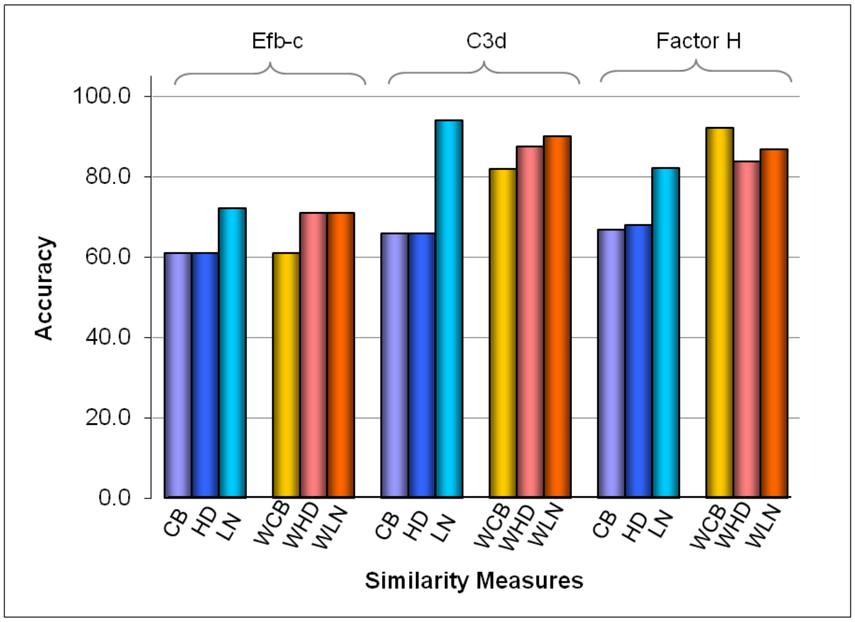
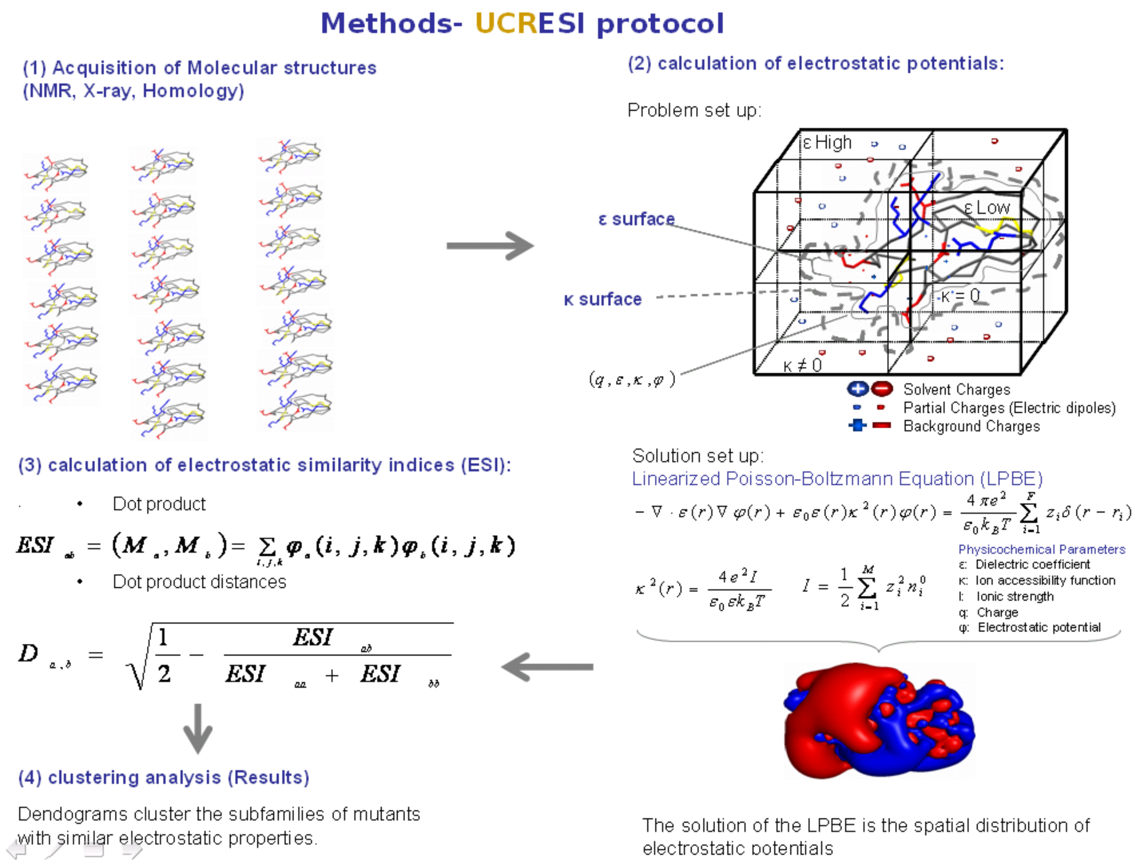


