

EMBEDDED CLUSTERING SYSTEMS WITH BIOLOGICAL DATA MINING
APPLICATIONS

by

YONGHUI CHEN

KEVIN D REILLY, COMMITTEE CHAIR
ALAN P SPRAGUE
STEPHEN BARNES
PURUSHOTHAM V BANGALORE
GITENDRA USWATTE-ARATCHI

A DISSERTATION

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

BIRMINGHAM, ALABAMA

May 30, 2006

Copyright by
Yonghui Chen
May 30, 2006

EMBEDDED CLUSTERING SYSTEMS WITH BIOLOGICAL DATA MINING APPLICATIONS

YONGHUI CHEN

ABSTRACT

Mining biological/biomedical data is an emerging area of intersection between biological/biomedical research and computer science. Clustering methods are among core components in many data mining studies, typically, embedded within a complex system to solve real-world problems which require a comprehensive systems approach. This thesis introduces some embedded clustering systems for selected biological problems. Aspects of clustering that are discussed include the clustering algorithm itself and its applications in whole systems views. In chapter 2, MABAC, a new clustering algorithm is introduced along with appropriate testing. In chapter 3, SEQOPTICS, a protein sequences clustering method is presented with data sets extracted from internet databases. Results of SEQOPTICS are compared with two other clustering methods. In chapter 4, an allosteric network of Ligand Gated Ion Channels (LGICs) is discovered by clustering on coupling and correlation analysis. In chapter 5, IMAR, a data mining system for identification of movement classes and analysis in stroke rehabilitation procedures, is introduced. In it clustering, classification and database techniques are integrated into the systems context. The algorithm and some biological data mining systems embedded with clustering are described on chapter-by-chapter base. Results are presented and evaluated in each chapter.

ACKNOWLEDGEMENTS

This thesis is the outcome of years of work during which I have been supported by many people. I would like to take this opportunity to express my gratitude to all of them.

I am deeply indebted to my supervisor Dr. Kevin D. Reilly, whose encouragement and stimulating suggestions helped me throughout the research and writing of this thesis. He has put in much effort to help me on every aspects towards my graduation. Special thanks go to my co-advisor, Dr. Alan P. Sprague, who directed me to the door of data mining research. His expertise on algorithm design and data mining made his suggestions very valuable to me.

My sincere thanks are due to Dr. Gitendra Uswatte, who provided me with a range of biomedical problems to be solved by computer systems, and supported me with his research grant. I would like to thank Dr. Purushotham V. Bangalore who kept an eye on the progress of my work and always was available when I needed his advice. I would also like to thank Dr. Stephen Barnes, who monitored my work and expanded effort in reading my writings and providing me with valuable comments on early versions of this thesis.

I am grateful to Dr. Chengcui Zhang. Although she is not a formal mentor of my committee, she helped me with useful ideas as well as facilities for me to do experiments. I am also grateful to the department of Computer & Information Sciences at the University of Alabama at Birmingham for providing me excellent work environments, specially helpful and friendly classmates and colleagues, during the past years.

With a deep sense of gratitude, I wish to express my thanks to my parents who formed part of my vision and taught me the good things that really matter in life. The support and encouragement from my sisters and brothers rendered me the sense and the value of siblings. I am glad to be one of them.

Especially, I would like to give my special thanks to my wife Yong, for her laughter and inspiration. Without her loving support, understanding and sharing I would never have completed my present work.

Finally, I would like to thank all whose direct and indirect support helped me completing my thesis in time.

TABLE OF CONTENTS

	<i>Page</i>
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF FIGURES	xi
LIST OF TABLES	xii
CHAPTER.....	1
1 INTRODUCTION	1
1.1 Background.....	1
1.2 Data Mining.....	3
1.2.1 Data mining defined	3
1.2.2 Overview	4
1.2.3 Data mining in this dissertation	4
1.2.4 Data mining methodology	5
1.3 Clustering Analysis	6
1.3.1 Clustering defined	6
1.3.2 Overview	6
1.3.3 Clustering in this dissertation	7
1.3.4 Clustering methodology.....	9
1.4 Chapter Overview	13
1.4.1 MABAC, matrix-based clustering algorithm	13
1.4.2 SEQOPTICS, a protein sequence clustering method	14
1.4.3 PANA, protein allosteric network analysis for ligand gated ion channels.....	15
1.4.4 IMAR, identifying movement from accelerometers read- ings in rehabilitation study	16
1.5 How Projects Integrate	17

2	MABAC: MATRIX-BASED CLUSTERING	19
2.1	Introduction	19
2.1.1	Typical uses of clustering	20
2.1.2	Quality of clusters	20
2.1.3	Methods of clustering	21
2.1.4	Illustration of hierarchical clustering.....	21
2.1.5	MABAC	22
2.2	Related Work	23
2.2.1	SLINK	23
2.2.2	OPTICS	23
2.2.3	ROCK.....	24
2.2.4	CHAMELEON	24
2.3	Similarities Among Above Algorithms	24
2.3.1	Square of a graph: OPTICS and ROCK	25
2.3.2	Merging by an edge cut measure: CHAMELEON and ROCK.....	25
2.3.3	Merging by single link: SLINK and OPTICS	26
2.4	MABAC: Matrix Based Clustering algorithm	26
2.4.1	Overview	26
2.4.2	Construction of the similarity matrix	28
2.4.3	Matrix operation	29
2.4.4	Goodness function.....	29
2.4.5	Hierarchical clustering according to the goodness function	30
2.4.6	Remarks on MABAC's goodness function	30
2.5	Experimental Results.....	32
2.5.1	Comparative results - five examples	32
2.5.2	MABAC and fuzzy clustering	35
2.6	MABAC Performance Analysis	38
2.7	Conclusions and Future Work	39
3	PROTEIN SEQUENCE CLUSTERING METHOD: SEQOPTICS	41
3.1	Introduction	41
3.1.1	Public protein databases.....	42
3.1.2	Sequences clustering methods	43
3.2	OPTICS: Background	46
3.2.1	Definitions	46

3.2.2	Extracting clusters	48
3.2.3	Protein distance measures	48
3.3	Methods	49
3.3.1	SEQOPTICS overview	50
3.3.2	Data sets	51
3.3.3	Computing distance	52
3.3.4	OPTICS clustering	53
3.4	Results	54
3.4.1	Visualization of the cluster structure	55
3.4.2	Extraction of the final clusters	57
3.4.3	Validation of the cluster set	57
3.5	Conclusion and Future Work	62
4	PROTEIN ALLOSTERIC NETWORK ANALYSIS FOR LIGAND GATED ION CHANNELS	64
4.1	Introduction	64
4.1.1	Ligand Gated Ion Channels	65
4.1.2	Statistical coupling analysis	66
4.1.3	McLachlan-based substitution correlation analysis	67
4.1.4	Clustering analysis	67
4.2	Methods	68
4.2.1	Data source and multiple sequence alignment	69
4.2.2	Static and coupling energy calculation	70
4.2.3	Correlated mutational analysis	71
4.2.4	Clustering analysis	71
4.2.5	Visual presentation	72
4.3	Results	73
4.3.1	Static energy	73
4.3.2	Coupling analysis results	75
4.3.3	Correlated mutation analysis	77
4.3.4	Clustering analysis	77
4.4	Results Discussion	78
4.4.1	Comparison of SCA and McBASC in clustering results ..	82
4.4.2	Activation pathway	83
4.4.3	Sites of action for allosteric modulators	87
4.5	Conclusion and Future Work	88

5	IMAR: IDENTIFYING MOVEMENT FROM ACCELEROMETER RECORDINGS OF REHABILITATION PATIENTS	90
5.1	Introduction	90
5.2	Overview	92
5.2.1	Getting data into the repository	94
5.2.2	Extracting information from the repository	95
5.2.3	System evolution	95
5.3	Data Collection	96
5.3.1	IMAR input	96
5.3.2	Accelerometer measurements	98
5.3.3	Accelerometers in the IMAR context	100
5.3.4	Shaping tasks	101
5.3.5	Functional movements	102
5.3.6	Data sources	103
5.3.7	Data preprocessing and storage	106
5.3.8	Data storage	109
5.4	Mining the Repository	110
5.4.1	Data extraction and visualization	111
5.4.2	Clustering	113
5.4.3	Classification	116
5.4.4	Classification results	117
5.5	Summary and Conclusions	120
6	CONCLUSIONS	123
6.1	Summary	123
6.2	Clustering - Methods and Techniques	125
6.3	System Views	127
6.4	Contributions: Software Systems and Publications	130
	LIST OF REFERENCES	132

LIST OF FIGURES

<i>Figure</i>	<i>Page</i>
2.1 Hierarchical Clustering Dendrogram Example	22
2.2 MABAC Overview	27
2.3 Two Basic Ideas of MABAC	31
2.4 Clusters Results of Data Sets	33
2.5 Matlab Fuzzy k -means Results of Data Set 4	37
2.6 Matlab Fuzzy k -means Results with Added Dimension	38
3.1 OPTICS: Core Point and Reachability Distance	46
3.2 Example about OPTICS Core Objects and Core Distances	46
3.3 SEQOPTICS Overview	50
3.4 Data Set 1 (Pfam)	55
3.5 Data Set 2 (Pfam)	55
3.6 Data Set 3 (NCBI)	55
3.7 Data Set 4 (Swiss-Prot)	55
3.8 Comparison of Two Cluster Sets T and M	60
4.1 System Overview of Protein Allosteric Network Analysis (PANA)	68
4.2 Static Energy Results	74

4.3	Coupling Energy and its Relationship to Inserted Gaps	76
4.4	Coupling Energy Clustering Results	79
4.5	Correlation Coefficient Clustering Results	80
4.6	Mapping Highly Coupled Sites onto 3D Structure	84
5.1	Conceptual Model	94
5.2	IMAR Input Phase	98
5.3	Experiment Set-up	100
5.4	Accelerometer Data Example	104
5.5	Graphical User Interface for Loading Data:.....	107
5.6	Graphical User Interface for Querying Data	111
5.7	Visualization of IMAR Example.....	112
5.8	Two Layer Neural Network Structure Example in IMAR	118
5.9	Neural Network Prediction vs. Therapist-Report	120

LIST OF TABLES

<i>Table</i>	<i>Page</i>
2.1 Comparison of Clustering Results on Desired Clusters	32
3.1 Data Source	51
3.2 Comparison of Clustering Results	61
4.1 Highly Coupled Sites Clustered by SCA or McBASC	81
5.1 Clustering Results of IMAR	116
5.2 Neural Net Classification Results of IMAR	119

CHAPTER 1

INTRODUCTION

This research is a study on data mining systems, particularly, with biological or biomedical interest. Clustering, often used in data mining system, occupy much of the discourse. The plan of this introduction is to first overview some background information on data mining systems and clustering. In both of these overviews we point to places in each of the studies in the following chapters where data mining system and clustering considerations arise. After this background information, each of the four central chapters (chapters 2-5) are abstracted with the intent to emphasize the features of each study and some of their inter-workings.

1.1 Background

In recent years, high-throughput experimental methods in biological/biomedical research have resulted in an enormous amount of data. Much of the data is stored in repositories which are publicly available, e.g., genome data, protein sequences and structures, and information relevant to clinicians and biomedical researchers. Other data banks are less widely available while yet others are private, but such collections may well become available in the future, given advances in network systems research and development that goes on under the label grid computing.

The richness of the experimental data stimulates increasing attention of researchers in biological/biomedical science and in the computing sciences. How to manage the data and extract information from it is a big challenge because of the complex nature

of the data. This requires advanced computing techniques with biological/biomedical information involved.

Computer processing ranges from low to high ends: collecting data, building databanks for local processing, opening them to basic query operations and to more advanced information retrieval processes, to sophisticated modeling and simulation, with results fed back to a core databank. Concepts from data mining arise and will be discussed at the appropriate points.

Data resources, whether from the internet or generated locally, rarely sit on a “shelf” waiting for use. Typically, they need to be cleaned, integrated, and transformed before further research can be applied. Collecting, gaining access to and making use of the mounting data requires work intersecting biological and computing expertise. Opportunity in the computing sciences spawns basic and applied research, from algorithm levels to comprehensive (integrated) application system levels.

In this document, collections of data available on the world-wide-web and local ones from on-going experiments are utilized. There is an intended diversity in the data addressed, from a computing angle to test features in the methods devised. Focus revolves about manipulating stored data to establish a more goal-directed storage scheme and to facilitate subsequent processing. Such local storage extracted from remote locations and set up for convenient processing is a key part of the data warehouse concept, related in many ways to data mining system, and will be briefly discussed in chapter 5.

This dissertation addresses several issues related to the needs and opportunities afforded by the collected data and processing capabilities with computing techniques

associated with data mining, mainly clustering methods. In this introduction, the topics of data mining and clustering are addressed. Then the specifics of the research with emphasis on clustering in conjunction with other forms of analysis are discussed.

1.2 Data Mining

The topic of data mining is addressed since clustering is often dependent it. In this section what is data mining is briefly defined. Next, how data mining makes its several appearances in this document is reviewed on chapter-by-chapter base. Finally, data mining tools are introduced and will be encountered as the writing evolves.

1.2.1 Data mining defined

Data mining is the task of discovering “interesting information,” from large amounts of data, generally stored at remote locations and available on the internet [46]. Interesting information may be “regularities” in the data or “high-level information,” “knowledge” implied by the data.

The mining analogy suggests that the information is hidden in the source data repositories and must be extracted in some manner. To accomplish such a goal there are some widely used methods and approaches allowing data to be viewed from different angles [46] and evaluated via various criteria. Data mining may be seen in both a “narrow” sense and a “broad” sense as discussion below reveals.

1.2.2 Overview

The large amounts of data, often internet accessible, have generated the new interdisciplinary field of data mining. Applications of data mining in applied areas include ones of most concern to us in the biological sciences. This area is widely recognized as the one where science may well make its greatest breakthroughs in the near future.

In the next few subsections, two views of a data mining system are presented. In the first the mining is exercised over data selected from web resources and processed. The data are then discarded with new data culling needed for subsequent, similar processing. This is the narrower view of data mining systems. In the second case, the data is collected and stored long-term for further processing. The data warehouse concept, a single database or a collection of databases, is useful in this case. For present purposes, a warehouse can be taken simply as a collection of data from different databases, and some issues related to this notion within our context will be added later. This second case represents the broader concept of data mining.

1.2.3 Data mining in this dissertation

A data mining study first involves identifying relevant data, which can be difficult if data is hard to locate. In other cases, where there is plentiful data, as in the protein sequencing study (chapter 3), the issue may primarily be selecting among data sources, given the purposes of the research. In this study, an abstract notion of “adequate demonstration” is tailored to the goal of a study. A need for an “adequate diversity” among protein families also applies in the study.

Chapter 4 has common ground with chapter 3, in that it collects protein sequences from internet sources; some of these sources are the same as in the study of chapter 3. In this case, a different kind of constraint is put on the sources, appropriate to the (evolution-based) hypothesis in the protein study.

The data mining conceptualizations mentioned in the above paragraphs overlap with notions relating to knowledge representation systems, pattern recognition system, and simulation environments. Processes appearing in such systems may be most relevant in chapter 5 where a one-time “narrow” data mining system has advanced to the broader warehousing level containing a plethora of processing options with interdependencies, e.g., using clustering and achieving a certain kind of result indicative of how to proceed with a neural net model. The warehouse, in the long run, will store data results, documentation and suggestions for future research.

These advanced systems require important constraints on design of components, storage and processing, and so on. It is worth noting that several components (chapters 2, 3 and 4) have been implemented so that they might be realized in potentially very sophisticated settings such as that of chapter 5.

1.2.4 Data mining methodology

Whether data mining occurs in the narrow or broad sense, tools relevant to understanding acquired data may be very similar. Clustering is a tool used most frequently in this thesis (actually in every one of our four major examples in chapters 2-5). Some other tools includes many conventional statistical ones and some “numerical artificial intelligence,” such as neural networks [49], support vector machines [16], genetic

algorithms [71] and more. Activities with almost every one of these will be seen in later chapters.

1.3 Clustering Analysis

Clustering analysis is an important data mining tool applicable to the biological/biomedical problems of concern in this dissertation. Parallel to the discourse on data mining, cluster and clustering analysis are first defined, then clustering in the dissertation is briefly pointed out and methodologies of clustering are discussed.

1.3.1 Clustering defined

A cluster can be defined at a high level of abstraction: a cluster is a collection (a group) of data objects relatively similar to one another and relatively dissimilar to objects in (other) competing clusters [46]. A cluster of data objects can be treated collectively as one group. A general question facing researchers in many areas of inquiry is how to organize observed data into meaningful structures. Clustering analysis is used to address this question and is an important technique in data mining.

1.3.2 Overview

As a data mining tool, clustering analysis can be employed as a stand alone tool to meet the full need of the researcher. It can appear in a supporting role, either as cooperative processor or as a precursor to subsequent processing. Clustering analysis has been used in a variety of scientific, engineering and business areas, such as pattern recognition, image processing [110], marketing research [105], and biological data

mining [24], e.g., microarray data clustering [38] and protein sequences clustering discussed in chapter 3.

What clustering can do on its own and in conjunction with other forms of analysis is a principal concern in this dissertation. These “other” analyses include visualization, database methodology, neural networks and protein 3-D structural modeling. This will be further elaborated as: 1) a clustering approach at an algorithm level with an associated testing regime involving arguably complex data organizations and aimed at providing a tool which is called upon in later studies; 2) a systems-level effort in protein sequence clustering of a specific clustering method that utilizes visualization of clusters structure in a central role; 3) a second systems-level effort with complex biological front- and back-ends where clustering does “double duty” on intermediate results (see chapter 3) in a manner that, by cooperative computation, strengthens the ultimate conjectures of the study; 4) a third systems-level effort where clustering plays two major roles, one as a precursor of subsequent more intense probing via neural networks, and the other as a means of seeking mainly to confirm achieved results.

1.3.3 Clustering in this dissertation

Clustering is approached in several ways in this dissertation. First, a new clustering method was designed, implemented, tested (chapter 2), and made ready for usage in later chapter work where additional testing is entailed (chapter 5). This implemented system is called MABAC, Matrix Based Clustering algorithm. It is to be noted that this method involves exploiting features from other methods and merging

them into a programmed unit.

Second, an existing method is modified and put to work in a real world application context (clustering proteins from heterogeneous sources and re-identifying the sources). This method is called SEQOPTICS reflecting its application domain, (protein) sequences and the base system on which it is built, OPTICS [7]. The system addresses a number of clustering issues, among them user-system interaction aided by visualization.

Third, a clustering method is embedded in a novel scheme involving basic biological research (chapter 4). This portion of the work is called PANA, which stands for Protein Allosteric Network Analysis. In this research, clustering is an interface nestled in the middle of a complex multi-staged biological and biophysical analysis.

Fourth, another existing clustering method, Matlab Fuzzy C-means, along with MABAC (in chapter 5), is employed as a precursor stage to a follow-up neural network classification model. In some cases, clustering analysis is used to assess consistencies in the complex data of the study. The clustering that occurs can be viewed as part of a narrowly defined data mining effort, i.e., self-contained application as described above. Alternatively clustering can be viewed as occurring within a data warehouse system importing previously collected data from established files and new data containing movement (accelerometer) measurements from (stroke rehabilitation) patients. Processing of data in the warehouse in principle utilizes virtually all the tools associated with data mining.

1.3.4 Clustering methodology

Many clustering algorithms have been developed in research circles and generally fall into a much smaller number of classes or types. For example, Han and Kimber [46] includes five classes of clustering methods: 1) Partitioning, 2) Hierarchical, 3) Density-based, 4) Grid-based, and 5) Model-based. In Han and Kimber’s classification, hierarchical clustering includes clustering methods built upon graph structures. We suggest the latter as a sixth class because there is importance attached to such approaches in this work. Six categories of clustering methods then are:

- **Partitioning Methods** generally result in a set of k clusters. Each cluster may be represented by a centroid (k -medoids) or a cluster representative (k -means). Matlab Fuzzy C-means [54] is a version of k -means where each data point has different probabilities to different clusters. A partitioning method creates an initial partitioning given k , the number of partitions to construct. It uses an iterative relocation technique that attempts to improve the partitioning (e.g., decreases the sum of distance) by moving objects from one group to another. Partitioning methods share low time complexity and can be used for high dimensional data. However, one disadvantage is that a user needs to initialize the number of clusters; another disadvantage is that partitioning methods can only identify convex-shaped clusters and thus are not always good at handling outliers.
- **Hierarchical methods** can be classified as either agglomerative or divisive. Agglomerative clustering starts on “smaller” clusters, merging them into larger

ones. Divisive clustering works in the opposite direction. Agglomerative clustering is the main focus here since it is probably the most commonly used clustering methodology. The construction of an hierarchical agglomerative classification can be achieved by the general algorithm as described in chapter 2's introduction. In agglomerative hierarchical clustering methods two clusters may join by the shortest distance (Single Link or SLINK), the longest distance (Complete Link or CLINK), or the average distance (Average Link). Hierarchical clustering methods are common for several advantages: no need to specify the number of clusters; capability of identifying arbitrary shaped clusters; capability of handling high dimensional data. However, this class of methods is sensitive to outliers, and has high computing complexity.

- **Density-based methods** are based on a density of data points within a region. A density area can be defined with two parameters, maximum distance (ε) and minimum points (*MinPts*), e.g., a 2-D region that contains at least *MinPts* objects in a circle with radius ε . DBSCAN [86] and OPTICS [7] are among examples of this category. OPTICS, based on DBSCAN, produces an ordering of the data points to its density-based clustering structure. Density-based methods are popular because they are able to identify arbitrary shaped clusters by adjusting parameters and thus may be less sensitive to outliers. Additional information about DBSCAN and OPTICS is discussed in chapter 3. One main disadvantage about density-based methods is they require user input parameters that can affect the quality of clustering very much. Another shortcoming is that density-based methods are not suitable for high-dimensional data.