Figure 6-9. The performance comparison of wavelet and non-wavelet clustering methods using the inconsistency coefficient criterion.



Methods- UCRESI protocol

(1) Acquisition of Molecular structures (NMR, X-ray, Homology)

(2) calculation of electrostatic potentials:

Problem set up:

🔵 🔴 Solvent Charges
● ● Partial Charges (Electric dipoles)
✚ ▬ Background Charges

Solution set up:

Linearized Poisson-Boltzmann Equation (LPBE)

$$-\nabla \cdot \varepsilon(r)\nabla\varphi(r) + \varepsilon_0\varepsilon(r)\kappa^2(r)\varphi(r) = \frac{4\pi e^2}{\varepsilon_0 k_B T}\sum_{i=1}^{F} z_i\delta(r - r_i)$$

$$\kappa^2(r) = \frac{4e^2 I}{\varepsilon_0 \varepsilon k_B T} \qquad I = \frac{1}{2}\sum_{i=1}^{M} z_i^2 n_i^0$$

Physicochemical Parameters
ε: Dielectric coefficient
κ: Ion accessibility function
I: Ionic strength
q: Charge
φ: Electrostatic potential

The solution of the LPBE is the spatial distribution of electrostatic potentials

(3) calculation of electrostatic similarity indices (ESI):

- Dot product

$$ESI_{ab} = (M_a, M_b) = \sum_{i,j,k}\varphi_a(i,j,k)\varphi_b(i,j,k)$$

- Dot product distances

$$D_{a,b} = \sqrt{\frac{1}{2} - \frac{ESI_{ab}}{ESI_{aa} + ESI_{bb}}}$$

(4) clustering analysis (Results)

Dendograms cluster the subfamilies of mutants with similar electrostatic properties.

Figure 6-10. The protocol that was used to perform the similarity analysis

Chapter 7

Conclusions

In this dissertation, we have analyzed many aspects of locality and its applications to real datasets. We began with covering the issues related to how semi-supervised clustering can incorporate magnetically affected paths in order to achieve better accuracy. Additionally, we have proposed efficient and effective data mining methods for molecular similarity analysis. We have defined several locality patterns commonly observed in electrostatic potential distributions of biological molecules and explained how wavelet transformation can be utilized in similarity functions so as to recognize these patterns. This approach has shown promising results in capturing the correlations among proteins, especially the ones sharing the same ancestor.

In this chapter, we summarize the dissertation briefly, discuss its contributions, and suggest directions for future research.

7.1 **Contributions**

The most important contributions are summarized below:

1. ***Magnetically Affected Paths*: ** We have presented a semi-supervised concept that locally manipulates the edges of a graph structure in order to aid the distance measure. We have named this concept MAP, as an acronym of the words "Magnetically Affected Paths," due to the fact that it simulates the electromagnetic field characteristics in a graph structure and adjusts the shortest paths between the objects based on user constraints. The basis for selecting graphs as a natural target of the concept lies in the ease of implementation in a graph domain, rather than in a Cartesian space. The must-link and cannot-link constraints are expressed as special edges which exert a force on regular edges. The regular edges resonate with special edges and imitate their characteristics so that the objects in the vicinity of positive (must-link) edges get closer to each other, whereas the objects in the vicinity of negative (cannot-link) edges get away from each other. The impact factor of a constraint on an object is determined by the distance and alignment of the object relative to the constraint edge.
2. ***MAPClus Framework*: ** One important goal of this thesis was to develop a flexible clustering algorithm with the capability of partitioning both vector and graph data. We have accomplished this goal by integrating the MAP concept into a semi-supervised clustering framework. The MAPClus framework implements a 3-step model in order to perform clustering. First, it converts the vector data into a