- **Graph based methods** construct a proximity graph with each point representing a vertex and each edge standing for some distance between a pair of points. Further processing is based on the graph or decomposed graph. Three commonly used graph-based methods includes ROCK (Robust Clustering using links) [43], SNN (Shared Near Neighbor [33]), and CHAMELEON [56]. is used. ROCK uses the Jaccard coefficient and a thresholding criterion to establish “links” between categorical-attribute samples. CHAMELEON starts with partitioning the data into a large number of clusters by partitioning the k -nearest neighbor graph where every vertex is connected with only k closest neighbors. SNN employs a shared near neighbor graph, where “near” is defined by user, e.g., a distance less than an assigned cutoff value. Graph-based methods usually provide good clustering quality, however they add extra cost for building a graph.
- **Grid-based methods** first split the dense regions of the clustering space into a finite number of cells. A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter. Clusters are then formed by connecting the dense cells. In this category of clustering are STING [106], WaveCluster [88], and CLIQUE (Clustering In QUEst) [1]. In STING (A Statistical Information Grid Approach), the spatial area is divided into rectangular cells. Statistical information of each cell is calculated, stored, and used to answer queries. WaveCluster is a multi-resolution clustering approach which applies a wavelet transformation to the feature space. CLIQUE automatically identifies subspaces of a high dimensional data space that allow better clustering

than the original space. CLIQUE can be considered as density-based as well as grid-based. It partitions each dimension into the same number of equal length units. A cluster is a maximal set of connected dense units within a subspace. Grid-based methods are good for large data sets because of their low time complexity; and they are easy to parallelize. However, they are not good for high dimensional data.

- **Model-based methods** attempt to optimize the fit between the data and some mathematical model. Statistical and AI approaches are often used in these methods. A model based clustering method may be viewed as a form of clustering via machine learning. It produces a classification scheme for a set of unlabelled objects and finds characteristic descriptions for each class. Among popular approaches used in Model-based clustering are Neural Networks and competitive learning [37]. In neural network approaches, each cluster is represented as an exemplar, acting as a prototype of the cluster. New objects are distributed to the cluster whose exemplar is the most similar according to some distance measure. In competitive learning, a hierarchical architecture of several units (neurons). Neurons compete in a “winner-takes-all fashion for the object currently being presented. The Self Organizing feature Map (SOM) [61] is a good example of a model-based method. In SOM, clustering is performed by having several units competing for the current object. The unit whose weight vector is closest to the current object wins. The winner and its neighbors learn by having their weights adjusted. Model-based methods provide a statistical model for the data and thus may not be sensitive to outliers. Moreover, they

are not scalable for large dimensional data and can not identify non-ellipsoidal and non-convex clusters.

An insight into our overall work effort can be made at this point: over the course of this work, members from almost all these classes have been utilized in one way or another (developed for use from scratch, adapted, adopted, or employed for test purposes).

1.4 Chapter Overview

In the previous sections the discussion was oriented to two main topics, data mining and clustering. From here to the end of this introductory section, the coverage switches to the important elements that make clear how each chapter can be the subject matter of one or more publications (conference papers, poster session presentations, and journal articles). These following systems overview, with the names attached to them: MABAC, SEQOPTICS, PANA, and IMAR, are discussed in detail in chapters 2-5. These brief descriptions illustrate how data mining and clustering evolve within the coverage.

1.4.1 MABAC, matrix-based clustering algorithm

Several purposes of MABAC (Matrix Based Clustering) exist so that its matrix base serves to characterize only a part of what MABAC is all about [26]. MABAC's goals are pitched dually: first, at the algorithm level where it is designed to apply significant features from selected existing methods (see chapter 2 for detail); second, at the systems levels where MABAC is meant to operate as an agent in communication

with other cooperative and competing agents (see chapter 5 for additional comment).

MABAC work began with a survey of prominent clustering methods as detailed in a conference paper which provides part of chapter 2's text [25]. Several clustering methods from which we drew inspiration are outlined in the previous section of this chapter. MABAC itself can be viewed as a hierarchical clustering method with a goodness function based on notions of bond and inner bond that in turn involve direct and indirect link measures [26, 25].

MABAC employs an operation close to matrix multiplication, but with a little modification (as shown in chapter 2). The matrix base is thought helpful for calculations on advanced parallel machinery, though this goal has only been modestly pursued. Besides this point, there is possible exploitation of sparse matrix techniques, since the base matrix may sometimes be sparse via selection of a cut-off parameter.

The goal in this study is to achieve good clustering performance relative to other clustering methods. Experimental data of different shapes and densities is tested in chapter 2. Real data is tested on MABAC in chapter 5 particularly, to provide useful classification information.

1.4.2 SEQOPTICS, a protein sequence clustering method

Protein sequence clustering has been widely used as a part of protein structure and function analysis [81, 64, 57, 108, 63]. The protein sequences clustering effort, SEQOPTICS (chapter 3) [24] is no different in its goals, though the specifics of our study have some novelties to be related shortly.

SEQOPTICS' major attraction lies in the visualization aspects, which it inherits

from its base, OPTICS [7]. SEQOPTICS was put to work on protein sequencing of data collected from open sources on internet. After selecting data, preliminary operations such as cleaning and reformatting were performed on them, and then the cluster action began. Discussions of chapter 3 show SEQOPTICS' interactive work in the visual context. After clustering, output data is set up to allow for further processing.

The goal of SEQOPTICS is to provide good clustering for protein sequence data. In chapter 3, output is sent to an evaluation framework for comparisons with competitive algorithms. Using the latter in default modes, SEQOPTIC emerges in good form according to several commonly used evaluation criteria. SEQOPTICS, its design and operation along with these results, have been accepted for publication [24].

1.4.3 PANA, protein allosteric network analysis for ligand gated ion channels

In chapter 4, clustering activity is demonstrated on Ligand-gated Ion Channels (LGICs), a protein family containing subunits within and across cell membranes. PANA, deriving from "Protein Allosteric Network Analysis," exploits structural models, statistics coupling analysis, and correlation analysis, combined with the clustering activities.

In this system, multiple protein sequences are extracted from public database and then aligned. At the start of the effort, the multiple sequence alignment profile is analyzed by two methods, statistical coupling analysis (SCA) [48] and correlation analysis (McBASC) [39], to produce two independent 2-D matrices of scores. These matrices are then clustered and results are configured for further processing, espe-

cially, validation by 3-D modeling.

The goal is to obtain some important clues about the functional structure of a protein family, LGICs. The results demonstrate an allosteric network of residues that regulates LGICs cooperatively. Chapter 4 describes the details of this system and a research paper has recently been accepted by Journal of Biological Chemistry [23].

1.4.4 IMAR, identifying movement from accelerometers readings in rehabilitation study

IMAR stands for “Identifying Movement from Accelerometer Readings” to automating characteristics of stroke rehabilitation patients’s movement. This study concerns research in which clustering and classification methods are again embedded in a larger data collection and processing context. The system includes key data collection tasks with cleaning and organizing, a rigorously defined central storage system built over past and current data files, and subsequent processing. The latter includes clustering with MABAC and MATLAB’s Fuzzy C-means and classification with Neural Network Models.

This identification is geared to perform differently from previous measurement that included video recordings along with accelerometer readings. A very important goal is to eliminate the video recordings due to their being a human-intensive and costly evaluation task. Another main goal is to develop IMAR to a point where the data mining purview can be assessed relative to the streams of research under such labels as data warehouse, knowledge representation systems, decision support systems and simulation environments.

1.5 *How Projects Integrate*

Some of the earliest comments in this introduction indicate integrations among the projects. In particular, they point to algorithmic work (chapter 2) which carries over into diverse systems projects (chapters 3-5). In these projects, the clustering analysis occurs in distinctly different roles: self-contained (chapter 3) and different embedded modes (chapters 4-5). Chapter 4's embedding mode includes protein structural models as pre- and post-operations relative to the clustering, while the embedding in chapter 5 occurs both as precursor and co-processor agents.

A highly automated system, an important goal in applied computing sciences, is also a main long term goal of each system described in chapter 2-5. All four of these systems provide solid proving grounds for a "generalized application" in the computer science sense. They share a common pattern of pre-processing, processing and post-processing stages with clustering techniques being an important tool. Ultimately it is hoped that automated systems will run with only a few instructions from users, and even provide "surprises" as well as planned results, with research suggestions stored in a data repository accessible to a grid style of operation.

Each study can be viewed as being combined with or embedded within systems and related to ongoing research under labels as knowledge-based systems, decision support systems, data warehouses, and, even, simulation environments. The systems study of clustering protein sequences (chapter 3) clearly needs its results incorporated into a warehouse for further analysis; This research needs a large data set which might ultimately find itself a component in a grid based computational scheme. The study of

ion channels (chapter 4) also has a complex set of initial results that draw attention for further research and provide guidelines on specific work to be done (and recorded in the warehouse with the existing results). The study of stroke rehabilitation patients (chapter 5) currently is pretty much a “local” study but may soon lead to a fully endowed data warehousing system, again with implications for the data grid. The algorithm work (chapter 2) undergoes a testing regime where each recorded result feeds the others. Results need to be incorporated into a data warehouse so they can be employed in further research on the algorithm itself and on its relationships to other complementary and competing algorithms.

CHAPTER 2

MABAC: MATRIX-BASED CLUSTERING

Clustering has been widely used in a variety of areas and many clustering algorithms have been developed in response [46]. In this chapter a matrix based clustering algorithm (MABAC) is presented. MABAC measures the “bond” of two clusters based on a goodness function which is computed via matrix manipulation that utilizes not only direct links but also indirect links between two clusters. The effectiveness of MABAC is demonstrated with several data sets that contain points in 2D space, a couple of which can not be captured by widely discussed methods such as OPTICS [7], CHAMELEON [56], or Matlab Fuzzy Clustering [54]. MABAC may be plugged into applications such as protein sequence analysis as in Chapter 3, microarray data analysis, multiple sequence alignment, and in the studies on stroke rehabilitation movement reported in Chapter 5 in this document.

2.1 Introduction

A cluster is a collection of data objects relatively similar to one another in some respect and relatively dissimilar to the objects in other clusters [46]. Clustering analysis is an important technique in data mining. It is a process of grouping a set of physical or abstract objects into classes of similar objects. Clustering can be viewed as unsupervised classification. Usually there are no predefined classes.

2.1.1 Typical uses of clustering

Clustering often is employed as a stand-alone tool to get insight into data distributions. It can be also used as a preprocessing step for other algorithms. Applications of clustering that are widely cited include but are not limited the following: market or customer segmentation [105], web document classification [109], spatial data analysis [110] and biological studies such as micro array data analysis [38], sequence data analysis [24]. Clustering used in preprocessing for other algorithms is illustrated in the popular MATLAB Fuzzy Toolkit manual, as a preprocessing to a Neural Network classification [54].

2.1.2 Quality of clusters

Good clusters show high similarity within a group and low similarity between clusters. The quality of a clustering result typically is assessed not only by some mathematical measures but the ability to discover hidden patterns. Evaluation of clustering algorithms typically uses criteria such as efficiency and effectiveness [46].

With increasing data size efficiency is very important. To overcome problems of efficiency people can use sampling in combination with clustering or use a faster computer with more memory. However, the quality of the clustering result is even more important and it is generally more difficult to improve.

2.1.3 *Methods of clustering*

Classical approaches [46] to clustering include partitioning methods such as k-means, hierarchical clustering, density-based approaches and graph-based algorithms. There is also a variety of soft clustering techniques, such as those based on fuzzy logic or statistical mechanics. In these cases, a data point may belong to multiple clusters with different degrees of membership. Hierarchical agglomerative clustering methods are very popular with prominent versions such as single-link (SLINK) [89], CHAMELEON [56], BIRCH [111], CURE [42]. Chapter 1 provides more details about six main categories of clustering.

2.1.4 *Illustration of hierarchical clustering*

Hierarchical clustering may be represented by a two dimensional diagram known as dendrogram (see Figure 2.1) which illustrates the fusions or divisions made at each successive stage of analysis [47]. Here we consider single linkage (SLINK), one of the simplest agglomerative hierarchical clustering method using a dendrogram to show the core idea of hierarchical agglomerative clustering. SLINK is also known as the nearest neighbor technique. The defining feature of the method is that the distance between groups is defined as the minimum distance between the closest pair of clusters.

In Figure 2.1, at the start each object is a cluster. B merges with C in the first stage since the distance between B and C is the minimum among all pairs of clusters. The distance between a cluster, say BC , and A may be the closest distance (Single-

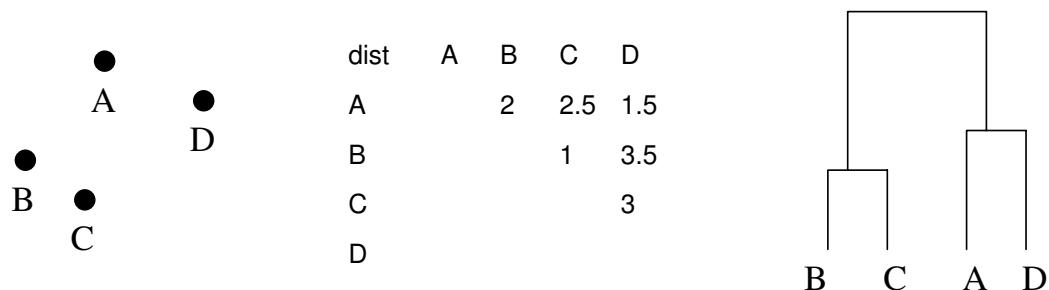


Figure 2.1: Hierarchical Clustering Dendrogram Example

Note the adjacency matrix representation of the initial situation in the center of the diagram, and the dendrogram at the right

Link Distance). Here the Single-Link Distance between BC , and A is the distance between A and B . In other schemas, average distance or maximum distance may be employed. The closest distance among the now three clusters BC , A , and D is the distance between A and D ; thus A and D are merged. Lastly two clusters BC and AD are merged. The clustering can stop at any number of 4, 3, 2, or 1 clusters.

All hierarchical agglomerative clustering methods share the same topology as shown in this example. Differences arise because of the way distance (or similarity) between clusters is defined.

2.1.5 MABAC

In this chapter we present a novel hierarchical agglomerative clustering algorithm called MABAC (Matrix Based Clustering Method) [26] that measures the similarity of two clusters using a new “bond” concept. “Bond” is computed according to operations on the similarity matrix (often referred to as the adjacency matrix) and counts not only direct link between two clusters but also an indirect link between two clusters. The effectiveness of MABAC is demonstrated with several data sets containing points

in 2-D space and clusters of different shape, density, size, noise and artifacts.

2.2 *Related Work*

A principal view adopted in MABAC is that improved clustering quality can be achieved through exploiting commonalities among existing clustering methods, e.g., considerations relating to merging clusters and criteria for it. Several commonalities discussed in this chapter includes single link merging (SLINK, OPTICS), edge cut merging (CHAMELEON, ROCK), and criteria based on the square of the adjacency matrix (OPTICS, ROCK). In response, some comparative information is analyzed to uncover related clustering methods. This serves as background for MABAC, a new hierarchical clustering algorithm.

2.2.1 *SLINK*

The SLINK (single link method), as mentioned earlier, is probably the best known of hierarchical methods. It operates by joining, at each step, two most similar objects that are not yet in the same cluster. The name single link refers to the joining of pairs of clusters by the single shortest link between them [89]. SLINK is easy to implement but the quality of clustering may not be fully satisfactory.

2.2.2 *OPTICS*

OPTICS (Ordering Points To Identify the Clustering Structure) creates an ordering of a data set representing a “density” based clustering structure [7]. The cluster ordering is said to contain information comparable to other density-based clustering

algorithms. Parameter adjustments are important in density-based methods. A big advantage of OPTICS is its provision of a visualization for datasets so that user can interactively choose clustering parameters that affect the results.

2.2.3 ROCK

ROCK employs a link notion but employs a distance measure during merging of points [43]. It also introduces a “global” property stated to promote good quality. Although ROCK was developed for categorical data, some of the ideas of ROCK can be used in other types of data by other algorithms such as CHAMELEON.

2.2.4 CHAMELEON

CHAMELEON is a graph-based hierarchical clustering algorithm designed to fit a “dynamic” model for merging [56]. Two clusters are merged only if relative interconnectivity and relative closeness are both high.

2.3 Similarities Among Above Algorithms

The MABAC algorithm was created after the exploration of the commonalities among the above agglomerative hierarchical clustering algorithms. In the following several subsections we describe three matrix based methods, each of which has been used by two of the above algorithms. These algorithms are agglomerative hierarchical algorithms: in each cycle, the two most similar (by some criterion) clusters are merged. MABAC applies a new similarity measure that is rooted in these methods.

2.3.1 Square of a graph: OPTICS and ROCK

In OPTICS, a data set is represented by an ε -threshold graph, where ε is *reachability distance* chosen by user (defined in the original paper [7]). Let the adjacency matrix of the graph be called A , and let $B = A + I$ where I is an identity matrix. It may be seen that every diagonal entry of the product of this matrix with itself, $B_{[v][v]}^2$, is the number of points adjacent to v , i.e., having distance from v less or equal than the user chosen ε . In OPTICS core points are defined as points v such that $B_{[v][v]}^2$ is larger than or equal to a threshold (called *MinPts*). The merging process acts principally on core points.

In ROCK, the notion of bond is the central tool. Bond is defined as the number of common neighbors. Where A is an adjacency matrix, let $B = A + I$ where I is an identity matrix, it may be seen that $B_{[v][w]}^2$ is the number of common neighbors of v and w .

2.3.2 Merging by an edge cut measure: CHAMELEON and ROCK

CHAMELEON and ROCK are both based on weighted graphs. For CHAMELEON, weights are a similarity measure on vertices (commonly, the input file is a square matrix of similarities between vertices). For ROCK, we regard B^2 as the adjacency matrix of a weighted graph. CHAMELEON and ROCK perform clustering according to an edge cut measure. Where X and Y are disjoint sets of vertices, we define (X, Y) to be the set of edges having one end in X and one end in Y , and define its capacity $cap(X, Y)$ to be the sum of weights of the edges in (X, Y) .

In ROCK, for any pair of vertices v and w , $cap(v, w)$ is written as $link(v, w)$ and $cap(X, Y)$ is $link(X, Y)$. In the merging process, ROCK uses cut capacity to decide which two clusters to merge: ROCK merges the two clusters C and C' such that $cap(C, C')/f$ is maximum, where f is a function of $|C|$, $|C'|$, and a third quantity θ (θ is a constant provided by the user).

CHAMELEON merges the two clusters C and C' such that $cap(C, C')/g$ is maximum, where g is a function of the capacities of edge cuts of C and C' . Whereas ROCK uses an absolute criterion (although f takes the number of vertices in C and C' into account, f does not take density of points in C and C' into account), CHAMELEON uses a relative criterion (g does take density into account).

2.3.3 Merging by single link: SLINK and OPTICS

Whereas SLINK applies the single link measure to all edges of the graph, OPTICS restricts the single link merging process to the edges that join core points. This distinction disappears if $MinPts$ is set to a sufficiently low value. For example, if the $MinPts$ is set to 1, OPTICS becomes SLINK.

2.4 MABAC: Matrix Based Clustering algorithm

2.4.1 Overview

In this section we present MABAC, a new clustering algorithm aimed at achieving results that may be an extension or an improvement over some of the existing agglomerative hierarchical clustering algorithms discussed in Section 2.1. Several examples

(see Section 2.5) will explain it in detail.

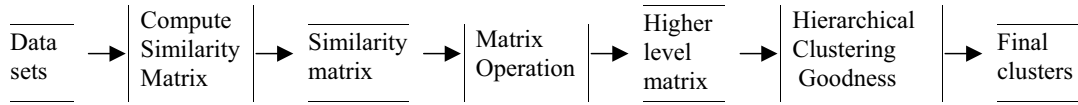


Figure 2.2: MABAC Overview

Figure 2.2 provides the key stages (phases) of this algorithm. MABAC operates on a matrix M in which $M[i, j]$ represents the similarity between two data points i and j . There are three phases in the algorithm seen in Figure 2.2’s “process” blocks (vertically delimited).

First a similarity matrix is computed from input data. Different similarity functions can be employed. In this chapter, for two-dimensional data we restrict the choices to use the reciprocal of Euclidean distance as a similarity measure.

Second, matrix operations are applied to the similarity matrix. The most common operation is “*multiple*,” which is somewhat different from standard matrix multiplication (see Section 2.4.3). Another operation is “*pow*”, i.e., the exponential of the similarity matrix.

Third, a general hierarchical clustering algorithm is applied with a specified goodness function. The goodness function is essential for the clustering quality. We use a unique goodness function that measures both between-group and within-group properties.

A key feature of MABAC is that it determines the pair of most similar sub-clusters by taking account both direct link and indirect link. Alternately, it merges two clusters according to both within-cluster and between-cluster properties.

2.4.2 Construction of the similarity matrix

Many methods can be applied to build a similarity matrix. Since experimental data sets in this section are two-dimensional we use a function to convert the Euclidian distance to similarity values. The similarity value should lie between 0 and 1 where 0 means no similarity at all and 1 means identity.

Two parameters are used here: a normalized coefficient α and a cutoff value γ . Both α and γ are values between 0 and 1. Link of a data point to itself is set at 1, i.e., $link[i][i] = 1$. Link of a point to another point is defined as

$$link[i][j] = \alpha * (min/distance[i][j]) \quad (2.1)$$

where min is the minimum distance determined over the entire distance matrix. If distance between two points i and j is larger than $\gamma * max$, then $link[i][j] = 0$, where max is the maximum distance.

The higher the cutoff value γ , the more global the information becomes. In a similar fashion, lower cutoff values lead to more local clustering information. A commonly used cutoff value is 0.3 but depends on properties of data. The cutoff value here is very related to the cutoff value in DBSCAN [86].

The normalized coefficient α is often set in the range $0.5 \sim 0.8$. It affects results very much. For high dimensional data or protein sequence data, different methods may be applied to do the transformation. For example, the linear correlation coefficient may be used for high dimensional data such as micro array data set. A normalized Smith-Waterman score [90] is used for protein sequence clustering as in Chapter 2.

2.4.3 Matrix operation

A very common operation is the multiplication of two matrices. But MABAC's matrix multiplication is a little different from general multiplication. The following pseudocode illustrates what “*multiple*” does. Notice that the algorithm does not sum up the link between an object and itself. This removes the duplication of the direct link from a data point to itself. For example, in Figure 2.3, the link from *a* to *b* through *a* or *b* should not be counted.

Pseudocode 1 (Algorithm for the initial bond matrix).

```

for i = 0 to m1Row
  for j = 0 to m2Column
    for k = 0 to m1Column
      if(k!=j) mult[i][j] += m1[i][k] * m2[k][j];

```

2.4.4 Goodness function

As discussed earlier, a main difference among hierarchical agglomerative clustering methods is the distance (or similarity) measure. The original intention was to use a single goodness function that can mimic OPTICS, ROCK and CHAMELEON. But it came out as a unified function that counts both within cluster information and between cluster information. Two definitions are used:

1. $bond(c1, c2) = \frac{link(c1, c2)}{|c1|*|c2|}$, where $link(c1, c2) = \sum link(p1, p2)$, $p1 \in c1$, $p2 \in c2$,
i.e., the average link between two clusters;

2. $innerBond(c) = bond(c, c)$.

The goodness function is defined as the following:

$$goodness(c1, c2) = \frac{bond(c1, c2)}{innerBond(c1) \times \frac{|c1|}{|c1|+|c2|} + innerBond(c2) \times \frac{|c2|}{|c1|+|c2|}} \quad (2.2)$$

This goodness function counts not only the link between two clusters but also the data shape and density within each individual cluster by introducing weighted $bond$ and $innerBond$.

2.4.5 Hierarchical clustering according to the goodness function

The hierarchical algorithm and the general hierarchical agglomerative clustering algorithm are the same for this operation. As discussed earlier, at each stage two clusters merge according to the goodness function until the termination criterion is reached. In MABAC experiments, the stop criterion is the number of clusters defined by users. In the future, we hope to define an optimized criterion according to the goodness function. The pseudocode is as the following.

Pseudocode 2 (Algorithm for merging clusters).

while (number of clusters > required number)

Find the closest pair according to the goodness(u,v);

Merge(u,v);

2.4.6 Remarks on MABAC's goodness function

The goodness function defines two basic ideas of MABAC:

1. The bond between two clusters depends not only on the direct link but also on indirect links.
2. The merging criterion depends not only on the bond between two clusters but also on the inner bond of each cluster.

This is explained by two examples as shown in Figure 2.3. Figure 2.3A depicts the first idea of MABAC, where the bond between two points a and b is the sum of direct link (0.1) and indirect links (product of 0.2×0.2 and 0.3×0.15 , corresponding to the dotted-line), i.e., $bond(a, b) = 0.1 + 0.2 \times 0.2 + 0.3 \times 0.15 = 0.185$. This idea is further supported by the experiment results in the Section 2.5.

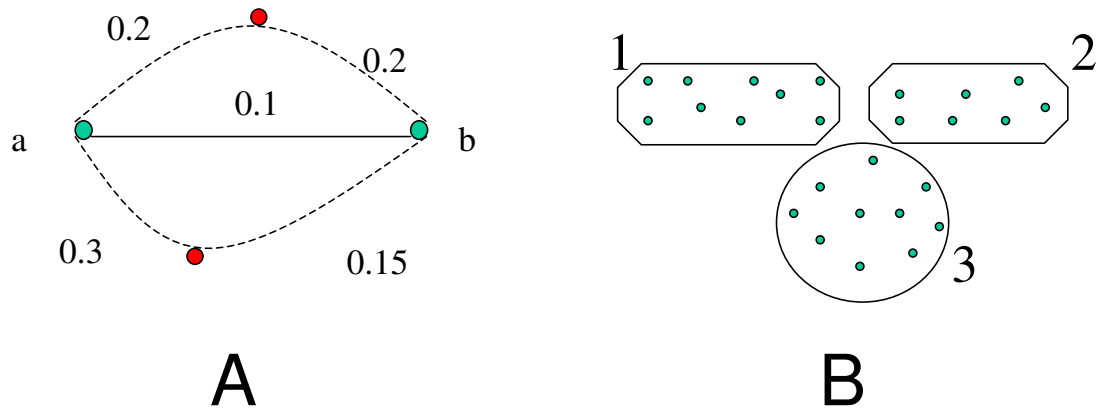


Figure 2.3: Two Basic Ideas of MABAC

A. The bond between two grey points depends on not only the direct link but also the indirect link through two black points. B. Cluster 1 merges with cluster 2 instead of cluster 3 because of *innerBond*.

In Figure 2.3B, Cluster 1 merges with cluster 2 instead of cluster 3. Although the $bond(1, 2)$ (the bond between cluster 1 to cluster 2) is equal to the $bond(1, 3)$ (the bond between cluster 1 to cluster 3), cluster 1 merges with cluster 2 because $innerBond(2)$ (the inner bond of cluster 2) is smaller than $innerBond(3)$.

2.5 Experimental Results

In this section, experiments with MABAC are presented and compared with performance of publicly available version of OPTICS, CHAMELEON, and Matlab Fuzzy C-means. Two dimensional data are tested in these experiments in part because it is easy to visualize the resulting cluster structures. Also, we do not have "true" clusters and therefore we cannot apply numerical evaluation techniques such as the *Jaccard Coefficient* [52]. Two dimensional data provides a natural first level of testing; higher dimensional cases (for MABAC and other clustering techniques) will be demonstrated in Chapter 5, where "real world" data is used allowing more stringent evaluation methods are employed.

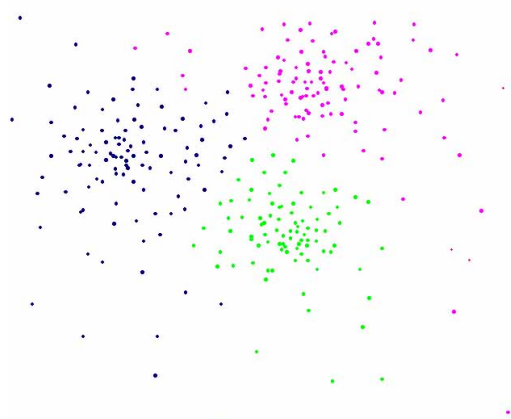
2.5.1 Comparative results - five examples

Five datasets with different "shape" and density were used. The data distributions and MABAC clustering results (different colors represent different clusters) are shown in Figure 2.4. The comparison of results from different clustering methods on these 5 data sets are summarized in Table 2.1.

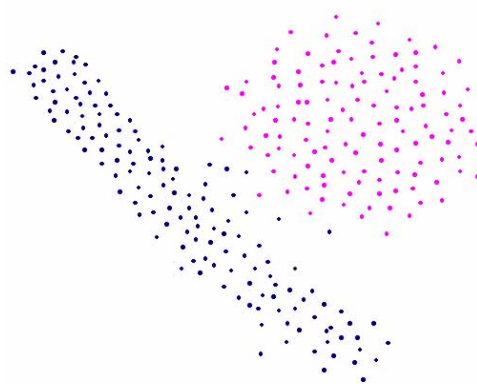
Table 2.1: Comparison of Clustering Results on Desired Clusters

Data Set	MABAC	OPTICS	CHAMELEON	Fuzzy k-means
1	Yes	Yes	No	Yes
2	Yes	Yes	Yes	No
3	Yes	No	No	Yes
4	Yes	Yes	No	No
5	Yes	Yes	Yes	Yes

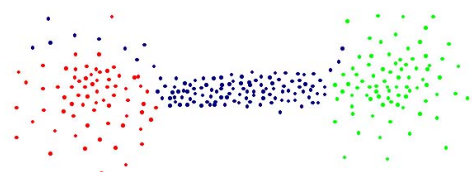
The first data set (dt1) shown in Figure 2.4A is taken from a paper on OPTICS [7].



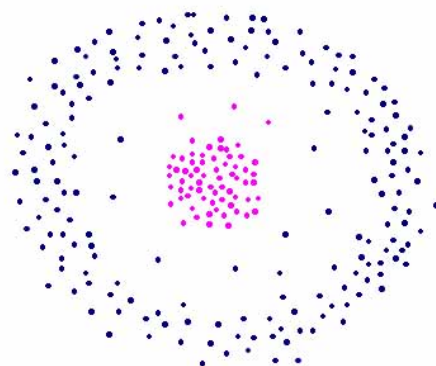
A. Data set 1



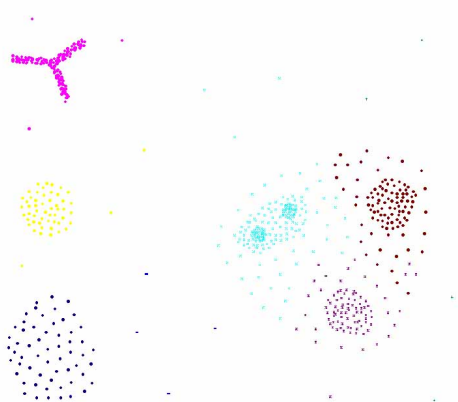
B. Data set 2



C. Data set 3



D. Data set 4



E. Data Set 5

Figure 2.4: Clusters Results of Data Sets

It contains three clusters with different densities. OPTICS achieves a three-cluster structure but, unfortunately, the results fail to consume all the data points leaving them as noise. CHAMELEON fails to give a three-cluster result. Matlab Fuzzy C-means is a partitioning clustering method and it works well for regularly shaped clusters like this example. MABAC results are shown in Figure 2.4A with three clusters in different colors.

The second data set (dt2) contains two clusters with totally different “shape,” like “hamburger and hot dog.” OPTICS works only if parameters are carefully chosen. CHAMELEON results are good in this case. MABAC also gives a very good clustering result as shown in the Figure 2.4B. MatLab Fuzzy C-means does not give the similar cluster structure as shown in the Figure 2.4B. Some more analysis of Matlab Fuzzy C-means results will be further analyzed later in this chapter.

The third data set (dt3) contains three irregularly shaped clusters. Neither OPTICS nor CHAMELEON obtains a three-cluster result, whereas MABAC does, as seen in Figure 2.4C. Matlab Fuzzy C-means gives similar results to MABAC.

The fourth data set (dt4) contains two groups of data points forming a central core and an outer ring. CHAMELEON fails to detect the two-cluster result shown in Figure 2.4D. Both OPTICS and MABAC recognize this two-cluster structure. Matlab Fuzzy C-means clustering does not provide similar results to MABAC for this data set and will be further analyzed later in this chapter.

The fifth data set (dt5), also from the paper on OPTICS [7], is seen in Figure 2.4E. MABAC and OPTICS both give very good results. CHAMELEON has a problem finding the small clusters in this data set. Matlab Fuzzy C-means works well for this

data set.

2.5.2 MABAC and fuzzy clustering

In studies with MABAC, accent has been at the algorithm level. But in some later work, MABAC is used at a higher systems level (see chapter 5) and might work collaboratively (or competitively) with other clustering methods. These methods may be viewed as a set of “agents” in a way similar to what was shown in [82], where a neural network agent (NNA) develops “interesting” results that could be shared with other agents.

A similar situation obtains here in the sense that MABAC “agent” produces “interesting” (non-intuitive and even “artistic”) results that it can share with other agents. To illustrate, fuzzy clustering is taken as MABAC’s “other” agent, similar in ways to chapter 5. Two studies are summarized here.

The first study was based on the dt4, which has some points in the center and some outside points forming a ring. MABAC reported two clusters, the core and the ring. When the (Matlab Fuzzy C-Means) number of clusters was set at two, these data points were approximately divided into two parts so that points from both the core and ring were in each of the clusters.

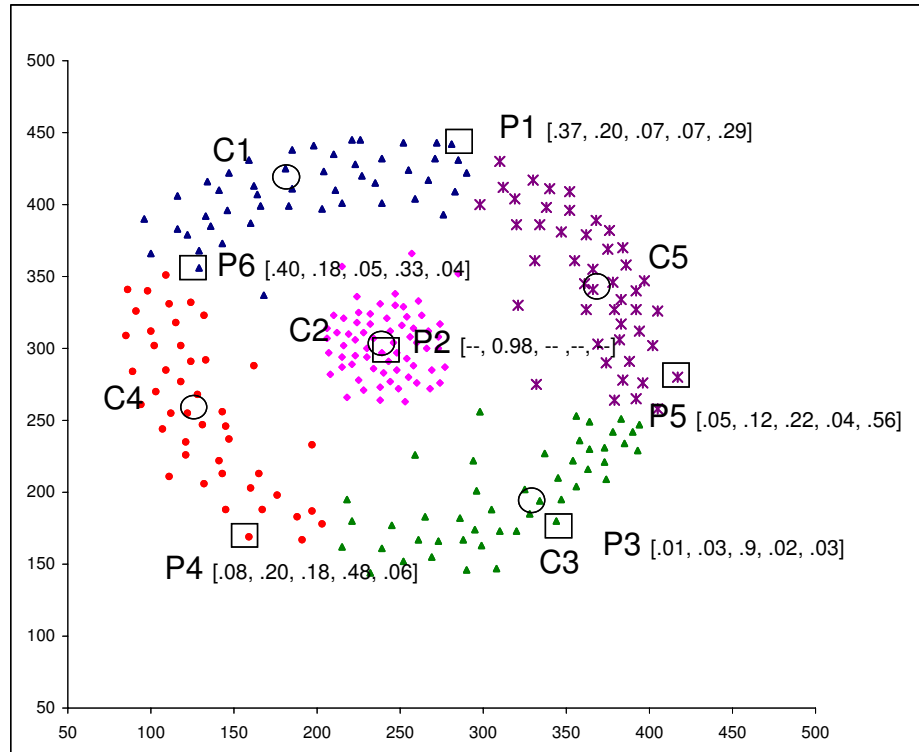
A strategy was adopted to separate out the core and then to analyze ring membership values. Increasing the number of clusters to five resulted in a core cluster and four other (approximately equally sized) ring clusters (see Figure 2.5, where C1-C5 represent cluster centers.) Membership values of several points (P1-P6) in clusters C1-C5 were displayed.

Points near borders of ring clusters (e.g., P1, P5, and P6) have large membership values in adjacent ring clusters, establishing an affinity between rings, putatively establishing a “higher level clustering.” Points farther from the borders showed strongest affinity to their native cluster and often, secondly, to the core. The strong core along with more ambiguous peripheral regions seems a good initial result towards a supra-cluster constellation. A point in the heart of core, e.g., P2, shows almost zero affinity relative to any other cluster. This may be expected but it helps establish integrity to the separation of core from ring.

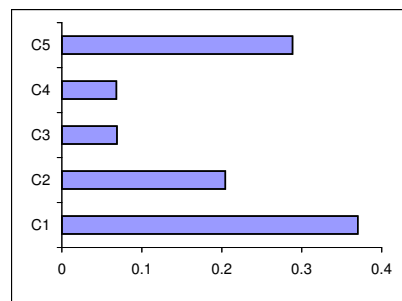
The graphic (Figure 2.5B) showing memberships for P1 is instructive to how the cluster structure of data can be visualized. A graph like this can be produced for every point in the data set.

In the second study, two of the above data sets, dt2 and dt4, the ones on which fuzzy clustering did not achieve the desired results, were augmented with an extra dimension. Data objects that were originally close to each other in the plane became separated in the new 3-D coordinates. The inner core of dt4 was moved up a distance comparable to the radius of outer circle whereas the “hamburger” of dt2 was moved up a distance comparable to the radius of “hamburger”. Three-dimensional projections of desired clusters appeared as shown in Figure 2.6A and Figure 2.6B. In the agent purview, this analysis is a result of searching for fuzzy agent results comparable in performance to MABAC’s agent.

These results are meant to be an initiating step in viewing MABAC in a sub-system, agent-style form of computation. Further research is necessary to improve the fuzzy agent. The notion of applying clustering to the membership values of the



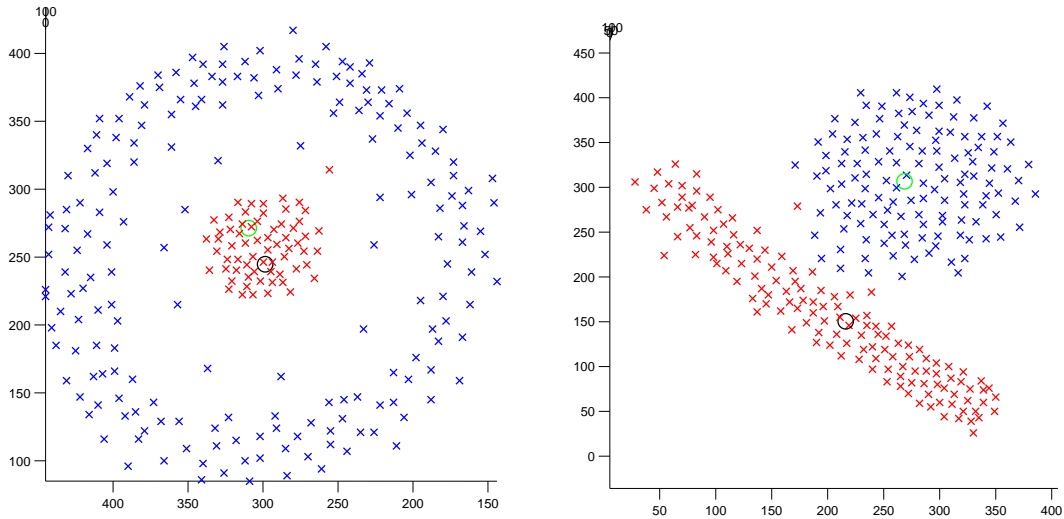
A. 5 different clusters are labelled in different colors. 6 data points are used in analysis of these results.



B. Membership values of P1 to different clusters.

Figure 2.5: Matlab Fuzzy k -means Results of Data Set 4

initial round of clustering is a potentially productive idea. Two dimensional data points used in this chapter may be viewed in the plane. However, high dimensional data is difficult to visualize, and thus it is necessary to explore Matlab's or other fuzzy clustering schemes so that strategies for dealing with higher dimensional data can be developed.



A. Two different clusters are discovered by Matlab Fuzzy C-means on dt2 if 3-d information is added.

B. Two different clusters are discovered by Matlab Fuzzy C-means on dt4 if 3-d information is added.

Figure 2.6: Matlab Fuzzy k -means Results with Added Dimension

2.6 MABAC Performance Analysis

In this section attention is restricted to MABAC, the algorithm created by us. The overall computational complexity of MABAC depends on the time to carry out the matrix operation. If it is a full matrix, then it requires $O(n^3)$ to do the matrix multiplication. If it is a sparse matrix data structure, the average number of non-zero neighbors of a vertex is m , then the time complexity reduces to $O(mn^2)$.

The amount of time required by the first phase is $O(n^2)$. Basically in this step one just needs to convert the distance to similarity. For high dimensional data the computation per distance is $O(d)$ where d is the number of dimensions. So the total complexity for this step would be $O(dn^2)$. The amount of time required by the third phase is $O(n^3)$. If a priority queue is used, each merge is $O(n \log n)$. Thus this phase uses $O(n^2 \log n)$ time.

Overall the time complexity is $O(mn^2)$. This is an expensive algorithm but the clustering quality is deemed good. One advantage of this algorithm is this can be used for outliers detection by checking the density of each of the data points. Low-density data points tend to be outliers. Another advantage is that it is possible to create a termination criterion according to the goodness function.

2.7 Conclusions and Future Work

In this chapter a novel hierarchical algorithm, MABAC, was discussed. It involves not only direct link but also an indirect link, not only between-cluster information but also within-cluster information. MABAC, in the experiments described, was able to discover clusters with different shape and density. Comparisons were made with several other clustering algorithms from diverse clustering approaches (OPTICS, CHAMELEON, Matlab Fuzzy C-means). MABAC is also employed in chapter 5 in a data mining system to identify movement of patients.

MABAC has proved useful in some applications but not all of them. Some possible future work about MABAC, includes:

- Speed up MABAC by optimizing the algorithm with sparse matrix and/or graph

theory.

- Try parallel computing, so that MABAC can be applied to much larger datasets.
- Apply MABAC in : 1) microarray data analysis, because clustering has already been proved to be very useful in this task; 2) protein/DNA sequences clustering and extending it to create a multiple sequences alignment method.

CHAPTER 3

PROTEIN SEQUENCE CLUSTERING METHOD: SEQOPTICS

Protein sequence clustering has been widely used as a part of the analysis of protein structure and function. In this chapter, we demonstrate an approach to clustering proteins, SEQOPTICS. SEQOPTICS is based on OPTICS (Ordering Points To Identify the Clustering Structure) [7], an attractive approach due to its emphasis on visualization of results and support for interactive work. SEQOPTICS gets its initial “SEQ” because of its use in performing protein sequence clustering. SEQOPTICS is tested on data sets gathered from different web-based data sources [10, 8]. Visualization of the sequence clustering structure is demonstrated. Results are evaluated by comparison with two other available methods using the Jaccard coefficient, precision and recall. This system is open for future optimization including algorithm optimization and parallel computing for system speed-up.

3.1 Introduction

Extracting useful information from biological sequences is becoming an increasingly important problem due to rapid growth of public databases [107]. Among biological sequences, protein sequences are an especially interesting category, due to their being essential to life and endowed by a rich alphabet (20 amino acids).