graph structure such that the distance relationships between objects are preserved. The algorithm runs a k-nearest neighbor search for each object in the dataset and connects the objects in certain vicinity together to construct the graph. If the data is already given as a graph structure, it skips the graph construction routine. Second, the algorithm adjusts the edge weights using the probabilistic insight provided by constraint edges. Once all edges are adjusted according to their proximity to the constraint edges, it runs all pairs in the *k*-shortest path algorithm to extract the distance matrix. Even though a single path was enough to capture acceptable distances, the experimental results suggest that the distances become more accurate when multiple shortest paths are used. Finally, the framework runs a graph-compatible clustering algorithm to find the clusters. We have optimized the framework implementation in several ways for better time efficiency. First of all, we have extended the Dijkstra's shortest path algorithm to find multiple shortest paths between any pair of nodes. In addition, we have introduced a divide-and-conquer algorithm which partitions the graph into equal-sized sub-graphs, calculates the *k*-shortest path distances locally, and then merges them back together to extract the global distances. Although we managed to speed up the clustering algorithm to some degree, it was not enough for the algorithm to compete with state-of-the-art algorithms, such as GraClus and MPCK-Means, in terms of efficiency. Thus, we have implemented a multilevel version of the framework which has almost the same performance as the standard K-Means algorithm and the same or better accuracy as the other clustering algorithms with which it was compared.

3. ***Formalization of Locality Patterns***: A family of locality patterns which are usually observed in electrostatic potential distributions of proteins have been defined to assist the similarity analysis of two proteins in a quantitative way. We have also given several scenarios where each pattern can usually be observed. We have discussed the patterns under three headings: proportionality, displacement and scaling. (i) The *proportionality pattern* indicates the changes in the electrostatic potential magnitude. For example, an increase or a decrease in pH value in vivo will make bio molecules have a more negative or a more positive net charge [96]. The ratio between the initial and final values of the electrostatic potentials can be expressed as a proportionality pattern. This pattern plays a significant role in protein recognition because the bioactivities among proteins are mostly driven by the diversity of charges. (ii) The *displacement pattern* reveals the particular regions which exist in both proteins but at different locations. Due to the preservation of the carbon backbone structure, the pattern is usually measured in terms of a displacement angle where the origin is located at the center of the protein. The empirical evaluations suggest that this pattern is of paramount importance to molecular similarity analysis, especially in analysis of homology modeling [97] where the proteins are derived from a common ancestor. (iii) Finally, a *scaling pattern* is commonly encountered in electrostatic potential

distributions when an expansion or shrinkage is observed in the area of a particular region without any other characteristic change. This pattern usually happens as a result of a mutation such as when a hydrophobic amino acid is substituted with a hydrophilic amino acid or a specific amino acid is substituted with an amino acid whose net charge is different from the original's charge [97].

4. ***Wavelet Transformation-Based Similarity Indices***: Our earlier experiments suggested that the state-of-the-art molecular similarity measures could not recognize the locality patterns and thus could not take them into account in similarity calculations. Hence, we have proposed *WCB*, *WHD* and *WLN* similarity measures which are capable of discriminating between the locality patterns. Unlike conventional methods, these measures apply a three-dimensional wavelet transformation on the electrostatic potential distribution of proteins to find the corresponding wavelet coefficients. Then, they determine the similarity using the wavelet coefficients. This approach exploits the ability of wavelet transformation to analyze the spectral components of an electrostatic potential distribution and suggests a localized and more sensitive way of measuring the similarity. To the best of our knowledge, our similarity measures are the first of their kind which support true three-dimensional analysis in molecular informatics. We have generated several toy data models in which we isolated one locality pattern at a time in order to determine which methods were sensitive to each pattern. Our empirical evaluations suggested that WHB and WLN were able to recognize all

three patterns, whereas WCB could only discriminate between displacement and scaling patterns. Furthermore, the WLN similarity measure was more responsive to the patterns than the WHB and WCB measures in the experiments.