3.1.1 *Public protein databases*

As additional protein sequences become available and computational methods improve, we can begin to better understand protein structure and function. There are several well-known protein databases discussed here:

- The Pfam [10] web site, developed by Sanger Institute, is a large collection of multiple sequence alignments and many common protein domains and families. Pfam is a database of two parts, the first is the curated part of Pfam containing over 7973 protein families. To give Pfam a more comprehensive coverage of known proteins Pfam-B contains a large number of small families taken from the ProDom database that do not overlap with Pfam-A. Although the quality of Pfam-B families is lower than Pfam-A, they can be useful when no Pfam-A families are found.
- The Protein Information Resource (PIR) [107] database is a semi-automatic protein family database aiming to be accurate and comprehensive in the public domain. Protein family classification is one of the main advantages of PIR.
- The National Center for Biotechnology Information (NCBI) [11] protein database is a collection of protein sequences from a variety of sources, including SwissProt, PIR (Protein Information Resource), PRF (Protein Research Foundation), PDB (Protein Data Bank), and translations from annotated coding regions in GeneBank and RefSeq. It is probably the largest popular protein database in the world.
- The Swiss-Prot website [8] is a protein sequence database that provides a high

level of annotation, a minimal level of redundancy and a high level of integration with other databases. It is maintained collaboratively by the Swiss Institute for Bioinformatics (SIB) and the European Bioinformatics Institute (EBI).

In this chapter three of these collections are used as follows: two data sets from Pfam, one from Swiss-Prot, due to its containing most extant protein sequences and popularity in research circles, and one from NCBI protein databases, for the same reasons as Swiss-Prot. The database we do not use is listed so the reader can ascertain another rich set of data worthy of exploration.

3.1.2 Sequences clustering methods

The goal of clustering protein sequences is to achieve a biologically meaningful partitioning [3]. Clustering a large set of protein sequences offers several advantages: Proteins are usually grouped into families based on the sequence similarity clustering, which provides some clues about the general features of that family and evolutionary evidence of proteins. Clustering also helps to infer the biological function of a new sequence by its similarity to some function-known sequences; Moreover, protein clustering can be used to facilitate protein 3-dimensional structure discovery, which may be very important for understanding protein function.

Recently developed clustering methods that have been successful in clustering a large number of sequences simultaneously include but are hardly limited to:

- ProClust (Protein Clustering) [81] uses a graph based approach combined with Hidden Markov Models (HMMs). It produces good results very favorably with PSI-BLAST [4] with data set containing both SCOP (Structural Classification

of Proteins) [75] and Swiss-Prot.

- SYSTERS (SYSTEmatic Re-Searching) [62, 63] takes into account the topology of the sequence space induced by the data itself to construct a biologically meaningful partitioning. SYSTERS overcomes the problem of an asymmetric distance matrix by computing a local pairwise alignment after performing a BLAST (Basic Local Alignment Search Tool) [3] search. It results in a hierarchical clustering with about 1 million sequences which can be browsed at <http://systers.molgen.mpg.de/>.
- GeneRage [32] is a fast method for clustering large protein data sets. GeneRage represents all similarity relationships within the dataset in a binary matrix. Removal of false positives is achieved through subsequent symmetrification of the matrix using a SmithWaterman dynamic programming alignment algorithm. Detection of multi-domain protein families and further false positive relationships within the symmetrical matrix is achieved through iterative processing of matrix elements with successive rounds of SmithWaterman dynamic programming alignments. Recursive single-linkage clustering of the corrected matrix allows efficient and accurate family representation for each protein in the dataset. GeneRage can quickly and accurately cluster large protein datasets into families.
- ProtoMap [108] offers an exhaustive classification of all the proteins in the Swiss-Prot and TrEMBL databases, into groups of related proteins. The analysis uses transitivity to identify homologous proteins, and within each group, every

two members are either directly or transitively related. Transitivity is applied restrictively in order to prevent unrelated proteins from clustering together. The classification is done at different levels of confidence, and results in a hierarchical organization of all the proteins.

- BLASTClust is part of the famous BLAST package [3]. It automatically and systematically clusters protein or DNA sequences based on pairwise matches found using the BLAST algorithm in case of proteins or Mega BLAST algorithm for DNA. BLASTClust finds pairs of sequences that have statistically significant matches and clusters them using single-linkage clustering.
- BAG (Biconnected components and Articulation points based Grouping of Sequences) [60] is a sequence clustering algorithm based on graph theory. It clusters sequences using two properties of an input graph, biconnected component and articulation point. BAG is claimed well suited for comparing a large number of proteins from multiple genomes.

Among these protein sequence clustering methods, the simplest and most widely used categories are based on hierarchical clustering algorithms (single linkage) [89]. Single-link aggregates all the sequences linked by a level of similarity above a given threshold, so that within a cluster any sequence is linked to at least one other sequence. This approach may yield fairly good results, but often a majority of sequences are grouped into one single huge cluster resulting from a massive chain effect due to multi-domain proteins. The BLASTClust program, one part of BLAST package from NCBI, is an example of single linkage protein sequence clustering

(<ftp://ftp.ncbi.nlm.nih.gov/blast/documents/README.bcl>). Another category, graph-based clustering algorithms, are also commonly employed for the clustering quality. BAG [60] is a sequence clustering algorithms based on graph theory and is web available at <http://bio.informatics.indiana.edu/sunkim/BAG/>. We use BLASTClust and BAG in this paper, the other protein clustering methods are also listed for future possible explorations.

3.2 OPTICS: Background

OPTICS (Ordering Points To Identify the Clustering Structure) [7] is a density-based clustering method [86] and is popular because it orders the data into a density-based clustering structure corresponding to a broad range of parameter settings.

3.2.1 Definitions

Some definitions referenced from the original paper [7] might help us understanding how OPTICS, or similarly DBSCAN (see appendix of chapter 1), works. We discuss them here with the examples shown in Figure 3.1 and Figure 3.2.

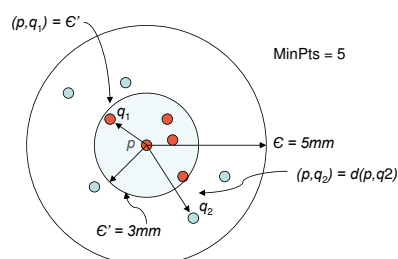


Figure 3.1: OPTICS: Core Point and Reachability Distance

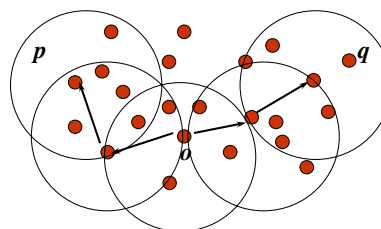


Figure 3.2: Example about OPTICS Core Objects and Core Distances

- *Core object*: An object p is in the ε -neighborhood of q if the distance from p to q is less than ε ; A *Core object* has at least $MinPts$ neighbors in its ε -neighborhood. In Figure 3.1, p is a *core object* with $MinPts$ set at 5 and ε at 3mm, while $q2$ is not a *core object* under these conditions.
- *Core-Distance*: The Core-Distance of an object p is the smallest ε -value that makes p a core object. If p is not a core object, the core-distance of p is undefined (a big value). In Figure 3.1, the *Core-Distance* of p is 3mm with $MinPts$ set at 5 and ε as 5mm. The *Core-Distance* of p is undefined with $MinPts$ set at 5 and ε at 3mm.
- *Reachability-Distance*: The Reachability-Distance of an object q with respect to another object p is the greater value of *Core-Distance* of p and the distance between p and q . If p is not a core object, the *Reachability – Distance* between p and q is undefined. Notice here two objects are involved in the definition of *Reachability-Distance*. In Figure 3.1, given $MinPts$ at 5 and ε at 5mm, the *Reachability-Distance* of $q1$ to p ($RD(q1, p)$) is the *Core – Distance* of p , 3mm, and $RD(q2, p)$ is the distance between $q2$ to p .
- *Reachability-Plot*: A reachability plot (see Figure 3.4) is a bar chart that shows each object's *Reachability-Distance* in the order the object was processed which demonstrates the cluster structure of data. Intuitively, OPTICS proceed those *CoreObjects* that satisfy some density condition ($MinPts$ and ε). Objects are processed according to the *Reachability-Distance* along those *Core object*. In Figure 3.2, clustering goes from *Core object* o to *Core object* p and q according

to the *Core-Distance*.

3.2.2 *Extracting clusters*

In OPTICS, a cluster is a set of *density-connected* objects which is maximal with respect to density-reachability. The final clusters can be extracted by either ε -cutoff or steepness of the plot. We use the ε -cutoff method in SEOPTICS, which is a Density-Based cluster extracting. For more detailed information about OPTICS algorithm, please refer the original paper [7].

OPTICS is a good solution to density-based cluster ordering and can be viewed as DBSCAN [86] with a broad range of parameters (all cases with ε in DBSCAN less or equal than ε in OPTICS are covered). For density-based methods, it is difficult to decide the input parameters that the algorithm is sensitive to. Although it does not produce clusters explicitly, OPTICS generates an augmented ordering of data sets representing its density-based clustering structure, and this structure (look ahead to Figure 3.4-3.7) can be visualized. Since OPTICS does not limit cluster extraction to global parameters, it is possible to extract cluster information interactively as well as automatically.

3.2.3 *Protein distance measures*

In typical protein sequence clustering a suitable distance measure is needed. Some functionally related sequences share little or no discernible sequence similarity and detection of these relationships is difficult. An easy measure to compute, which we used and will describe in more detail momentarily, is based on a pair-wise sequence

similarity known as the Smith-Waterman method [90]. Other distance measures, based on BLAST [3] or FASTA [80], are also very commonly used.

How to evaluate clustering quality is important to any proposed clustering scheme. For two-dimensional data, it is clear that we can plot the data and read the distribution to tell how good the clustering results are. But in high dimension data or sequence clustering, direct visualization is normally not feasible [45]. In protein sequence clustering, a popular measure of clustering quality is based on how well the clusters established by the clustering algorithm match extant protein families identified by biological experts [62] and residing within a widely accessible database. We choose this approach and another method which compares SEQOPTICS results with those of other protein clustering methods (BLASTClust and BAG) using validation criteria such as *Jaccard Coefficient* as described by Halkidi [45].

3.3 Methods

In the following paragraphs we first describe how SEQOPTICS works. Then SEQOPTICS is tested with several biological data sets. Visualization results of the clustering are presented. Moreover, the clustering results are analyzed according to the protein families identified by biologist and the results are also compared with those of two existing methods, BLASTClust and BAG. Our results demonstrate that SEQOPTICS performs better in terms of clustering quality. Some future work that needs to be done with our method includes system speed-up and algorithm optimization.

3.3.1 SEQOPTICS overview

SEQOPTICS expands the use of OPTICS to a protein sequence analysis method, where it has not been used. Figure 3.3 shows an overview of SEQOPTICS “in action.”

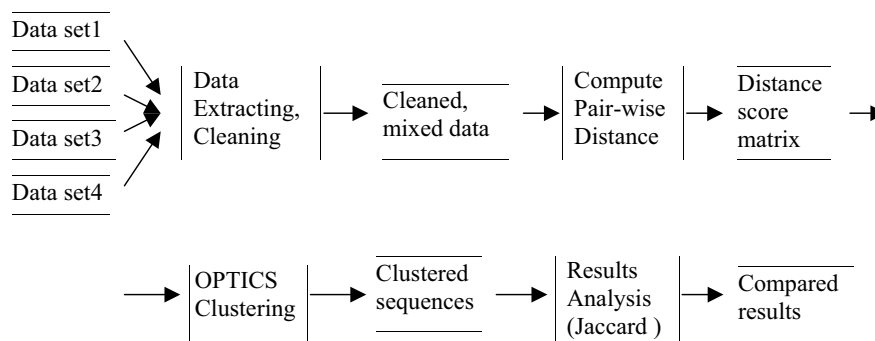


Figure 3.3: SEQOPTICS Overview

A set of extracted protein sequences (Data set) from data repository start an action leading to clustered sequences which are forwarded to the Jaquard-based results analysis.

- First, data sets are created by extraction of a collection of protein sequences from one or more families in public databases such as Pfam, Swiss-Prot and NCBI (already mentioned above). Shuffling the sequences within and across the family derivatives creates an input databank for testing purposes. The number of proteins in the input file is over 1000.
- Second, pairwise distances between the proteins are computed. A normalized form of the Smith-Waterman score is used, which we soon describe. This produces a matrix of pairwise distance scores (the Distance score matrix in Figure 3.3).
- Third, the OPTICS algorithm, in our adopted form, executes clustering. A clustering structure is graphically presented from which clusters are extracted.

- Fourth, the clustering results are analyzed and compared to results from other methods using *Precision*, *Recall*, and the *Jaccard coefficient*, a popular measure of clustering quality [45].

3.3.2 Data sets

The data sets, extracted from protein repositories as outlined in Section 3.1, are shown in more detail in Table 3.1. Two of them are from Pfam since it is a popular protein families database. Pfam multiple alignments come in two forms. In the first, “seed” alignments are representative, non-redundant sets of sequences that are checked reasonably carefully in a manual alignment editor. In the second, “full” alignments are HMM-generated automatic alignments of every homologous domain [10]. Two other data sets are from NCBI and Swiss-Prot.

Each protein sequence is labelled by a notation that indicates its origin: these labels define the assumed “true” cluster that we wish to retrieve from the mixed protein sequence file. For example, a sequence extracted from the “IGA1” family from the Pfam databank, is labelled as such and it is assumed that our clustering is correct if it assigns the sequence to the originating “IGA1” cluster.

Table 3.1: Data Source

Data set	1	2	3	4
From	Pfam (197)	Pfam (268)	NCBI (319)	Swiss-Prot (295)
Families	cytoB(75) GABAR(54) bac_globin(51) glucokinase(17)	bac_globin(51) IGA1(98) band3(119)	cytoC(86) GABAR(44) GAPDH(47) GFAT(78) GPCR(64)	GAPDH(122) casein kappa(62) globin (111)

Note: The number in parenthesis is the number of sequences in each family

- Data set 1(see Table 3.1) contains 197 protein sequences from four different families of Pfam database: 75 sequences of cytochrom_B561 (cytoB), 54 sequences of GABA Receptor(GABAR), 51 sequences of bac_globin, and 17 sequences of glucokinase.
- Data set 2 contains 268 sequences of three families of globin superfamily from Pfam database: bac_globin containing 51 sequences, IGA1 containing 98 sequences, and band_3_cytochrome(band3) containing 119 sequences.
- Data set 3 contains 319 sequences from five families of NCBI: 86 cytochrome C(cytoC) sequences, 44 GABAR sequences, 47 GAPDH sequences, 78 GFAT sequences, and 64 GPCR sequences.
- Data set 4 contains 295 sequences of three families from Swiss-Prot database including: 122 GAPDHs, 62 casein kappas, and 111 globins.

Of course, after the protein sequences from different families are mixed and shuffled, the original clustering is eliminated and the stage is set for seeing if the clustering method can reconstruct it.

3.3.3 Computing distance

Our approach, consonant with others [32, 31, 81], starts with a distance measure. When data sets are from different protein families, it is a common practice to use a normalized pairwise local alignment score by Smith-Waterman dynamic programming algorithm [90]. There are several parameters in Smith-Waterman, for example, scoring matrix, open gap penalty and extending gap penalty. Different scoring matrices

including BLOSUM50 (BLOcks SUBstitution Matrix) and PAM250 (Point Accepted Mutation) have been tried. BLOSUM50, which is also used in FASTA (FAST All) [80], is used in this paper. The open gap penalty taken is 12 and the extending gap penalty is 2. The final similarity score between two protein sequences is then normalized by the following:

$$SN(a, b) = \frac{S(a, b)}{\text{Min}(S(a, a), S(b, b))} \quad (3.1)$$

where $S(a, b)$ is the Smith-Waterman local alignment score between two sequences a and b ; $S(a, a)$ is the score of sequence a to itself; $S(b, b)$ is the score of sequence b to itself; and $SN(a, b)$ is the normalized score. The distance between two protein sequences is defined as:

$$\text{Distance}(a, b) = 1 - SN(a, b); \quad (3.2)$$

This form of the score is called normalized because every distance score lies between 0 and 1. Other distance measures can also be adjusted in a similar manner.

3.3.4 OPTICS clustering

The OPTICS method is briefly described in Section 3.1. The core part of OPTICS is implemented in JAVA according to the psuedo-code in the paper [7]; the resulting implementation is called SEQOPTICS. We applied it to the protein sequences of the sample data, yielding the clustered sequences. The core OPTICS part was tested with the data sets from the author [7]. Two parameters need to be chosen, ε and $MinPts$. In this chapter, since the distance between any two protein sequences is between 0 and 1, we can use a single ε for all data set, for example, set ε as 0.99, which is slightly smaller than 1. The $MinPts$ used here is 10 just for experimental

purposes. For the whole protein database, ε can use 0.95–0.99. Since the average number of sequences in a family is around 30, setting *MinPts* as 10 should catch most or all families.

There are two main advantages to applying OPTICS in protein sequences clustering analysis: 1) OPTICS can find a local density region; 2) OPTICS produces an ordering of the database representing its density based clustering structure and this ordering can be visualized, for example, in a *reachability plot*. The cluster ordering actually contains information about every cluster, i.e., OPTICS enables the extraction of not only “traditional” cluster information, but also clustering structures intrinsic to the original data groupings (see Figure 3.4-3.7).

3.4 Results

We describe results in the following subsections:

- First, visualization results of the ordering of each data set, as depicted in overview diagram (Figure 3.3), are presented. These provide clues about clustering structure.
- Second, the density-based cluster sets are extracted from the ordering reachability distance plot by a cutoff value, which is decided by the user.
- Third, the clustering results are validated, by comparing the SEQOPTICS developed clusters and the original clusters in the originating database. The principle of true- and false-positives, etc., is used, along with the *Jaccard coefficient*.

To judge the resulting clustering set’s biological accuracy, we need to compare it to a “true” cluster set. However, there is no generally accepted “true” cluster set.

All automatic protein clustering methods are based on “all against all” sequence comparison (each sequence is compared to all other sequences). Real clusters need to be verified by biological expertise. Since it is impossible to have “real” clustering, we have to assume the original database clusters are the “real” clusters. This is the way most automatic protein clustering does. For example, we assume the sequences from the *glucokinase* family of Pfam are in the same cluster.

3.4.1 Visualization of the cluster structure

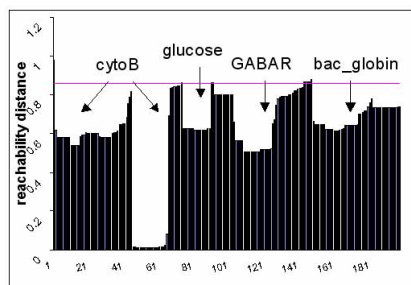


Figure 3.4: Data Set 1 (Pfam)

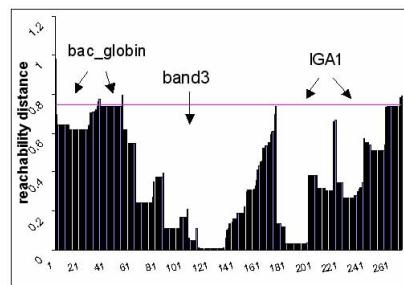


Figure 3.5: Data Set 2 (Pfam)

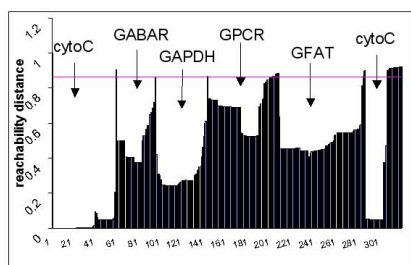


Figure 3.6: Data Set 3 (NCBI)

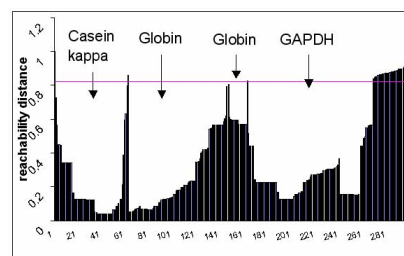


Figure 3.7: Data Set 4 (Swiss-Prot)

A reachability distance plot is provided by SEQOPTICS for each data set. In a

reachability distance plot, the horizontal axis represents a sequence ordering and the vertical axis represents the reachability distance itself. A valley in the plots serves as feature we can use as a preliminary candidate of a cluster set.

- For data set 1, there are five apparent valleys, as seen in Figure 3.4: The first two valleys are composed of sequences from cytochrom_B562; The third valley consists of sequences from glucokinase; The fourth valley contains sequences from GABAR family; The fifth valley are sequences from bac_globin family.
- For data set 2, there are three valleys, as seen in Figure 3.5: The first one is composed of sequences from bac_globin; The second valley is composed of sequences from band3 family; The third valley contains only sequences from IGA1.
- For data set 3, there are six valleys as seen in Figure 3.6: The first one and last one contain only cytoC sequences; The second valley contains only sequences from GABAR; The third valley contains sequences GAPDH; The fourth valley contains GPCR sequences; The fifth valley contains only GFAT.
- For data set 4, there are four main valleys as seen in Figure 3.7: The first valley contains only casein kappa sequences; The second and third valley contain only globins; the fourth valley is composed of GAPDHs.

From these figures, we can ascertain that each valley contains exclusively one sequence family. However, each family may contain more than one cluster. In most existing clustering methods, proteins from the same family tend to be split into different clusters. However, it is not common that the sequences from different families are

put in the same cluster. This will be explained later in the results evaluation.

3.4.2 Extraction of the final clusters

The algorithm of extracting clusters is described in the OPTICS paper [7]. The final density-based clusters were extracted by using a cutoff value decided by the user. In SEQOPTICS, the sequence starting a valley with *reachability distance* higher than the cutoff is deemed to be in the same cluster as the rest of the sequences in the valley. Any sequence with *reachability distance* higher than the cutoff is deemed to be noise if it does not start a new valley.

Wrapping up these observations, and continuing to apply them to the other clusters, we can argue:

- In Figure 3.4, there are four clusters, the cutoff value being set at 0.865.
- Similarly, there are four clusters in Figure 3.5 given cutoff 0.745,
- six clusters arise in Figure 3.6 given cutoff 0.860, and
- three clusters emerge in Figure 3.7 given cutoff 0.820.

3.4.3 Validation of the cluster set

To judge the resulting cluster sets with respect to its biological accuracy, the following problems need to be addressed:

- There is no generally accepted “true” cluster set. That is to say, those “true” clusters are always “biased”. However, if appropriate data source is chosen, then the “bias” can be limited.

- There are some automatically generated cluster sets and some manually generated cluster sets. Those cluster sets are usually organized in “families,” thus make the validation easier.

Automatically generated cluster sets are not necessarily biologically correct. They are normally based on all-against-all sequence comparisons. Pfam is an example of this category. Pfam is a large collection of multiple sequence alignments and hidden Markov models covering many common protein domains and families based on the UniProtKB/Swiss-Prot Protein Knowledgebase. Pfam seeds contain the seed alignments of the families and thus are more accurate than general Pfam families. In this thesis Pfam seeds were used for testing to reduce “bias” of “true” cluster.

Manually generated cluster sets often contain only well-characterized sequence stretches (representing folds or domains) instead of full-length sequences (e.g., SCOP). They are normally also based on sequence comparisons with subsequent manual interaction (e.g. PIR).

NCBI is probably the most complete protein sequences database. UniProtKB/Swiss-Prot provides a high level of annotation (such as the description of the function of a protein. SEQOPTICS data are extracted from NCBI and SwissProt since they are probably two most complete databases in biological research. Those extracted data are further manually processed, i.e., we select those protein similar in annotation and sequences so that “bias” is reduced.

As it has been mentioned earlier in this paper, we treat the original database cluster as the “true” cluster against which the algorithm derived clusters are judged. Based on this assumption, several statistics might be used to evaluate the result.

Some of them were discussed as below.

A cluster set of n data points from the experiment can be represented by the formula $m = \frac{n*(n-1)}{2}$ values in a triangular matrix M , where for $i < j$, $M_{ij} = 1$, if and only if i and j are in the same cluster and $M_{ij} = 0$ otherwise. If T is a matrix of “true” clusters, the two cluster sets (“true” and “experimental”) can be compared based on the following numbers:

- a is “true positive,” i.e., the number of sequence pairs clustered together in both sets, which can be defined as: $a = |(i, j)|M_{ij} = 1 \wedge T_{ij} = 1, i < j|$
- b is “false negative,” i.e., the number of sequence pairs clustered together in the true cluster set, but not in the current clustering solution, defined as: $b = |(i, j)|M_{ij} = 0 \wedge T_{ij} = 1, i < j|$
- c is “false positive,” i.e., the number of sequence pairs clustered in the current solution, but not in the true cluster set, defined as: $c = |(i, j)|M_{ij} = 1 \wedge T_{ij} = 0, i < j|$
- d is “true negative,” i.e., the number of sequence pairs not clustered in either current solution or the true cluster set, defined as: $d = |(i, j)|M_{ij} = 0 \wedge T_{ij} = 0, i < j|$

There are many validation techniques as cited in reference [45] . Here we use three parameters based on the above: *Precision*, *Recall* [6, 104], and *Jaccard Coefficient* [52].

Precision is defined as:

$$P = \frac{a}{(a + c)} \quad (3.3)$$

Recall is defined as:

$$R = \frac{a}{(a + b)} \quad (3.4)$$

Jaccard Coefficient is defined as:

$$S = \frac{a}{(a + b + c)} \quad (3.5)$$

All three parameters have value between 0 and 1. The better the clustering, the bigger the values. In a perfect clustering which is identical to the true cluster, $P = 1$, $R = 1$, and $S = 1$. Most existing sequence clustering methods perform well in terms of *Precision* but not in *Recall*. This is shown in the (later) validation result given in Table 3.2 along with additional comparative values.

$T =$	A	B	C	D	$M =$	A	B	C	D	$a = \{B, C\} = 1$
	A	1	1	0		A	0	0	1	$b = \{A, B\}, \{A, C\} = 2$
	B		1	0		B		1	0	$c = \{A, D\} = 1$
	C			0		C			0	$d = \{B, D\}, \{C, D\} = 2$
	D					D				

Figure 3.8: Comparison of Two Cluster Sets T and M

By counting those sequence pairs clustered in the same way and those clustered differently in T and M , the *Jaccard Coefficient* is: $S(T, M) = \frac{1}{(1+2+1)} = 0.25$. the *Precision* is: $P(T, M) = \frac{1}{(1+1)} = 0.5$, the *Recall* is: $S(T, M) = \frac{1}{(1+2)} = 0.33$

We clustered the same data sets with two other clustering methods, BLAST-Clust [3] and BAG [60], using default parameters of these methods. BAG is a graph based clustering method and most existing methods are based on graph clustering. BLASTClust is chosen because it is from NCBI BLAST package. This hierarchical sequence clustering method is very popular in biological research.

Again using the *Jaccard Coefficient*, *Precision* and *Recall*, we developed comparison values shown in Table 3.2. From Table 3.2, we see that SEQOPTICS produces good results relative to each original cluster set in terms of *Jaccard Coefficient*. Every SEQOPTICS *Jaccard Coefficient* is higher than 0.65 and the highest being 0.85. It is also seen in the table that SEQOPTICS outperforms BAG and BLASTClust on all the data sets chosen on this criterion. The performance with BAG exceeds BLASTClust for the same reason. However, BAG and BLASTClust tend to give more clusters than the “true” clusters, explaining why the *Precision* of those two methods on all data sets are 1. But neither of these two performs well in terms of *Recall*. Overall, SEQOPTICS performs better than BAG and BLASTClust and seems a promising method in terms of both clustering quality coupled with its graphical representation of clustering structure.

Table 3.2: Comparison of Clustering Results

Data Set	<i>BLASTClust</i>			<i>BAG</i>			<i>SEQOPTICS</i>		
	<i>S</i>	<i>P</i>	<i>R</i>	<i>S</i>	<i>P</i>	<i>R</i>	<i>S</i>	<i>P</i>	<i>R</i>
1(Pfam)	0.05	1.00	0.05	0.27	1.00	0.27	0.83	0.99	0.84
2(Pfam)	0.04	1.00	0.04	0.20	1.00	0.20	0.85	0.98	0.87
3(NCBI)	0.11	1.00	0.11	0.60	1.00	0.60	0.66	0.82	0.78
4(Swiss-Prot)	0.06	1.00	0.06	0.50	1.00	0.50	0.81	0.99	0.82

The clustering results of 4 data sets by three methods according to three parameters: *S*(*Jaccard Coefficient*), *P*(*Precision*), *R*(*Recall*) show that SEQOPTICS outperforms BAG and BLASTClust. BAG and BLASTClust tend to give more clusters than “true” clusters. Take Pfam1 as an example, SEQOPTICS gives 4 clusters, BAG results in 24 clusters and BLASTClust gives 121 clusters. Therefore, BAG and BLASTClust give high *Precision* values and low *Recall* value.

Although manual cluster sets combined with biological experiment and the experts’ information are the ultimate validation criterion, computer-evaluation can be

considered a tool at the disposal of experts in evaluating clustering results.

3.5 *Conclusion and Future Work*

In this chapter we described a programmed system, SEQOPTICS, for protein sequences clustering whose steps are shown in Figure 3.3. A core portion (phase) of the system is based on OPTICS clustering and visualization, which we believe is being used here for the first time in protein sequence clustering. Prior to this phase, it is necessary to compute a distance measure between (protein) sequences. A normalized Smith-Waterman score is used in this paper to compute the required distance. The final system phase, Results Analysis (Figure 3.3), demonstrates adequacy of our approach for small-scale data and the usefulness of the cluster structure visualization.

According to Ankerst [7], one good feature of OPTICS is that it is unnecessary to limit oneself to a single set of global parameters. An augmented cluster ordering contains information equivalent to density based clusterings corresponding to a broad range of parameter settings; as such, the cluster ordering is a versatile base for both automatic and interactive cluster analysis. A second good feature lies in the visualization of the data set distribution. Depending on data set size, one can either represent the cluster-ordering graphically for a small data set or employ an alternate technique (appropriate) for large data sets. In this paper, we demonstrated that the visualization of cluster structure in SEQOPTICS is meaningful.

The data sets used in this chapter tend to be small and SEQOPTICS has performed well on them. However, for large data sets such as the whole collections of sequences, we believe that there is a need for improvements, we suggest:

- Apply parallel computing tools, specifically forged to provide processing speed in the presence of large scale computation, e.g., the Message Passing Interface (MPI) [41].
- Implement visualization techniques that scale up well as the datasets become larger. For example, it is difficult to visualize the clusters structure of a data set with 1000 clusters in one screen. Therefore, we need to improve the visualization technique.
- Use some other distance measure for protein sequence distance, e.g., BLAST or FASTA, reputed to be faster than most competitive methods.

Each of these directions offer short- and long-term opportunities for development of a sequence clustering system. This part of dissertation thus provides a sample system for future development.

CHAPTER 4

PROTEIN ALLOSTERIC NETWORK ANALYSIS FOR LIGAND GATED ION CHANNELS

A protein allosteric network structural analysis system is constructed which embeds clustering analysis in a context that conceptually progresses through several stages: multiple sequences alignment, coupling analysis and covariance analysis, clustering analysis, and 3-D modelling; By analyzing amino acid covariance in multiple sequence alignments, an energetically interconnected network in the cys-loop family of ligand-gated ion channels (LGICs) has been identified. Statistical coupling and correlated mutational analysis along with clustering have revealed a highly coupled cluster. Mapping the positions in the cluster onto a 3-D structural model demonstrates that these highly coupled positions form an interconnected network linking experimentally identified binding domains through the coupling region to gating machinery.

4.1 Introduction

Protein structure prediction has been a challenging problem for years. For allosteric proteins, movement of one residue (for example, triggered by another molecule) impacts other remote residues. Accordingly, it is very difficult to detect a protein's detailed structure since it is formed as interconnected residues whose movements constitute a dynamic network. However, by examining available subunit sequences of ligand-gated ion channels (LGICs) from different species, it appears possible to find potential coupling pairs of residues for further experimental validation. LGIC

is chosen as a target for system work based on our previous work on the GABA receptor [19], one member of the LGIC family.

4.1.1 Ligand Gated Ion Channels

LGICs mediate fast synaptic transmission for communication between neurons and there are five subgroups of cys-loop family in LGICs. Studies using site-directed mutagenesis, affinity labelling, cysteine accessibility test, and electron microscopy in the last two decades have demonstrated that all members of this receptor family have similar structural architecture [55]. Each receptor is comprised of five subunits. Each subunit has a large amino-terminal extracellular domain, which forms an agonist binding site, four transmembrane domains (M1-M4), which form an ion conduction pore, and a large intracellular loop, which can interact with intracellular proteins for receptor clustering and regulation [21].

LGICs are allosteric proteins in which binding of a neurotransmitter to a binding site in the extracellular amino-terminal domain controls distant gating machinery in the transmembrane domain to open the ion conduction pore. Such a kinetic mechanism of channel activation can be best described by an allosteric model in which agonist binding and channel gating are highly coupled [21]. A long range coupling of the agonist binding domain to the gating machinery requires an interconnected allosteric network, through which binding energy can be reliably transmitted in a form of a “conformational wave” from the agonist binding site to the gating machinery to open the channel. The information about this interconnected allosteric network however is not readily available by directly examining structural models.

The structure model of amino-terminal extracellular domain is further extended by the crystal structure of a homologous protein, acetylcholine binding protein (AChBP) [15]. The structural model of nicotinic receptor transmembrane domains is also available via electron microscopy at 4 Å resolution [55]. These structural models however are static. The detailed network formed by interconnected residues mediating channel function remains unclear.

4.1.2 Statistical coupling analysis

Statistical coupling analysis (SCA) is a sequence-based statistical method designed to estimate thermodynamic coupling of two residues in a protein. The basis of this method is that the coupling of two sites in a protein, either directly or allosterically, should cause these two positions to co-evolve. Such co-evolved residues can be identified by analyzing a large and diverse multiple sequence alignment (MSA) of a protein family for the distribution probability of 20 amino acid residues at each position [98]. With this method, the degree of residue covariance at two sites, in a form of “coupling energy”, can be determined by observing the effect of perturbation at one site (extracting a subset of sequence alignment with a relatively conserved residue) on the amino acid distribution of the other site.

Prediction of potential interacting residues could dramatically reduce the work of exhaustive mutagenesis scanning and facilitates identification of functionally important residues in the interconnected allosteric network of the protein for the mechanisms of binding-gating coupling of the entire family. This method has been successfully used to define interconnected allosteric networks of several protein families, such

as PDZ domains [68], G-proteins [48], G-protein-coupled receptors, serine proteases, globins [98], and retinoid X receptors [68].

4.1.3 McLachlan-based substitution correlation analysis

McLachlan-based substitution correlation (McBASC) is another approach to find co-variant positions in a protein family [39], although it is more frequently used to find direct contacting residues [36]. By comparing pairs of sequences in an MSA, this method assigns a score for each comparison at each position based on the change of amino acid residue properties for every pair of sequences. Correlation analysis (correlation coefficient) of these mutational scores between two sites from the MSA of a protein family then can identify co-evolved sites.

4.1.4 Clustering analysis

Clustering analysis plays an important role after statistical coupling analysis by providing possible clusters of coupled residues which are potential interacting residues. The described data mining system conceptually includes multiple sequence alignment, coupling analysis and covariance analysis, clustering analysis and 3-D modelling (see Figure 4.1). This computer system is very useful since it dramatically reduces the effort of exhaustive mutagenesis scanning and facilitates identification of a potential interconnected allosteric network underlying protein function.

In this chapter, based on two approaches described above, a group of genetically covariant sites of LGICs are clustered. Mapping these positions onto the 3D structural model of a nicotinic receptor subunit reveals that these positions are mainly clustered

in functionally important domains, forming an interconnected allosteric network linking agonist binding site to the gating machinery via coupling domains. In addition, these highly coupled positions are clustered in transmembrane domains, the recent focus for the sites of action of many allosteric modulators. This system reveals a genetically interconnected network of LGICs, which potentially serves as the activation pathway and plays an important role in allosteric modulation.

4.2 Methods

Figure 4.1 is the guideline to illustrate the computational steps. As a preliminary step, multiple sequences of LGICs were downloaded from a data source. These sequences are aligned into a profile “Multiple Sequences Alignment” (MSA). The MSA is then used for coupling and covariance analysis to produce a 2-D matrix for further clustering. Clustering produces a group of sites that compose an allosteric network which is mapped on 3-D modelling. The 3-D modelling results demonstrate an allosteric network of LGICs.

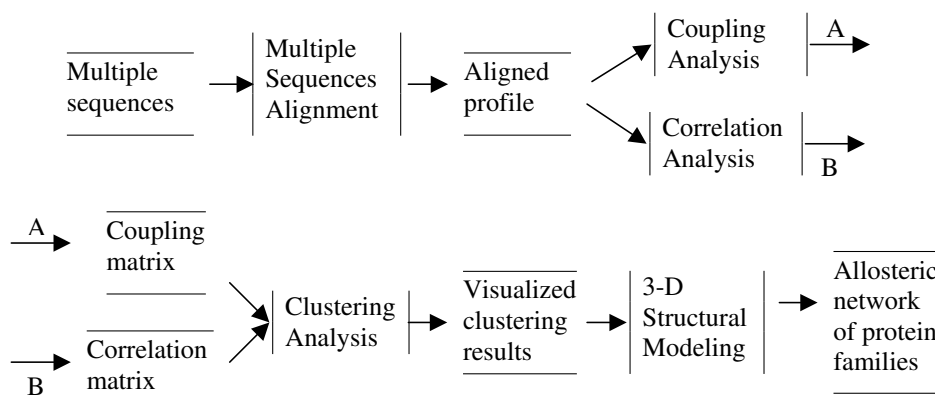


Figure 4.1: System Overview of Protein Allosteric Network Analysis (PANA)

4.2.1 Data source and multiple sequence alignment

The amino acid sequences of subunits in the cys-loop receptor family of ligand-gated ion channels were downloaded from the Ligand-Gated Ion Channel Database in European Bioinformatics Institute website where redundancy check has already been performed; the website is (<http://www.ebi.ac.uk/compneur-srv/LGICdb/LGIC.html>). Based on the length distribution histogram of all sequences (data not shown), those sequences that clearly do not belong to the same population are excluded. Extra long sequences (> 700 residues) could have different structure, and extra short sequences (< 250 residues) are likely incomplete sequences which would introduce un-natural gaps and influence coupling analysis (see Figure 4.3B). Thus, these extra long and short sequences were excluded for further analysis. The remaining 389 sequences were used for the analysis.

In the first stage, as shown in Figure 4.1's "Multiple Sequences Alignment" process (MSA), all sequences were aligned using Clustalw1.83 [97] package with default parameters (10.00 gap opening penalty, 0.20 gap extension penalty, and Gonnet series of the protein weight matrix). Since the structural model of the Torpedo nicotinic receptor is the best model available, for all calculations, the numbering in the α subunit of Torpedo californica nicotinic receptor was used, ignoring the signal peptide and gaps inserted into the subunit during the sequence alignment. The result of this stage, "Aligned profile," becomes the input into the second stage of processing, as shown in the figure.

4.2.2 Static and coupling energy calculation

In the second stage, static and coupling energy were calculated with the method provide by Suel [98]. The static energy, ΔG , for each position was calculated by the following equation:

$$\Delta G_i^{start} = kT^* \sqrt{\sum_x \left(\ln \frac{P_i^x}{P_{MSA}^x} \right)^2} \quad (4.1)$$

where kT^* is an arbitrary energy unit. The probability of any amino acid x at site i (P_i^x) was obtained by the binomial probability of the observed number of x amino acids given its mean frequency in all proteins. P_{MSA}^x is the probability of any amino acid x at all sites in the multiple sequence alignment (MSA). The full distribution of amino acids at a site i can then be characterized by a 20-element vector of all x . P_i^x was calculated by the following equation:

$$P(X) = \frac{N!}{n_x!(N - n_x)!} P_x^{n_x} (1 - p_x)^{(N - n_x)} \quad (4.2)$$

where N is the total number of sequences, n_x is the number of sequences with amino acid x at position i , and P_x is the mean frequency of amino acid x in all proteins from Swiss-Prot database. Stirling's approximation was used for the evaluation of large factorials.

Coupling energy between two sites, i and j , was calculated by perturbation analysis (see below) using the follow equation:

$$\Delta \Delta G_i^{start} = kT^* \sqrt{\sum_x \left(\ln \frac{P_{i|\delta_j}^x}{P_{MSA|\delta_j}^x} - \ln \frac{P_i^x}{P_{MSA}^x} \right)^2} \quad (4.3)$$

The perturbation at each site was made by extracting the sub-alignment containing the only amino acid residue with frequency above a cut off value (30% in this paper).

The basis of this method is that the coupling of two sites in a protein, either directly or allosterically, should cause these two positions to co-evolve. These co-evolved residues can be identified by analyzing a large and diverse multiple sequence alignment (MSA) of a protein family [98] since each family includes proteins from a variety of species in the phylogenetic tree.

The calculation of coupling energy for all sites created a 2-D matrix data set, recognized as the “Coupling matrix” in Figure 4.1. This part of the system is given a more detailed treatment in Section 4.3 with Figures 4.3 and visualizations are applied with the two outputs.

4.2.3 Correlated mutational analysis

Correlated mutational analysis was carried out using the McLachlan-based substitution correlation (McBASC) [39]. The program for this calculation, written in JAVA, was downloaded from Fodor’s website <http://www.afodor.net>, modified for formatted output, and executed under a local JAVA environment. Similar to the coupling analysis, the calculation of correlation analysis created a 2-D matrix data set, “Correlation matrix” in Figure 4.1. This part of the system is given a more detailed treatment in Section 4.3 with Figures 4.4 and 4.5 being visualizations.

4.2.4 Clustering analysis

The third stage of computation, “Clustering Analysis” was applied to the output matrix of the previous step to extract information from this data set. HCE (Hierarchical Clustering Explorer) provided by Seo [87] was used since it provides its users with

interactive visual feedback, which facilitates searching for network residues. HCE is basically an hierarchical clustering method with visualization features. The resulting file, “Visualized clustering results,” represents the terminal point of this round of computation.

MABAC [25] was also tried for its good quality of clustering in test cases as described in Chapter 2 of this dissertation. However, MABAC does not provide the visualization of the clustering structure. In the future, a tree-view method will be used to implement a visualization schema of MABAC.

4.2.5 *Visual presentation*

The clustering results produce high coupling sites used for subsequent processing. For visual presentation of the highly coupled residues in the structural model of α subunit, a homology model of the amino-terminal domain of the α subunit of the Torpedo nicotinic receptor was obtained by Swiss-Model using AChBP (pdb ID: 1I9B) as a template. The structure model of the transmembrane domains of the α subunit was downloaded from the Protein Data Bank with the pdb ID of 1OED. The amino acid residues with static energy or high coupling energy were mapped onto the above models using molecular graphics program Swiss-pdbViewer v3.7 (<http://us.expasy.org/spdbv/>). The resulting image was saved as a POV-Ray 3.5 scene file. The final image of the model was rendered by POV-Ray 3.6 software (<http://mac.povray.org/>). Those visualized sites provide possible coupling network composed of clusters of high coupling sites.