5. ***Analysis of C3d/Efb-c and Factor H proteins***: In addition to the toy models, we have applied our MRA-based similarity measures to three different complement protein datasets. These real datasets were obtained either via alanine scanning or homology modeling. In alanine scanning, we replace every single ionizable amino acid with alanine in order to determine the contribution of specific residues to a protein’s function. To demonstrate the effectiveness of our similarity measures in conjunction with alanine scanning, we have performed a systematic study on C3d and Efb-c mutant datasets. In the study, the goal was to determine whether a specific amino acid residue played an important role in the C3d/Efb-c association. Theoretically, each mutation would cause either an enhancement or an inhibition in the bioactivity. We determined the actual enhancing and inhibitory mutants experimentally using C3d/Efb-c association-free energies in advance. Then, we performed a hierarchical clustering to cluster the mutants into two groups, inhibitory and enhancing, using only electrostatic potential information. The experiments were conducted using both conventional and MRA-based similarity measures. According to the quantitative analysis of alanine mutant clustering, MRA-based methods provided up to 45% better clustering quality in contrast to the conventional methods. In homology modeling experiments, we have used our

measures in the task of recognizing the functionality of computationally generated protein structures. We have chosen the Factor H CCP module datasets to demonstrate the validity of these methods. The class labels of CCP modules were determined based on the net charge, which is an important identifier for molecular functionality just like association-free energy. Once again, the clustering algorithms with the MRA-based similarity measures achieved up to 40% better accuracy over the conventional methods. Also, the dendrograms generated by the hierarchical clustering provided meaningful cluster descriptions.

7.2 **Future Research**

We are currently extending the research described in this thesis in many possible ways. In this section, we highlight several important directions for future work and present our preliminary results.

The MAPClus algorithm may suffer severely from the usage of a quadratic distance function such as a Euclidean metric in a graph construction phase, especially when some features are more dominant than others. For example, the proline attribute in the Wine dataset [72] takes values between 278 and 1680, while the other attributes usually take values between 7.6 and 19.3 on average. When we calculated the k-nearest neighbors using the Euclidean distance, the low quality of the graph structure did not allow the algorithm to improve clustering results significantly. The same conditions were observed in the Forest dataset. In order to reduce this negative effect on clustering, we can utilize a local feature selection algorithm in graph construction, such as a Principal

Component Analysis [98] or a Locally Adaptive Metric[19], to determine appropriate feature weights for the Euclidean distance function and increase the quality of the graph.

During our evaluations, another problem we encountered was that the current stochastic model used for the readjustment routine may overestimate escalation and reduction ratios. The current model was established based on a simplified two-cluster data model which is similar to the one used in support vector machines [99]. By extending the current model to a multiple-cluster model with the insight provided by statistical analysis, we may increase the accuracy even more.

In addition to a graph domain, we have implemented a simple version of the MAPClus framework in a Cartesian space. The Cartesian implementation utilized the line of sight to determine the object pairs affected by some constraints. The preliminary results demonstrated a significant increase in accuracy; however, the time complexity was not very satisfactory when a vast amount of constraints were involved in the calculations. One solution we are working on is extracting a kernel matrix [100] based on the principals of the MAP concept and then applying this matrix to the distance or adjacency matrix.

For the MRA-based similarity analysis, we have realized that using different weight functions may result in completely different yet meaningful clusters. For example, in Factor H experiments, assigning higher weights to the 2nd and 3rd level coefficients increased the influence of structural similarity on the clustering. A similar statement was proposed by Qiu et al. [60] in their study of protein secondary structure prediction. They found that the lower resolution scales corresponded well to the α-helices and the connecting peptides. However, the importance of each coefficient level may differ based on the application. Thus, we have considered estimating the weight function automatically when provided with the appropriate user knowledge. The challenge here is that the user knowledge about data is very limited in biological applications and it is very hard to come up with a mechanism that calculates the weights with very limited knowledge.

Another interesting problem is to apply the MRA-based similarity measures to the different types of data. In our analysis, we have applied the methods to the electrostatic potential distributions calculated in a vacuum. It is well known that the proteins change their electrostatic characteristics in different environments, particularly based on the ionic strength of the medium [101]. We can use the locality-based similarity measures to investigate the influence of ionic strength on the protein activity. Following this line of research, we will also apply the methods to hydrophobicity [102] and molecular dynamics [103].

Bibliography

[1] K.D. Borne, “Astroinformatics: A Data-Oriented Approach to Astronomy,” Louisiana, USA: 2009.

[2] D. Chivian, T. Robertson, R. Bonneau, and D. Baker, “Ab Initio Methods,” *METHODS OF BIOCHEMICAL ANALYSIS*, vol. 44, 2003, pp. 547-558.

[3] Jiawei Han and Micheline Kamber, *Data mining: concepts and techniques*, San Francisco: Elsevier Inc., 2006.

[4] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey, “Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections,” 1992, pp. 318–329.

[5] I. Dhillon, J. Fan, and Y. Guan, “Efficient Clustering of Very Large Document Collections,” *Data Mining for Scientific and Engineering Applications*, Kluwer Academic Publishers, 2001.

[6] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering techniques,” *KDD Workshop on Text Mining*, 2000.

[7] X. Xu, M. Ester, H. Kriegel, and J. Sander, “A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases,” *Proceedings of the Fourteenth International Conference on Data Engineering*, IEEE Computer Society, 1998, pp. 324-331.

[8] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” *Readings in database systems (3rd ed.)*, Morgan Kaufmann Publishers Inc., 1998, pp. 580-592.

[9] F.C. Payton, “Data mining in health care applications,” *Data mining: opportunities and challenges*, IGI Publishing, 2003, pp. 350-365.

[10] A. Ben-Dor and Z. Yakhini, “Clustering gene expression patterns,” *Proceedings of the third annual international conference on Computational molecular biology*, Lyon, France: ACM, 1999, pp. 33-42.

[11] T.H. Hinke, J. Rushing, H. Ranganath, and S.J. Graves, “Techniques and Experience in Mining RemotelySensed Satellite Data,” *Artif. Intell. Rev.*, vol. 14, 2000, pp. 503-531.

[12] Dan Braha, *Data mining for design and manufacturing: methods and applications*, Springer, 2001.

[13] H. Hakkoymaz, G. Chatzimilioudis, D. Gunopulos, and H. Mannila, “Applying Electromagnetic Field Theory Concepts to Clustering with Constraints,” *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, Bled, Slovenia: Springer-Verlag, 2009, pp. 485-

500.

[14] G.F. Tzortzis and A.C. Likas, “The global kernel k-means algorithm for clustering in feature space,” *Trans. Neur. Netw.*, vol. 20, 2009, pp. 1181-1194.

[15] B. Yan and C. Domeniconi, “An Adaptive Kernel Method for Semi-supervised Clustering,” *Machine Learning: ECML 2006*, 2006.

[16] Yadolah Dodge, *The Oxford dictionary of statistical terms*, International Statistical Institute, 2003.

[17] A.L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial Intelligence*, vol. 97, Dec. 1997, pp. 245-271.

[18] R. Weber, H. Schek, and S. Blott, “A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces,” *Proceedings of the 24rd International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., 1998, pp. 194-205.

[19] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, “Locally adaptive metrics for clustering high dimensional data,” *Data Min. Knowl. Discov.*, vol. 14, 2007, pp. 63-97.

[20] E. Xing, A. Ng, M. Jordan, and S. Russell, “Distance Metric Learning, with Application to Clustering with Side-information,” *Advances in Neural Information Processing Systems 15*, 2002, pp. 512, 505.

[21] Huan Liu and Hiroshi Motoda, *Computational Methods of Feature Selection*, Chapman & Hall, CRC, 2008.

[22] H. Lutz, H. Stocker, and J. Harris, *Handbook of Physics*, Springer-Verlag, 2002.

[23] C. Reynolds, C. Burt, and W. Richards, “A Linear Molecular Similarity Index,” *Quant. Struct.*, vol. 11, 1992, pp. 34-35.

[24] E. Hodgkin and W. Richards, “Molecular similarity based on electrostatic potential and electric-field,” *International Journal of Quantum Chemistry*, vol. 14, 1987, pp. 105-110.

[25] R. Carbo, L. Leyda, and M. Arnau, “How similar is a molecule to another? An electron-density measure of similarity between 2 molecular structures,” *International Journal of Quantum Chemistry*, vol. 17, 1980, pp. 1185-1189.

[26] P. Berkhin, *Survey Of Clustering Data Mining Techniques*, 2002.

[27] A. Torda, “Protein Threading,” *The Proteomics Handbook*, 2005, pp. 921-938.