4.3 Results

4.3.1 Static energy

This phase of the system was implemented by the author in the JAVA programming language based on the Rama's paper [68]. To calculate the coupling, we started with counting occurrences of amino acids at each position in the MSA. Figure 4.2A shows the relative frequencies of amino acid residues in the cys-loop family of LGICs (open bar) and in all proteins from the Swiss-Prot database (filled bar) used for calculation. Note that the frequencies of amino acid residues in the MSA slightly deviate from those in all proteins. Hydrophobic residues, such as Leu, Ile, Phe, Val, and Trp in the MSA had slightly higher frequencies than average. Small and some hydrophilic residues such as Gly, Ala, and Lys had lower frequencies than average. This is expected for membrane proteins with multiple hydrophobic transmembrane stretches.

The amino acid frequencies at each site were then determined and converted to probabilities for all 20 amino acids using Equation 4.2. The probabilities then were used to calculate the static energy using Equation 4.1. If the amino acid distribution at a site is similar to the distribution for all positions, then the site is not conserved, and the static energy calculated by Equation 4.1 approaches zero. In contrast, if a site is conserved, its amino acid distribution will deviate from the mean, and the static energy at that site will be higher. Thus, the magnitude of the static energy represents the extent of deviation of the amino acid distribution at each site from the mean in MSA, and thus represents the extent of residue conservation at that site.

Figure 4.2B shows the static energy for all 437 positions using the numbering of Torpedo California nicotinic receptor α subunit. This static energy emanates from the “Coupling Analysis” process. Note that a stretch of positions toward the C-terminus has low static energy. This region corresponds to the large intracellular loop between M3 and M4, the most diversified region in this protein family.

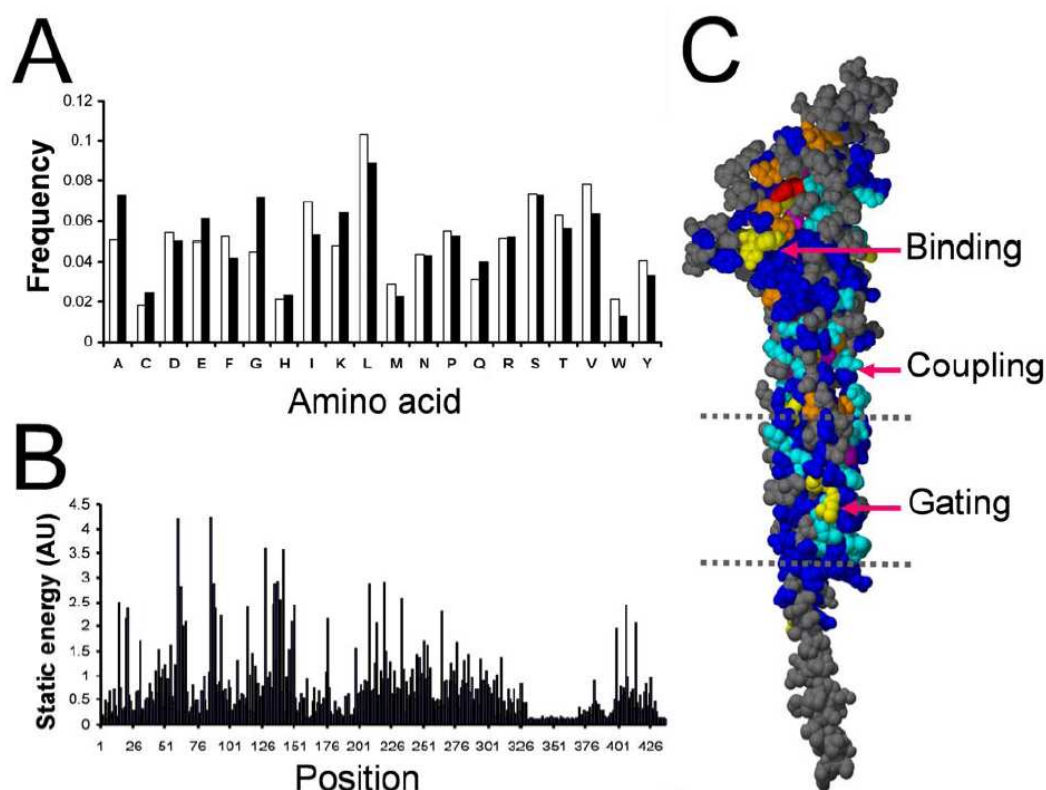


Figure 4.2: Static Energy Results

A. Amino acid frequencies in all proteins (filled bar) or in the LGIC’s MSA (open bar). B. Static energy (arbitrary unit) in all positions using Torpedo nicotinic receptor α subunit numbering. C. Mapping of static energy onto the EM structure of Torpedo nicotinic receptor α subunit. Scales are *gray* : < 0.5, *blue* : 0.5 ~ 1, *cyan* : 1 ~ 1.5, *yellow* : 1.5 ~ 2, *orange* : 2 ~ 2.5, *purple* : 2.5 ~ 3.0, *red* : > 3. Two dashlines represent two surfaces of the plasma membrane.

The static energy for all positions was then mapped onto the 3-D models of amino-terminal and transmembrane domains as in Figure 4.2C. The most conserved positions

(in red) are mainly located in the protein core near the binding site, and intermediately conserved positions are clustered in functionally important domains for binding, coupling, and channel gating as indicated by arrows.

4.3.2 *Coupling analysis results*

To calculate the statistical coupling energy, we performed perturbation analysis by extracting sequences from the Multiple Sequence Alignment (MSA) for each site containing conserved residue(s) ($> 30\%$). There are 253 positions meeting this criterion. Extracting sequences containing the conserved residue(s) at these positions (one position and one residue at a time) resulted in amino acid redistribution of all sites; this is called a perturbation. The amino acid probabilities at each site in extracted sequences were then determined and used for coupling energy calculation. If the perturbation at one site significantly changes the amino acid distribution at another site, then these two sites have high coupling energy. Otherwise, they have low coupling energy.

The calculation of the perturbation resulted in a 437×253 matrix of the coupling energy (Figure 4.3A). In some regions of the receptor, such as the large intracellular loop (301-402) between 3rd and 4th transmembrane domains with the most diversified sequences and low static energy (Figure 4.2B), the alignment generated large gaps at many positions. To determine whether gaps can influence the coupling result, we examined the relationship between the number of gaps at a position and the mean coupling energy of all positions in response to the same perturbation. Figure 4.3B plots the number of gaps against the mean coupling energy for each perturbation.

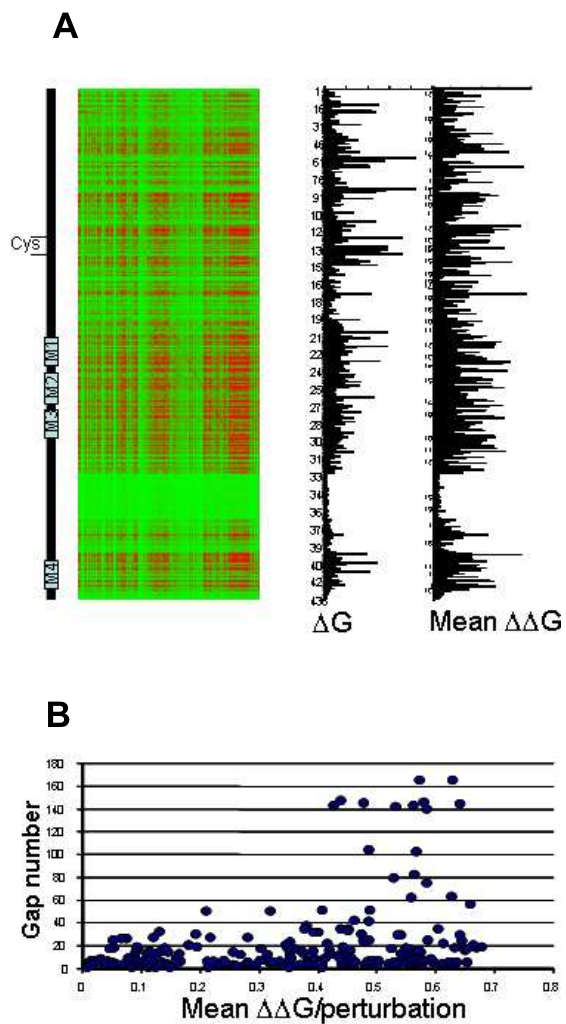


Figure 4.3: Coupling Energy and its Relationship to Inserted Gaps

A. Coupling energy in all positions (rows) with 253 different perturbations (columns). Cysteine loop and four transmembrane domains are shown on the left. The static energy (ΔG_i) and mean coupling energy (mean $\Delta\Delta G_i$) for all perturbations are plotted in corresponding positions. B. Relationship between mean coupling energy per perturbation and number of gaps at each site.

Note that all positions with more than 60 gaps in an aligned position have high mean coupling, suggesting the number of gaps has some influence in the coupling energy calculation. To avoid this potential influence, we discarded positions with more than 60 gaps and most M3-M4 intracellular loop (sites 296-392) for further analysis in both rows (coupling) and columns (perturbation). This resulted in a 311×219 2-D matrix, as the “Coupling matrix” appearing in Figure 4.1.

4.3.3 Correlated mutation analysis

Correlated mutation analysis is performed with the same set of MSA. This resulted in a 437×437 matrix. Similarly, to extract information from the large data set, we first removed positions with large gaps (> 60) and intracellular loop (296-392) to avoid potential influence of gaps and improper alignment.

4.3.4 Clustering analysis

A clustering analysis for both coupling and correlated mutation analysis is performed to identify highly coupled sites from the large data sets from the above. This clustering is depicted by Figure 4.1’s “Clustering Analysis.” Figure 4.4A shows the clustering result of coupling energy for this matrix with 219 rows (perturbation) and 311 columns. Examining this output is important for further processing. For example, the sites with high coupling energy are mainly clustered in the right bottom as indicated with three yellow boxes. Figure 4.4B is a closer view of these clusters. Positions of all columns in this highly coupled cluster showed a similar coupling pattern to many perturbations, suggesting they are covariant in response to same set of

perturbations and thus are mutually coupled. The detailed positions in Figure 4.4B are listed in Table 4.1.

Similarly, to extract information from the large data set of correlated mutation analysis, positions with large gaps (> 60) and intracellular loop (296-392) are removed to avoid potential influence of gaps and improper alignment. This resulted in a 316×316 matrix, which again were clustered using the HCE3.0 software. The results are shown in Figure 4.5A. Note that there is a high correlation coefficient cluster (yellow box, the right bottom corner box) from the large background. The details of this cluster with high correlation coefficient are shown in Figure 4.5B, and the positions in this cluster are listed in Table 4.1.

4.4 Results Discussion

In search for an activation pathway, we used two statistical analysis along with clustering to systematically identify the genetically interconnected positions in the cys-loop family of ligand gated ion channels. Highly coupled positions predicted by both methods overlapped by 70%. Mapping these positions onto the 3-D structural model demonstrated that these highly coupled positions were mainly clustered in important functional domains, linking binding site through coupling region to the gating machinery. Thus, our results suggest an interconnected network that may serve as an activation pathway, coupling agonist binding to channel function. The finding can be used as a guide for experimental design and facilitate elucidation of the activation mechanism for ligand-gated ion channels.

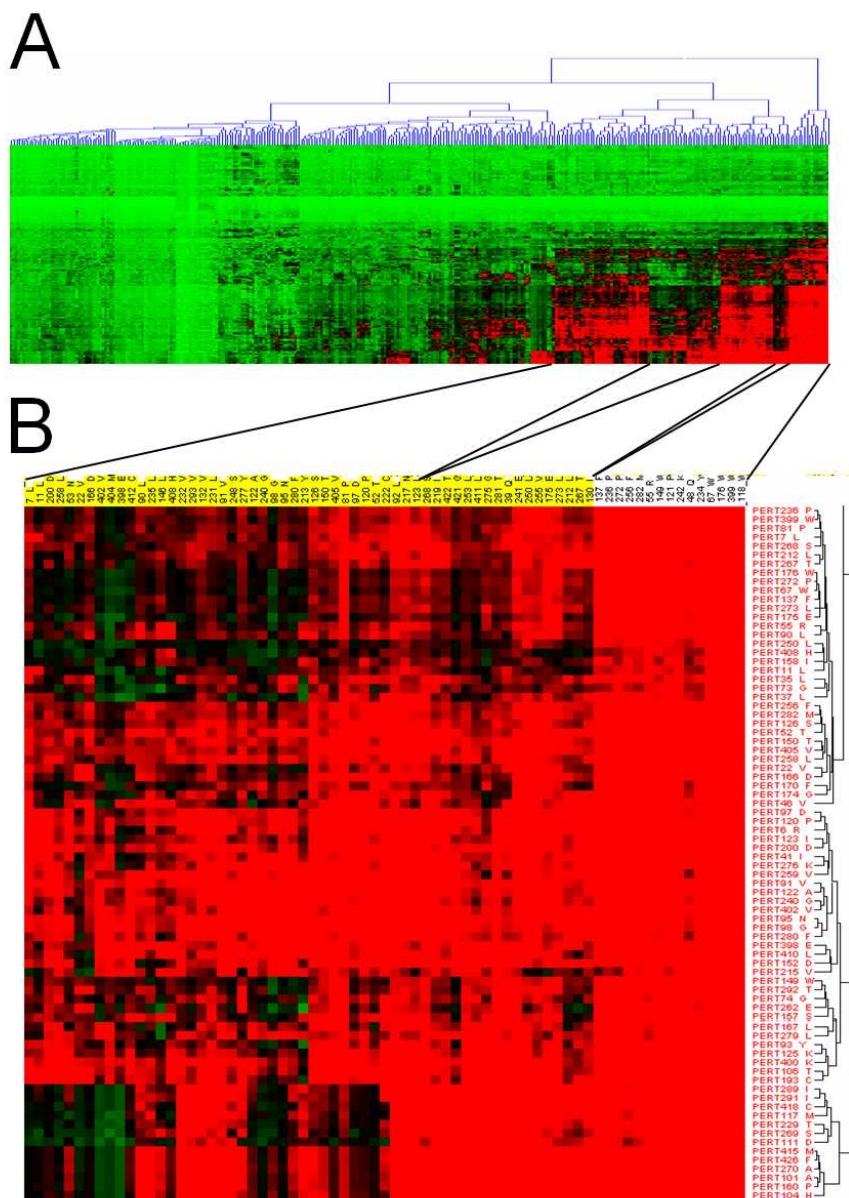


Figure 4.4: Coupling Energy Clustering Results

A. Clustering results of the coupling energy. Highly coupled clusters are highlighted in yellow boxes. B. Closer view of these highly coupled positions. The detailed sites in this cluster are listed in Table 4.1.

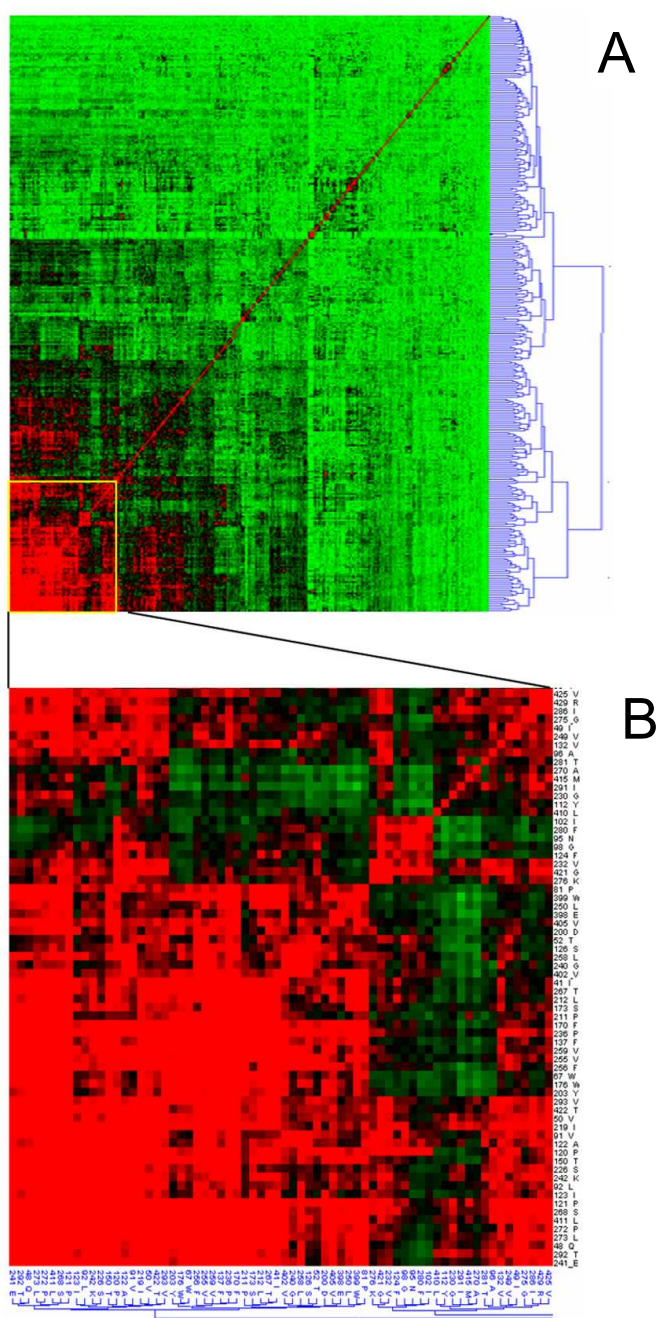


Figure 4.5: Correlation Coefficient Clustering Results
 A. Clustering results of the correlation coefficients. The cluster with high correlation coefficient is highlighted in a yellow box. B. Close view of the highly correlated cluster. The detailed sites in this cluster are listed in Table 4.1.

Table 4.1: Highly Coupled Sites Clustered by SCA or McBASC

SCA	McBASC	SCA	McBASC	SCA	McBASC	SCA	McBASC
7		121	121		226		276
11		122	122		230	277	
22		123	123	231		280	280
39			124	232	232	281	281
	41	126	126	234		282	
48	48	130		235			286
	49	132	132	236	236		291
	50	137	137	240	240		292
52	52	146		241	241	293	293
53		149		242	242	398	398
55		150	150	248		399	399
67	67	166			249	402	402
81	81		170	250	250	404	
90			173	253		405	405
91	91	175		255	255	408	
92	92	176	176	256	256	410	
95	95	200	200	258	258	411	411
	96		203		259	412	
97			211	267	267		415
98	98	212	212	268	268	421	421
	102	213			270	422	422
	112	217		272	272		425
118		219	219	273	273		429
120	120	222		275	275		

4.4.1 Comparison of SCA and McBASC in clustering results

Positions listed in Table 4.1 allow us to compare the identified covariant positions by clustering results of the two methods. Note that positions predicted by two methods substantially overlap. In fact, the overlapping sites represent 62% of the total number of positions predicted by SCA and 65% of the total number of positions predicted by McBASC.

If an additional stringent step is taken by removing the sites with 20 or more gaps (sites 7, 11, 22, 81, 95, 166, 230, 240, 398, 399, 425, and 429), then the results are more consistent with each other. The overlapping sites represent 69% and 68% for the predictions by SCA and McBASC, respectively.

These prediction differences could be due to different scoring methods: SCA uses amino acid frequency and observes changes in the frequency distribution in response to a perturbation at a site by extracting a fraction of total number of sequences containing a relatively conserved residue, whereas McBASC uses a score matrix with consideration of amino acid properties and compares all possible pairs. Thus, theoretically McBASC more effectively uses sequence data and therefore could be a better predictor for genetically covariant positions in a protein family. Nevertheless, the positions predicted by both methods are the most reliable ones with respect to high coupling. Positions predicted by only one method still should be considered to be coupled but with slightly lower coupling strength.

To visualize this genetically interconnected network, we mapped the positions predicted by both methods (after removing the sites with 20 or more gaps) in the 3D

structure of the Torpedo nicotinic receptor α subunit (Figure 4.6A). Each position in the high coupling cluster is shown with its side-chain and color coded as: red represents the positions predicted only by McBASC; green represents positions predicted only by SCA; yellow represents positions predicted by both methods. Note that yellow residues form a sparse network with high density in binding domains, coupling region, and gating machinery. The red and green residues further fill gaps to link yellow clusters together, forming an interconnected network.

Two salient features are apparent in this interconnected network. First, those positions are highly clustered in functionally important domains connecting agonist binding site through the coupling region to the gating machinery, forming a putative activation pathway. Second, many positions are concentrated in a region of recent focus for the sites of action of many allosteric modulators [77] including transmembrane domains. These two aspects are further discussed in detail below.