[28] R. Xu and D. Wunsch, “Survey of Clustering Algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, May. 2005, pp. 678, 645.

[29] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, Montreal, Quebec, Canada: ACM, 1996, pp. 114, 103.

[30] R.T. Ng and J. Han, “CLARANS: A Method for Clustering Objects for Spatial Data Mining,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 14, 2002, pp. 1003-1016.

[31] H. Bock, “Clustering Methods: A History of k-Means Algorithms,” *Selected*

Contributions in Data Analysis and Classification, Springer Berlin Heidelberg, 2007, pp. 172, 161.

[32] S. Guha, R. Rastogi, and K. Shim, “CURE: an efficient clustering algorithm for large databases,” *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, Seattle, Washington, United States: ACM, 1998, pp. 73-84.

[33] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, 2002, pp. 881-892.

[34] “The Application of K-Medoids and PAM to the Clustering of Rules,” *Intelligent Data Engineering and Automated Learning – IDEAL 2004*, Springer Berlin / Heidelberg, 2004, pp. 173-178.

[35] R.T. Ng and J. Han, “Efficient and Effective Clustering Methods for Spatial Data Mining,” *Proceedings of the 20th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., 1994, pp. 144-155.

[36] Y. Yang, X. Guan, and J. You, “CLOPE: a fast and effective clustering algorithm for transactional data,” *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada: ACM, 2002, pp. 682-687.

[37] P.S. Bradley and U.M. Fayyad, “Refining Initial Points for K-Means Clustering,” *Proceedings of the Fifteenth International Conference on Machine Learning*,

Morgan Kaufmann Publishers Inc., 1998, pp. 91-99.

[38] Z. Zhang, J. Zhang, and H. Xue, "Improved K-Means Clustering Algorithm," *Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 5 - Volume 05*, IEEE Computer Society, 2008, pp. 169-172.

[39] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

[40] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A New Data Clustering Algorithm and Its Applications," *Data Min. Knowl. Discov.*, vol. 1, 1997, pp. 141-182.

[41] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained K-means Clustering with Background Knowledge," *Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., 2001, pp. 577-584.

[42] M. Bilenko, S. Basu, and R.J. Mooney, "Integrating Constraints and Metric Learning in Semi-Supervised Clustering," *In ICML*, 2004, pp. 81–88.

[43] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment Learning and Relevant Component Analysis," *Proceedings of the 7th European Conference on Computer Vision-Part IV*, Springer-Verlag, 2002, pp. 776-792.

[44] S. Basu, M. Bilenko, and R.J. Mooney, "A probabilistic framework for semi-supervised clustering," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, WA, USA: ACM,

2004, pp. 59-68.

[45] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, “Semi-supervised graph clustering: a kernel approach,” *Proceedings of the 22nd international conference on Machine learning*, Bonn, Germany: ACM, 2005, pp. 457-464.

[46] S. Schaeffer, “Graph clustering,” *Computer Science Review*, vol. 1, 2007, pp. 64, 27.

[47] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 1997, pp. 888–905.

[48] S. Dutt and W. Deng, “Cluster-aware iterative improvement techniques for partitioning large VLSI circuits,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 7, 2002, pp. 91-121.

[49] Benyu Guo and Pen-Yu Kuo, *Spectral methods and their applications*, World Scientific Publishing Co. Pte. Ltd., 1998.

[50] P.K. Chan, M.D.F. Schlag, and J.Y. Zien, “Spectral K-way ratio-cut partitioning and clustering,” *Proceedings of the 30th international Design Automation Conference*, Dallas, Texas, United States: ACM, 1993, pp. 749-754.

[51] A.J. Enright, S. Van Dongen, and C.A. Ouzounis, “An efficient algorithm for large-scale detection of protein families,” *Nucl. Acids Res.*, vol. 30, 2002, pp. 1575-1584.

[52] V. Satuluri and S. Parthasarathy, “Scalable graph clustering using stochastic flows:

applications to community discovery," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France: ACM, 2009, pp. 737-746.

[53] I. Dhillon, Yuqiang Guan, and B. Kulis, "Weighted Graph Cuts without Eigenvectors A Multilevel Approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, 2007, pp. 1944-1957.

[54] L. Sael, B. Li, D. La, Y. Fang, K. Ramani, R. Rustamov, and D. Kihara, "Fast protein tertiary structure retrieval based on global surface shape similarity," *Proteins: Structure, Function, and Bioinformatics*, vol. 72, 2008, pp. 1259-1273.