4.4.2 Activation pathway

The location of the highly coupled residues strongly suggests their potential importance in channel function. The agonist binding pocket of a receptor is located in the amino-terminal domain at subunit interfaces between two subunits, each contributing three binding loops [15]. Figure 4.6B and Figure 4.6C plot highly coupled residues in the context of other residues in two different views (principal face and complementary face). The residues in high coupling cluster for all three colors in Figure 4.6A are now in yellow. Important binding sites are highlighted with red in principal face and cyan in complementary face. For the convenience of numbering,

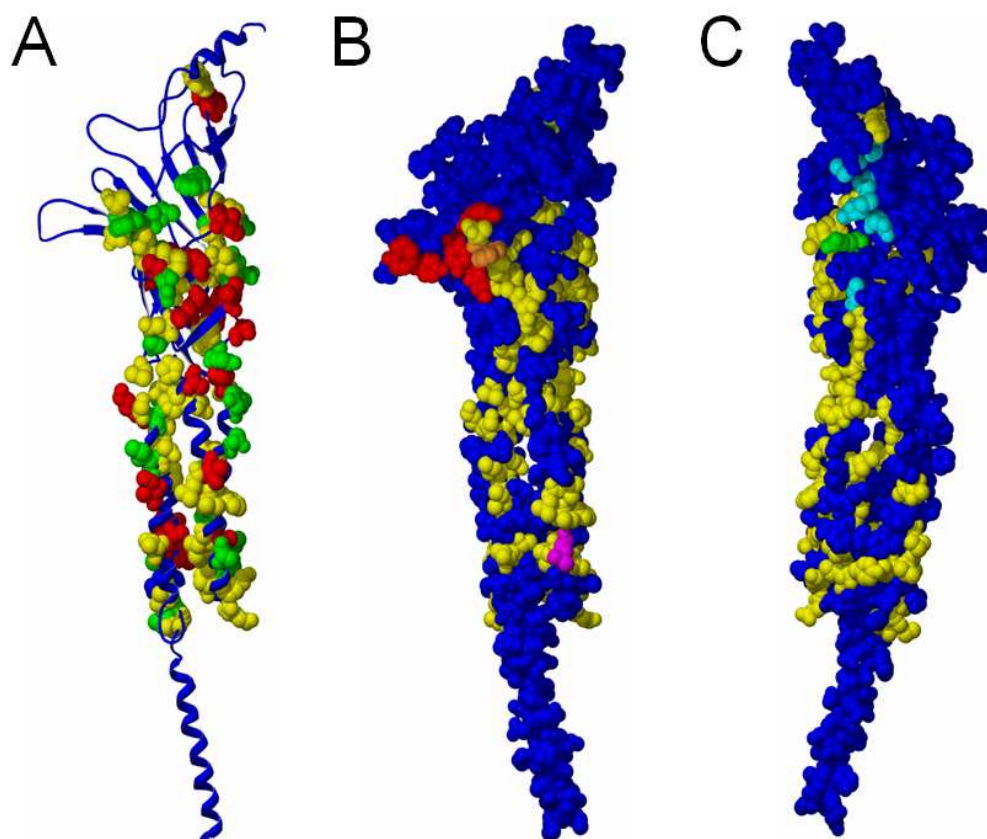


Figure 4.6: Mapping Highly Coupled Sites onto 3D Structure

A. Covariant sites predicted by SCA only (green), by McBASC only (red), or by both methods (yellow). B. Principal face of binding pocket (red) and other regions with all predicted coupling sites (yellow) and gate forming M2 leucine (purple). The orange residue represents overlap between red and yellow. C. Complementary face of binding pocket (cyan) and other regions with all predicted coupling sites (yellow). The green residue represents the overlap between cyan and yellow.

both faces are shown in the same subunit.

Only for homomeric channels does the same subunit contribute both principal face and complementary face of the binding site. In heteromeric channels, principal face and complementary face of the binding pocket are in different subunits. The overlapping residues are in orange for the overlapping between red and yellow or green for overlapping between cyan and yellow. The gate forming residues are highlighted in purple.

Note that only two highly coupled positions (sites 55 and 149) overlap with binding site residues. This is because most functionally important residues are highly conserved and non-variant, and thus escape detection by covariant analysis. However, these binding site residues are flanked by highly coupled residues in both principal and complementary faces (Figure 4.6B and Figure 4.6C). With the exception that predicted positions flank binding residues in loop E from the top of the molecule (Figure 4.6B, sites 67 and 112), all the other positions form an interconnected network connecting binding pocket to the gating machinery. Interestingly, in amino-terminal domain, the highly coupled residues are distributed only in the inner sheet. This is consistent with current understanding of the activation mechanism as suggested by 4Å electron microscopic study: activation involves a clockwise rotation of the inner sheet of the amino terminal domain around its own axis in each subunit [74].

The highly coupled positions also form clusters in the contact region between amino-terminal domain and transmembrane domain. This is a region believed to be crucial in coupling binding to channel gating. In fact, the coupling between amino-terminal domain and channel domain has been postulated to be mediated by the M2-

M3 linker [77, 28, 69]. More recent studies with EM structure of nicotinic receptor [99] or mutagenesis studies further suggest that the coupling is mediated by interactions between N-terminal domain loop2/loop 7 and transmembrane domain linker M2-M3, although crucial residues involved in coupling vary with different receptors. The rate-equilibrium free energy relationship analysis also suggests that both loop 2 and loop 7 (cysteine loop) are involved in channel activation [17]. Loop 9 (Loop F) is also required for the function of a chimera channel [14].

Although key residues for coupling are not predicted by our results, the residues required for benzodiazepine allosteric coupling in M2 and M2-M3 linker (GABAR γ 2T281, I282, S291 [14], GABAR α 1V279 [12] do overlap with the highly coupled positions. In addition, our results provides a possible link between binding domain and loop2 and loop 7 in this putative allosteric network (Figure 4.6A), which may represent the physical basis for inner sheet movement as a “rigid body” during channel activation [74]. As a final note, highly coupled residues are also clustered in the middle and intracellular end of M2 domain, a region with the putative ion channel gate (Figure 4.6C, cyan residues in the M2 region) as suggested by many studies [55, 20] and ultimately confirmed by EM studies of nicotinic receptor at 4Å resolution [99]. Again, the conserved M2 leucine is not predicted by covariant analysis but is surrounded by highly coupled residues.

The highly coupled positions, however, cover the position in the beginning of M2 in the intracellular end, the location of the selectivity filter, which differentiates cationic nicotinic and serotonin receptors from anionic GABA- and glycine receptors [20, 44]. In addition, other residues in the M2 domain [79] and other transmembrane domains

such as pre-M1 [73], M1 [30], M3 [65], and M4 [29] are also important in channel gating, and the M1-M2 and M2-M3 linkers have been suggested to act as hinges governing allosteric control of the M2 domain [74]. Given significance of all four transmembrane domains in channel gating, it is understandable that the highly coupled cluster covers these transmembrane domains.

In summary, an interconnected network is identified that physically links agonist binding domains to channel gating domain. This would represent an entire allosteric network, through which binding signals in the amino-terminal domain can be transduced to gating function in the distant location.

4.4.3 Sites of action for allosteric modulators

In addition to the gate-containing M2, our results showed that highly covariant clusters also include positions in M1, M3, and M4 domains. All four transmembrane domains, especially extracellular half of M2 and M3, are recently recognized as important sites of action for many allosteric modulators such as alcohol, general anesthetics, neurosteroids, and barbiturates [35]. Allosteric modulators for ligand-gated ion channels are compounds binding to a site distinct from the agonist binding site. With the exception of benzodiazepines, which, like agonists, bind to the amino-terminal domain but in a different subunit interface, most allosteric modulators exert their action by binding to the transmembrane domains of the receptor. In fact, the sites of action for some allosteric modulators, such as the site for barbiturate/neurosteroid/etomidate/propofol modulation (GABAR β 2G219 in M1 [18]), sites for alcohol (GlyR α 1S267 [72]), or neurosteroid (GABAR ρ 1I307 [76]) in M2, sites

for alcohol/anesthetics (GlyR α 1A288 [18]), barbiturates (GABAR ρ 1W328 [78]), and redox (GABAR β 3C313 [5]) in M3, sites for alcohol binding (AChR α H408, C412 [78]) in M4 are overlapping with or flanking to the highly coupled positions.

By binding to the transmembrane domains, these allosteric modulators can alter the energy landscape to favor channel opening and potentiate neurotransmitter action. Many of them, such as barbiturates, neurosteroids, and general anesthetics, can even open selective channels directly. Again, functionally important residues in channel gating and modulation in this region can be highly conserved and escape detection by covariant analysis. Nevertheless, given that many sites are overlapping with or flanking the experimentally identified sites for the action of many allosteric modulators, we have sufficient reason to believe that the predicted interconnected allosteric network should also serve as the framework to mediate allosteric modulation.

4.5 Conclusion and Future Work

The chapter's goal is an application system with clustering playing a key role, combined with statistics design, for analysis of a putative interconnected network of a protein family, Ligand-gated Ion Channels (LGICs). By applying statistical analysis with clustering analysis, we found that multiple positions in a cluster are mutually coupled, which reemphasizes an important concept: binding and channel function are mutually coupled [20]. This long range coupling requires an interconnected allosteric network [21].

The results suggest potential use for other allosteric proteins to aid in the study of coupling of proteins. By applying this system to LGIC, we hope for inspiration on

fertile modification so that it can do more in the process of protein structure analysis and prediction. However, caution should be exercised when applying the result to a particular member of the cys-loop family at precise positions. First, since the nature of interconnection with coordinated mutations, the effect of single point mutation at any particular site on channel function may vary with different receptors. Second, this analysis could be limited to the coupling between residues within one subunit. It may not account for the interaction between subunits. Some new methods need to be developed to solve this problem.

The system seen in Figure 4.1 can be used to project potential future work, whether it be in at the starting point of multiple sequence alignment or at any of the subsequent steps.

- Improve the initial calculational phase, i.e., modify the “Multiple Sequences Alignment” algorithm, as suggested with MABAC in chapter 2.
- Modify the “Coupling Analysis” method, taking into account work with association rules and entropy analysis, reputed to have been applied successfully in earlier work.
- Apply visualization techniques within our MABAC clustering method so that it can be applied within systems of this type.
- Modify the coupling and/or covariance method so that inter-subunit coupling is counted as mentioned in Section 4.4.2.

CHAPTER 5

IMAR: IDENTIFYING MOVEMENT FROM ACCELEROMETER RECORDINGS OF REHABILITATION PATIENTS

Research on stroke, a leading producer of disability and handicap, provides opportunities for applying and further developing computing systems to address a multiplicity of dimensions: prevention, diagnosis, and rehabilitation. IMAR (Identifying Movement from Accelerometer Recordings) is a tool for facilitating management and analysis of real-world rehabilitation research data that employs data repository and mining techniques. IMAR's programmed features addresses data collection, organization and management on the input side, information mining on the output side, and a data repository in-between. Input processing automates data cleaning, transformation and organization. Output exploitations range over graphical display and visualization, statistical manipulations, conventional and AI-based modeling. Prime goals of IMAR include 1) classifying patients movement as "functional/ nonfunctional," based on available data to avoid the present situation of costly video scoring of movements; 2) classifying movement related to training as "training" or "not" Some preliminary steps have been taken toward making the repository of system open to integration in other rehabilitation phases and to accommodating data from different sources.

5.1 Introduction

Disability and handicap are worldwide medical and social problems and stroke is a leading contributor to the situation in the USA. Challenges abound because the com-

combination of physical, cognitive and communication impairments stroke entails leaves many patients with long-lasting disability. Research accordingly is being addressed to all facets of stroke: prevention, diagnosis, surgical and non surgical treatment, and rehabilitation [92]. The work described here deals with a computer system that helps rehabilitation research.

In the rehabilitation field assessing “functional activity in the life situation” [40, 94, 101] implies an underlying task of “identifying movement” (IM) in laboratory and other settings, e.g., in home. The latter presents needs for novel departures in the experimental field and computer techniques, which can be used to assist in the automating of approaches.

In the general setting, IM constitutes a complex range of research probes and competing approaches [2, 91] covering movement measurement devices and procedures, theoretical frameworks, and computer use. In this chapter, IM involves all these features, with goals of producing a data repository and extracting information from it, vis-a-vis data under labels such as “functional (with opposite of non-functional)” and “training (vs. non-training)” movement provided in Section 5.3.

It has been postulated that “functional activity in the life situation” is the most important outcome to pursue and measure [96]. The development of new techniques that effectively transfer results obtained in the clinic setting to real-life setting requires innovations in measurement technique. Accelerometers have been used for measuring body movement (arm, leg, chest etc.) involved in activities of daily living (ADL) [102, 95]. However they have not proved sufficient to meet all needs of researchers. In particular, accelerometer measurements in rehabilitation study usually are coupled

with videos of the patients' activities. Information obtained from the videotapes is achieved by having a physical therapist laboriously watch and code the tapes for functionality and other measures of interest. Being able to extract desired information directly from accelerometers has thus become a goal and introduces computation into a prominent role.

Continuous measure of "functional activity in the life situation" call for analyzing large volumes of data and merits computing explorations. Computer methodologies have been applied to cognitive and biological study in similar situations [84, 85], and artificial neural networks have been used to recognize types of overall physical activity from accelerometers [59]. No attempt has been made to identify patterns in accelerometer data that distinguish functional from non-functional arm movement.

IMAR, a combination of data repository and data mining system, stores and extracts information about functional movement from accelerometer data. It also address the training and non-training movement. IMAR represents a programmed environment in which required information is extracted via a diverse set of data mining tools, e.g., clustering, neural nets, support vector machines, decision trees, and general simulation-visualization techniques. Examples of these are presented in this chapter.

5.2 Overview

Past data collection efforts in the settings that provide the data for this chapter have been isolated. These data are available in an intranet and the long-range view sees them on the internet as part of a data warehouse, a web-based or collaborative

grid system. Collaborations with other research centers that attack stroke rehabilitation methodologies are implied in the internet setting. A broad systems view required them for long range planning and diverse processing schemes of a data mining nature must fit into the view.

Other complex systems views referenced are relative to future work. These include simulation environments, knowledge bases, decision support systems. These views focus on feedback. As seen later (section 5.4), some results prove to be of limited usefulness and require further processing. These cases represent a simple form of feedback. More sophisticated feedback can be envisioned, some involving potentially complex artificial intelligence. Going into these in detail is not in the present scope; they are, however, kept up front to help define needs for follow-up.

Figure 5.1 portrays an overall comprehensive system view presented at top level. It starts from raw data, performs preliminary processing, stores information in a state-of-the-art database, and proceeds from there to retrieval action. This information retrieval includes several types of statistical processing, and advanced data mining techniques including soft computing models, e.g., neural nets, and clustering.

Data in the repository are subjected to several forms of processing, both group and individual (patient) data are addressed. The data and its processing constitutes an integrated distributed heterogeneous repository wherein models on stored data portray unified and coordinated attacks on data besides data repositories and warehouses.

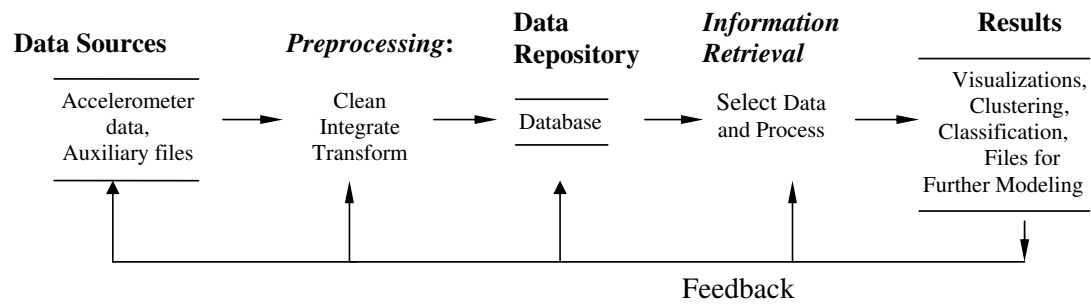


Figure 5.1: Conceptual Model

IMAR activity begins with raw data and terminates with a variety of results of interest to its users. The database is a focal point in that it represents a semi-permanent stage between raw data collection and processing results. Selected data is extracted from database for further processing. Results again feed back the Preprocessing and Information

5.2.1 Getting data into the repository

Getting experimental information into the data repository requires care. Input data has multiple files and may experience “corruption” which may lead to discarding input data or storing it into subsidiary files for further review prior to incorporation into the repository. Openings also should be envisioned for future possibilities, e.g., for on-line, interactive data collection directly from accelerometers to a database after pre-processing.

Current processing focuses are on certain problems that occur in data collection, e.g., with the equipment (the accelerometers) as well as difficulties in other aspects of data collection, particularly when human judgment of videos is involved.

Section 5.3 defines considerations necessitated by data collection and entry of data into the data repository with particular reference to accelerometer data collection and associated preliminary data events. System aids for computerized entry of information are discussed. This section also points to a few features about how subsequent

processing (after data repository entry) may affect treatment of data. Further remarks are made on what the processing options entail in data collection and overall organization of the repository; a few remarks appear in Section 5.4 relating to model results feedback and how it might be stored in the repository.

5.2.2 Extracting information from the repository

Figure 5.1 depicts a framework which organizes several activities which have been carried out previously in an ad-hoc manner, independent from any overall systematic attack. The processing examples presented later, as representatives of second half of Figure 5.1, provide insight into how a broad scope of research work naturally follows from the data accumulating phases. The work presented in Section 5.4 on data analysis and processing is highly suggestive of how modeling and simulation may play significant roles in the future systems.

The intention of the discourse is to establish a base for rich exploitation of the repository data. The initial need provoking the data repository may have been less ambitious. It may have stemmed merely from a desire to handle a large and growing body of experimental data. However, the stored data in the repository provides a great deal of data mining opportunities.

5.2.3 System evolution

Most computing systems in the stroke field are dedicated to subsystem considerations [34, 96, 2]. Sometimes they are restricted to a particular point of view about what data to collect and what basic processing to perform. At the opposite extreme,

systems addressing all facets of stroke appears infeasible at this time. Therefore, it is reasonable to develop systems with goals broader than single or narrow systems, which are open to integration with new subsystems and their implied components as well as to batteries of processing. The role of feedback (results, documentation, limitations and suggestion for further work) needs to be addressed. This chapter thus addresses several aspects of these issues though automation for most of these constitute future research effort.

The central data repositories are designed to be open in another important way, which can be characterized as open to collaborative processing within a grid perspective. The present system is an “intranet” system though a step up to the internet is expected. The work described in the intranet context carries over to the internet context, especially, creating a data repository and advanced data mining [50].

5.3 Data Collection

Data from patients during rehabilitation contains different content and forms. IMAR therefore requires integration and plans relative to potentially advanced data collection. This is especially for the future where current data may be joined with data with unknown or hypothesized content forms.

5.3.1 IMAR input

The input parts of IMAR at this time are primarily concerned with recently collected data. Data from patients in both lab and home settings appears in raw forms. These data undergo preliminary processing prior to entering the repository. Repos-

itory data must be organized in such a way that it promotes data mining activities ranging from simple data retrieval to data selections aimed at very specific modeling activities and associated visualizations.

Input data involves physical movement measurements obtained via accelerometers coupled with activity descriptions from video scoring labelled with “functional” and “training-related” movements (to be described in more detail shortly). These “function” and “training” values are obtained from human assessments of patient activities and represent a different stream of input values to be integrated with accelerometer data in the repository.

There are data integrations that impact input processing in a variety of ways. Some data relate to the ability to create groups of data for individual patients. Some data groups may be formed “across patients,” e.g., all patients performing the same or similar task.

Predicting functional and task designations solely from the accelerometer counts is a particular goal mentioned earlier. At a basic level, such predictions provide researchers with a second set of task evaluations with the human video scores. At an ultimate level is the most interesting and challenging notion of eliminating entirely the human component associated with the input. This means accelerometer data alone need be collected and the time-consuming and costly video observation work can be eliminated.

The plan of attack in the immediately following text is to introduce IMAR measurement details and to understand how these impact longer-term data collection and integration. The role of computing systems in this and already implemented

algorithms for part of the ultimate processing are outlined. Accent in this section therefore is on the left half of Figure 5.1, that is, up to storing information in the IMAR repository. Figure 5.2, which is displayed below, expands on Figure 5.1, exhibiting additional features and serving to guide the discussion.

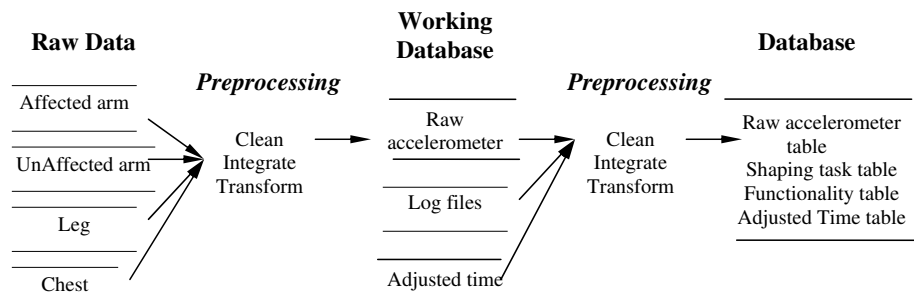


Figure 5.2: IMAR Input Phase

IMAR activity begins with raw data and terminates with a variety of results of interest to its users. The Database is a focal point in that it represents a semi-permanent stage between raw data collection and processing results. The file and processing action are described in some detail in the text.

5.3.2 Accelerometer measurements

Some background comments on accelerometer technology and employment in rehabilitation study are helpful. Accelerometers provide measures of physical movement during activities. They have capabilities that make them very useful in a number of real world situations [58]. Accordingly, they are valuable for analyses that span laboratory and home situations.

Accelerometers are approximate tools but nevertheless meet a variety of needs. For example, Fahrenberg [34] provides an account of how both posture and motion can be assessed using “wide bandwidth piezoresistive accelerometers.” These are not the accelerometers used in this study but they provide useful background information

on accelerometer. Accelerometers provide both direct current (DC) and alternating current (AC) components and each of these contributes relevant information: the DC component is a key to assessment of “slow motion and change in position referring to the gravitational axis” while the AC component calibrated represents “acceleration along the sensitive axis of the device.” Fahrenberg [34] gives an assessment of physical activities with a relatively large number of subjects and conditions: twenty-six student participants and eight conditions. In the experimental situations, this assessment represents a validation of results along these dimensions. The eight conditions studied include sitting, standing, lying supine, sitting and typing on a PC keyboard, walking, climbing stairs, walking downstairs, and cycling. The same set of conditions was repeated in reversed order and classification of physical activities according to the eight conditions (first trial) was based on four parameters: DC components, trunk, thigh, and lower leg and AC component trunk. Applied to the second trial, classification was correct in almost all patterns.

Keil and Taub [58] validated an application of accelerometers in rehabilitation studies. In order to determine the value of accelerometers as a measure of real world outcomes when, in particular, a subject is outside the laboratory, accelerometer recordings from the wrist were compared with simultaneous electromyogram (EMG) recordings from the lower and upper arm. There were considerably high correlations for standardized tasks as well as activities of daily life (ADL).