[55] Ying Zhou, Kaixing Zhang, and Yuankui Ma, "3D protein structure similarity comparison using a shape distribution method," *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*, 2008, pp. 233-236.

[56] R. Daras, D. Zarpalas, D. Tzovaras, and M. Strintzis, "3D shape-based techniques for protein classification," *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2005, pp. II-1130-3.

[57] T. Li, Q. Li, S. Zhu, and M. Ogihara, "A survey on wavelet applications in data mining," *SIGKDD Explor. Newsl.*, vol. 4, 2002, pp. 49-68.

[58] Robi Polikar, *The Wavelet Tutorial*.

[59] Z. Wen, K. Wang, M. Li, F. Nie, and Y. Yang, "Analyzing functional similarity of protein sequences with discrete wavelet transform," *Computational Biology and*

Chemistry, vol. 29, Jun. 2005, pp. 220-228.

[60] J. Qiu, R. Liang, X. Zou, and J. Mo, "Prediction of protein secondary structure based on continuous wavelet transform," *Talanta*, vol. 61, Nov. 2003, pp. 285-293.

[61] C.H. de Trad, Q. Fang, and I. Cosic, "Protein sequence comparison based on the wavelet transform approach," *Protein Eng.*, vol. 15, 2002, pp. 193-203.

[62] H. Hirakawa, S. Muta, and S. Kuhara, "The hydrophobic cores of proteins predicted by wavelet analysis," *Bioinformatics*, vol. 15, 1999, pp. 141-148.

[63] P. Lio and M. Vannucci, "Wavelet change-point prediction of transmembrane proteins," *Bioinformatics*, vol. 16, 2000, pp. 376-382.

[64] A.J. Mandell, K.A. Selz, and M.F. Shlesinger, "Wavelet transformation of protein hydrophobicity sequences suggests their memberships in structural families," *Physica A: Statistical and Theoretical Physics*, vol. 244, Oct. 1997, pp. 254-262.

[65] I. Davidson, K.L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," *In: Proceedings of the Tenth European Conference on Principles and Practice of Knowledge Discovery in Databases*, Springer, 2006, pp. 115–126.

[66] G. Karypis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, vol. 20, 1998, pp. 359-392.

[67] G. Karypis and V. Kumar, *METIS - Unstructured Graph Partitioning and Sparse*

Matrix Ordering System, Version 2.0, 1995.

[68] A. Brander and M. Sinclair, “A comparative study of k-shortest path algorithms,” *Proc. 11th UK Performance Engineering Worksh. for Computer and Telecommunications Systems*, 1995.

[69] H. Tong, C. Faloutsos, and J. Pan, “Fast Random Walk with Restart and Its Applications,” *Data Mining, 2006. ICDM '06. Sixth International Conference on*, IEEE Computer Society, 2006, pp. 613-622.

[70] X. Xu, N. Yuruk, Z. Feng, and T. Schweiger, “SCAN: a structural clustering algorithm for networks,” *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Jose, California, USA: ACM, 2007, pp. 824-833.

[71] G. Karypis and V. Kumar, “Multilevel Graph Partitioning Schemes,” *Proc. 24th Intern. Conf. Par. Proc., III*, CRC Press, 1995, pp. 113–122.

[72] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, 2010.

[73] M. Law, A. Topchy, and A. Jain, “Model-based clustering with probabilistic constraints,” 2005.

[74] N. Blomberg, R.R. Gabdoulline, M. Nilges, and R.C. Wade, “Classification of protein sequences by homology modeling and quantitative analysis of electrostatic similarity,” *Proteins: Structure, Function, and Genetics*, vol. 37, 1999, pp. 379–387.

[75] J. Petke, “Cumulative and discrete similarity analysis of electrostatic potentials and fields,” *Journal of Computational Chemistry*, vol. 14, 1993, pp. 928-933.

[76] C.A. Kieslich, R.D. Gorham, and D. Morikis, “Is the rigid-body assumption reasonable? Insights into the effects of dynamics on the electrostatic analysis of barnase-barstar,” *Journal of Non-Crystalline Solids*, 2010.

[77] R.C. Wade, R.R. Gabdoulline, and F.D. Rienzo, “Protein interaction property similarity analysis,” *International Journal of Quantum Chemistry*, vol. 83, 2001, pp. 122-127.

[78] A.S. Cheung, C.A. Kieslich, J. Yang, and D. Morikis, “Solvation effects in calculated electrostatic association free energies for the C3d-CR2 complex and comparison with experimental data,” *Biopolymers*, vol. 93, 2010, pp. 509-519.

[79] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, and J.A. Mccammon, “Electrostatics of Nanosystems: Application to microtubules and the ribosome,” *Proc. Natl. Acad. Sci. USA*, vol. 98, 2001, pp. 10037–10041.

[80] R.C. Wade, R.R. Gabdoulline, and F.D. Rienzo, “Protein interaction property similarity analysis,” *International Journal of Quantum Chemistry*, vol. 83, 2001, pp. 122-127.

[81] Lokenath Debnath, *Wavelet transforms and their applications*, Birkhauser, 2002.

[82] R. Gorham, C. Kieslich, and D. Morikis, “Electrostatic analysis of the association of the C3d/Efb-C protein complex,” *Under revision*, 2010.

[83] H. Vazquez, A.L.D. Victoria, C. Kieslich, and D. Morikis, “The role of

electrostatics in the function of Factor H, and its relation to complement system-mediated disease," *UCR Undergraduate Research Journal*, vol. 3, 2009, pp. 49–56.

[84] T. Kortemme, D.E. Kim, and D. Baker, "Computational Alanine Scanning of Protein-Protein Interfaces," *Sci. STKE*, vol. 2004, 2004, pp. pl2-.

[85] G. Vriend, "WHAT IF: A molecular modeling and drug design program," *J. Mol. Graph.*, vol. 8, 1990, pp. 52-56.

[86] C Kieslich, G Goodman, H Vazquez, and A Lopez De Victoria, "The effect of electrostatics on Factor H function and related pathologies," 2010.

[87] Richard Coico and Geoffrey Sunshine, *Immunology: a short course*, John Wiley & Sons, 2009.

[88] Anthony L. DeFranco, Richard M. Locksley, and Miranda Robertson, *Immunity: the immune response in infectious and inflammatory disease*, New Science Press, 2007.

[89] Richard A. Goldsby, *Immunology*, W.H. Freeman and Company, .

[90] Alister W. Dodds and Robert B. Sim, *Complement: a practical approach*, Oxford University Press, 1997.

[91] Hans-Uwe Simon and Cezmi A. Akdis, *CRC Desk Reference for Allergy and Asthma*, CRC Press, 2000.

[92] John D. Lambris, Ed., *Current topics in complement*, Springer, 2006.

[93] T. Dolinsky, P. Czodrowski, H. Li, J. Nielsen, J. Jensen, G. Klebe, and N. Baker,

"PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations.," *Nucleic acids research*, vol. 35, Jul. 2007.

[94] D Sitkoff, K.A. Sharp, and B Honig, "Accurate Calculation of Hydration Free Energies Using Macroscopic Solvent Models," *Journal of Physical Chemistry*, 1994.

[95] Jianfeng Yang, "Implementation of a high-throughput computational protocol for calculation and clustering of electrostatic potentials of protein families," *M.S. Thesis*, 2007.

[96] K. Nilsson, M. Andersson, and O. Ingans, "Conformational transitions of a free amino-acid-functionalized polythiophene induced by different buffer systems," *Journal of Physics: Condensed Matter*, vol. 14, 2002, pp. 10011-10020.

[97] Anthony K. Rappe and Dong Xu, *Molecular Mechanics Across Chemistry*, University Science Books, 1997.

[98] I.T. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[99] M. Hearst, "Support Vector Machines," *IEEE Intelligent Systems*, vol. 13, 1998, pp. 28, 18.

[100] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," *Advances in Neural Information Processing Systems 15*, 2003.

[101] P. Debye and E. Hückel, "The theory of electrolytes," *Zeitschrift für Physik*, 1923, pp. 305–324.

[102] K.E.V. Holde, W.C. Johnson, and P.S. Ho, *Principals of Pyhsical Biochemistry*,

Prentice-Hall, Inc., 1998.

[103] D. C. Rapaport, *The art of molecular dynamics simulation*, Cambridge University Press, 2004.