5.3.3 Accelerometers in the IMAR context

Accelerometers yield “counts” and a set of counts is associated with each accelerometer. In IMAR studies, two to four accelerometers have been in play during a data recording session, and accordingly 2-4 streams of recordings have emerged from them. Figure 5.3 depicts a “patient” with four accelerometers attached to his body (arms, leg, and chest) to capture body movements of the torso and three extremities. In some cases, only one accelerometer was worn on each arm. For the functionality study, the complete set of 4 accelerometers was worn when 4 units were available. On some occasions, when 4 units were not available, the chest unit was omitted. The devices can be used in both laboratory settings and everyday life situations [102, 34]. Depicted in Figure 5.3, along with the patient, is a view of an accelerometer and a case. Accelerometers are about the size and weight of a large wristwatch.

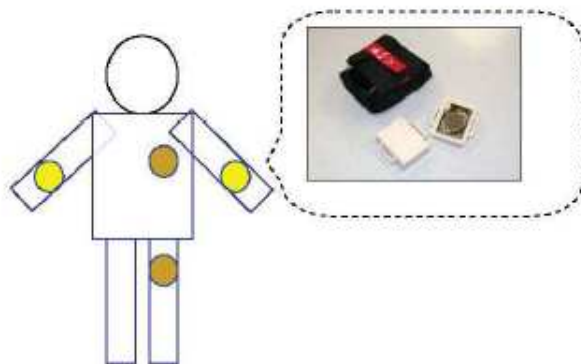


Figure 5.3: Experiment Set-up

A “subject” is depicted with four accelerometers attached on extremities and on the chest; some experiments use arm devices only. Recorded patient movement is the first stage of activity portrayed in Figure 5.1 At the right shows an accelerometer with case and cover.

In typical IMAR cases, accelerometer data may be quite extensive, e.g., it may

contain recordings at two-second intervals over a three-day period, representing more than 100,000 data points per accelerometer. Session accelerometer counts are stored within an accelerometer's own internal memory. Transmitting the data to its database currently is in a batch mode but entry is automated partially, which will be described shortly. It may be worth noting automation of delivery over a wireless system may be an option in the future; this could have an impact on processing styles.

With multiple accelerometers, each recording may start at a different time point (e.g., a human experimenter turns one on after another in succession). Thus, getting recordings into databases and organizing them into useful structures requires a number of basic adjustments. The present method is to manually set adjusted time by technicians but it will be automated in future.

“Identifying Movement classes,” a main component of IMAR, introduced in the Introduction and an important focus in Section 5.4), may be achieved in different ways, e.g., self reports, direct observations, videotaped activity periods that are scored later. Most of these involve human evaluation, e.g., the scoring of the videotapes by specifically trained physical therapists. Though results achieved in the latter fashion often are capable of passing reliability test and are repeatable, they tend to be both time consuming and costly.

5.3.4 *Shaping tasks*

During rehabilitation experiments, a patient is asked to do several “shaping tasks;” these tasks include moving objects about a table or relocating small objects from one container to another. Shaping tasks are used to improve patient recovery and to

measure patient movement in certain experiments. For each shaping task, a patient needs to carry out several (about 10) trials. In the clinic, a patient works for a limited amount of time (1 or 2 hours). In the home setting, he/she does similar exercises without therapist supervision. The patient is also asked to write a log identifying the start time and end time of each trial in the clinic lab. Patient movements are also recorded by a video camera. A patient tasks period is recorded to provide for evaluation of the relationship between tasking and recovery. The standard task period and inter-task period is defined by patient log file and video tape.

One goal of IMAR is to classify the accelerometer data into task period and non-task period without the patient log file or video tape. Another goal is to classify movements into functional and non-functional and is elaborated on next session.

5.3.5 *Functional movements*

Functionality of Arm Activity Rating (FAAR) Scale is coded into 4 intervals according to Gottman [9]:

- 0) no movement or activity;
- 1) nonfunctional activity- a movement or action that does not have any function (e.g., tic, tremor) or is largely secondary to movement of other parts of the body (e.g., arm swing when walking, passive movement);
- 2) nontask-related functional activity- a movement or action that some function although it does not accomplish a task;
- 3) task-related functional activity- a movement or action that helps to accomplish a task (e.g., grasping a can, wiping a table top).

Arm activity was not rated if a sufficient area of the arm was not visible; such epochs were coded as unobservable and treated as missing data points in analyses.

A 2-step scale was formed using the 4-step ratings by collapsing categories 0 and 1 to form a single super-category of nonfunctional activities and categories 2 and 3 to form a super-category of functional activities. This transformation was used because, prior to rating the videotape segments in the study, it seemed that the distinctions between functional and nonfunctional activities were rather sharp, whereas the distinctions within these super-categories were somewhat ambiguous (especially between non-task- and task-related functional activity). After observers received 12 hours of instruction and 24 hours of practice over 90% agreement on the 2-step scale was obtained and they were deemed ready to rate the videotape segments from real-world data in the study. In the data processing part of this study, a 2-step scale was used in the major studies [103].

5.3.6 Data sources

Three types of data sources need to be handled in IMAR: (accelerometer) raw data, log file and adjusted time; we elaborate these in the next few paragraphs. Accelerometers are built in a way that their stored contents can be loaded to a computer via a USB port. The raw accelerometer data contains a header followed by some data counts. An example of accelerometer data is shown in the Figure 5.4

The header contains start time, start date, download time, download date, SN (Serial Number) id , and epoch period. The trial information (start time and date, epoch period) can be found from the header. The rest of the data contains ac-

```

-----
SN:12583 Ver 2.2
Start Time 10:30:00
Start Date 08-21-2003
Epoch Period (hh:mm:ss) 00:00:02
Download Time 13:40:33
Download Date 08-26-2003
Current Memory Address: %131072
Battery Life Remaining: 1957 hrs  MODE= 2
-----
Epoch #      Counts
    1         -1
    2          0
    3          0
    .         .
    .         .
    .         .
   357        222
   358       1357
   359        499
    .         .
    .         .
    .         .
  131071         0
  131072         0

```

Figure 5.4: Accelerometer Data Example

Accelerometer data from a single device contains epoch-by-epoch (e.g., every 2 sec) movement (counts) over a designated observation period ranging up to almost 3 days. Header information includes a serial number (SN) and date-time information used in Preliminary Processing (Figure 5.1), e.g., in the cleaning and integration operations. (See the text for details, including spurious negative counts.)

celerometer counts on two-second basis typically for three days. Sometimes in a data collection episode, accelerometer raw data might contain more than one header because of a loading error. A warning is needed when this happens and this data set need to be checked by the therapist. The start time of each accelerometer for the same patient in an experiment is also checked for consistency. This is obvious since four accelerometers should start at the same time. This issue will be addressed in next section.

There are two types of log data. Patients need to write down the start and end time of each trial in each task so that the therapist can keep track of the training period versus non-training period. This type of log file is defined as *Shaping Task*. The therapist needs to rate the functionality from the video that records patients movement to get a training file for the neural net classification. This type of log file is defined as *Functionality*. In IMAR, functional movement is classified into two or four categories according to the criterion defined previously 5.3.5.

Adjusted time exists because sometimes the start time of the accelerometer is not always accurate. One of the reasons is that when the accelerometer is initiated, there might be a slight time delay that causes discrepancies. Another reason is that the timer of an accelerometer is not synchronized with the clock used by the patient. It is also possible that patients make errors when writing a log file. In all these cases, human intervention is called for and the analyst must adjust the time so that the timing of different accelerometer recordings is synchronized.

5.3.7 *Data preprocessing and storage*

In a data repository and data mining system, input data may be preprocessed to help improve quality. There are several methods used in data preprocessing: data cleaning, data integration and transformation. Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. As mentioned in Section 5.3.6, there are three distinct data sources. Data integration merges data from these sources. Data may also need to be transformed into forms appropriate for mining.

The raw accelerometer recording data is very large (recall that each accelerometer typically contains more than 100,000 counts). It increases the burden of database if there are data from large numbers of accelerometers. However, since most of the time, for example, when patient is sleeping, accelerometer counts are zeros, repository storage can be lowered if data is stored only when any one of four accelerometer counts is not 0. By using this method, more than 90% disk space is saved relative to a naive “store all” policy.

The data cleaning/preprocessing part runs at the backend of IMAR, while a user executes preprocessing with a Graphical User Interface (GUI). A friendly user interface assists in understanding the data since the users of this system are often therapists without computer programming experience.

IMAR provides graphical user interfaces for loading data, querying different portions of the data, presenting query results, and echoing error messages. All of them are designed and modified according to user requirements.

PediLoadRawData

SID: 614 Affected: 04F12S2.A09 SN:11509

tdID: 01 UnAffected: 04F12S2.C28 SN:12028

Start Date/Time: 2000-04-28 14:45:00.0 leg: 04F12S2.L30 SN:12030

End Date/Time: 2000-05-02 17:58:48.0 chest: 04F12S2.R31 SN:12031

The raw data you import is the following

Time	SID	TDID	Affected	UnAffect...	Leg	Chest
2000-04-28 14:45:00.0	614	01	-1	-1	-1	-1
2000-04-28 14:52:26.0	614	01	0	0	1	0
2000-04-28 14:52:28.0	614	01	0	0	5	6
2000-04-28 14:52:30.0	614	01	0	0	10	15
2000-04-28 14:52:32.0	614	01	0	0	0	1
2000-04-28 14:52:34.0	614	01	0	0	5	2
2000-04-28 14:52:36.0	614	01	1	0	29	44
2000-04-28 14:52:38.0	614	01	0	0	4	9
2000-04-28 14:52:40.0	614	01	1	0	4	14
2000-04-28 14:52:42.0	614	01	4	0	1	14
2000-04-28 14:52:44.0	614	01	5	3	11	2
2000-04-28 14:52:46.0	614	01	3	3	15	45
2000-04-28 14:52:48.0	614	01	0	0	16	7
2000-04-28 14:52:50.0	614	01	0	3	3	1

connectDB load to kidsData cancel

Figure 5.5: Graphical User Interface for Loading Data: Interface used for loading accelerometer raw data. User specifies SID, tdID and the data is then integrated in the data base.

Figure 5.5 shows a screen appearing during loading accelerometer data. Before loading any data, a user needs to press “connectDB” (green) button at the left bottom corner to check if the database connection is correct. If the connection is successful, the user can see all table names listed in the right-middle bottom box, which the user needs to choose later for loading the data. The user then defines the subject id and data set id in either of two ways: choosing one from the list if it is in the database, or typing in the corresponding text area.

There are four buttons corresponding to four accelerometers: affected arm, unaffected arm, leg and chest. As explained in Section 5.3.3, accelerometers for the affected arm and the unaffected arm are needed for every experiment. Therefore, two files from arms are required for each loading. Leg data and chest data are optional since they are only required in some experiments.

Time discrepancies were among these 2-4 accelerometer files (affected arm, unaffected arm, leg, chest), and other error sources were mentioned above where their causes were outlined. IMAR detects discrepancies and provides a warning message. If there is nothing wrong with the input files, the start time and end time of affected arm is printed in the corresponding text field. Users may load part of raw data by defining the start time and/or end time. After all of these are chosen, the user loads the data by simply clicking on “load” button and a table of integrated data comes out as shown in the middle part of Figure 5.5 (for confirmation).

5.3.8 Data storage

After raw data is loaded, the data is stored in repository for further processing, which is discussed in Section 5.4. In the repository (database), each type of data source identified in Figure 5.1 is stored in a table.

There are three types of tables in the database corresponding to the three types of source data:

- table *rawData* which contains information from accelerometer recordings,
- table *shpgTask* which contains start time and end time of each shaping task trial by patient,
- table *adjustTime* which contains information of adjusted time defined by data analyst.

Each table is described in terms of a schema. The schema of *rawData* is (*subject ID*, *tdID*, *start Time*, *affected arm*, *unaffected arm*, *leg*, *chest*). This table stores the 2-4 accelerometer entries for every two seconds. The patient ID is denoted as *subjectID* and the experiment ID is supplied as *tdID*. The main goal of this table is to integrate raw data from 2-4 accelerometers in each patient as shown in Figure 5.2. That is to say a group contains data cleaned and transformed from 2-4 accelerometers.

The schema of *shpgTask* is: (*subjectID*, *tdID*, *shpgID*, *trialID*, *start time*, *end time*). This table contains the start and end time of each patient in rehabilitation experiment trials. Each trial lasts from several seconds up to about two minutes. The start time and end time of each trial are recorded by patient. The original log file is transformed so that several IDs are marked. The transformed log file is stored

into the designated table so that the accelerometer recordings in training and non-training periods can be traced. The data in the *rawData* table is then retrieved by the timestamp in the *shpgTask* table later on for further processing.

The schema of *adjustTime* is (*subjectID*, *tdID*, *affected arm*, *unaffected arm*, *leg*, *chest*). Since accelerometers might be initiated incorrectly, the start time and end time of *shpgTask* could be shifted thereafter. This happens more often in home settings. The required adjusted value is input by the data analyst. The system updates the results on time so that the user can track the corrected information. In the future, it is hoped that this *adjustTime* table will be automated by the system.

Data input background is presented in this section. A first step cleans source data, transforms it as described in next section, and imports the results into a Microsoft sql server database. After the data is loaded into a repository database, the next step, as depicted in the Figure 5.1, is to retrieve data from the database, e.g., for simple inquiries to data analysis methods, such as statistics and modeling.

5.4 Mining the Repository

After data is secured in the repository it is then ready for further processing with the purpose of generating results that define important facets of the data. The repository is rich with data (hidden information) for data mining.

Several methods often identified with data mining techniques in this phase of the system include simulation, clustering and classification sometimes coupled with visualizations. In principle, processing results success stories and failures become available as feedback to earlier stages. In cases, results reach all the way back to the

data collection phase, and, do so when succeeding experiments are undertaken. Results feedback to designing new processing activities, in terms of our guiding diagrams (Figure 5.1), is the tightest loop. Examples of this tight loop include initial models, e.g., support vector machines and early ANNs produced less successful results the analysis of which led to improved models. A similar situation occurred in clustering work.

5.4.1 Data extraction and visualization

Any period of data is retrievable from the system and the stored data in the database can be extracted as the user requires. Figure 5.6 provides an example of User Interface for extracting data from the repository. The user has different options of picking data with subject ID “*patientID*” and data set ID “*tdID*”.

sTime	sID	tdID	clinicOrHome	affected	unaffected		
2003-06-16 16:00:00.0	614	01	01	-1	-1	0	0
2003-06-16 16:00:02.0	614	01	01	0	0	0	0
2003-06-16 16:00:04.0	614	01	01	0	0	0	0
2003-06-16 16:00:06.0	614	01	01	0	0	0	0
2003-06-16 16:00:08.0	614	01	01	0	0	0	0
2003-06-16 16:00:10.0	614	01	01	0	0	0	0
2003-06-16 16:00:12.0	614	01	01	0	0	0	0
2003-06-16 16:00:14.0	614	01	01	0	0	0	0
2003-06-16 16:00:16.0	614	01	01	0	0	0	0
2003-06-16 16:00:18.0	614	01	01	0	0	0	0
2003-06-16 16:00:20.0	614	01	01	0	0	0	0
2003-06-16 16:00:22.0	614	01	01	0	0	0	0
2003-06-16 16:00:24.0	614	01	01	0	0	0	0
2003-06-16 16:00:26.0	614	01	01	0	0	0	0

Figure 5.6: Graphical User Interface for Querying Data

There are several query methods in IMAR. For example, a user may need the tasking period accelerometer data to get some statistical properties of the data in a specific group including *sum*, *mean*, *variance*, *standard deviation*, etc. Similarly, accelerometer data of inter-tasking periods, functional or non-functional periods, may be extracted. The user may also wish to view accelerometer data from a specific time period to focus on properties of the period.

As mentioned earlier (Section 5.3), when the accelerometer raw data is loaded into the repository, all-zero inputs are deleted to save the disk space. But when the user extracts the data, these all-zero inputs are restored to retain the time stamp information. Therefore, the user does not lose any information.

All query results shown in the text area can be saved as an Excel file for visualization and further processing. IMAR itself provides a basic visualization, i.e., produces histograms of different accelerometer counts along with functionality/training data. Figure 5.7 shows an example.

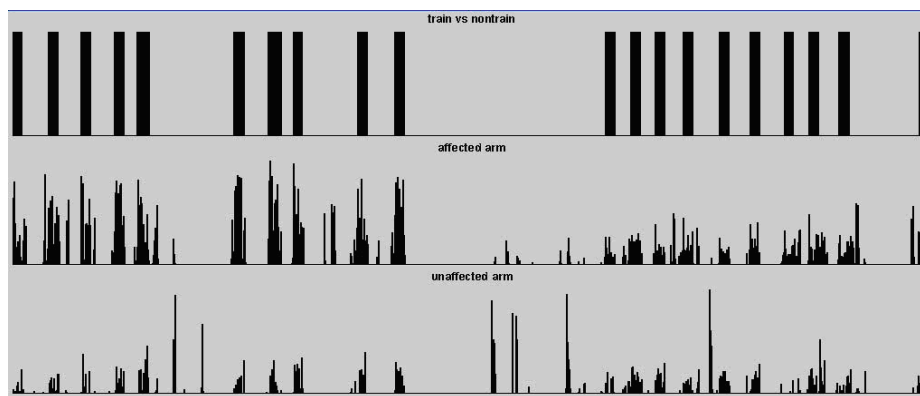


Figure 5.7: Visualization of IMAR Example

The top row demonstrates the tasking vs non-tasking periods. Peaks stand for

tasking periods and plains represent non-tasking periods (Functional/non-functional readings can be chosen instead of tasking/non-tasking for similar situation). The middle row shows accelerometer counts of an affected arm at different time correlated with the top row. The bottom row shows accelerometer counts of unaffected arm. In some other cases, accelerometer counts of leg and/or chest are plotted also.

These rows synchronize inputs from different accelerometers. This figure is created by integrating two tables in the repository's main database: *rawData* and *shpgTask*. From the example, the activity of the affected arm appears related to the tasking vs non-tasking period, while the activity of unaffected arm is less predictable. This information helps in design of a possible pattern that can be used to classify the data. For example, from this figure (in the training period), the affected arm intensity lasts longer than the other two. Therefore, it is postulated that a time window be used to optimize model input, as discussed later in classification. This emphasizes the idea of feedback in IMAR as shown in Figure 5.1. Proof Animation [83] and other animation techniques have also been implemented though they are not shown in this dissertation.

5.4.2 Clustering

Clustering analysis can be applied in the search for patterns in extracted data. Clustering is employed because it is a commonly used method in data mining, and can be used on its own and as a preprocessing step for classification.

Among various clustering methods, MABAC, a clustering method created by Chen et. al., is used because it proved effective in some non-intuitive 2-D data cases [26].

MABAC's effectiveness stems from its counting both inter-cluster and inner-cluster properties. Fuzzy C-means, probably the most widespread implementation of fuzzy clustering, is a fuzzified variant of k-means clustering. It is part of MATLAB's Fuzzy Logic Toolbox. Fuzzy C-means has been introduced in chapter 2 and is used again in this chapter.

The source data used for clustering is the nine-patient functionality data used by Uswatte [102] since this data has been error-checked and proven useful in research.

Two issues need to be considered: (1) how to define the input, and (2), how to define the distance measure between two inputs. Definitions relative to these two issues are applied in further classification, i.e., with other tools, mainly Artificial Neural Networks.

As mentioned in Section 5.3.8, raw counts from 2-4 accelerometers are stored in the same table of the repository's main database. The input is stored in a similar manner, but raw counts are transformed. In IMAR clustering and in later classification work, each input is a vector of transformed accelerometer counts. One way of transforming data is to use a threshold as indicated by Uswatte et al. [102]. But here the threshold is modified to a sigmoid function as following.

$$if(value > 10), value = 10 + \log(value - 10); \quad (5.1)$$

Another form of transformation is to use a sliding window, i.e., an input is combined with several consecutive prior inputs. This transformation improves the accuracy of clustering as shown in Table 5.1. The distance measure used here is the Euclidean distance of two transformed input vectors.

5.4.2.1 *Clustering results.* One way of measuring the quality of a clustering solution is cluster purity [93]. Given that there are k clusters of a dataset D and the size of cluster C_j is $|C_j|$, let $|C_j|_{class=i}$ denote the number of items of class i assigned to cluster j . Purity of cluster j relative to class i is calculated by

$$purity(C_j) = \frac{1}{|C_j|} \text{Max}_i(|C_j|_{class=i}) \quad (5.2)$$

The overall purity of a clustering solution is expressed as a weighted sum of individual cluster purities

$$purity = \sum_{j=1}^k \frac{|C_j|}{|D|} purity(C_j) \quad (5.3)$$

In general, the larger the value of purity, the better the solution.

The clustering results of MABAC and Matlab Fuzzy C-means are shown in Table 5.1. They indicate that: (1) the purity of all clusters by MABAC with threshold ranges from 0.626 to 0.982; (2) MABAC clustering performance matches or exceeds Matlab Fuzzy C-means according to purity in most cases (7 of 9); (3) the threshold method increases the purity of clusters in MABAC in most cases (6 of 9); (4) the time series method increases purity of clusters of Matlab Fuzzy C-means.

The above results indicate the effectiveness of clustering techniques in analyzing this data. Also, the two ways of adjusting input (threshold and sliding window) improve the quality of clusters and provide possible optimization of the classification. This overall first probing of the data followed by improved probing is another example of clustering techniques used as feedback in an application. Clustering results promote further processing, i.e., classification of the accelerometer recordings according to functionality or tasking period.

Table 5.1: Clustering Results of IMAR

Subject	MABAC		Fuzzy k-means	
	raw	threshold	threshold	threshold+time
Lee	0.654	0.688	0.653	0.660
333	0.974	0.974	0.934	0.974
335	0.695	0.899	0.815	0.849
330	0.675	0.676	0.626	0.686
332	0.982	0.982	0.982	0.982
341	0.628	0.891	0.865	0.836
343	0.845	0.845	0.825	0.845
344	0.724	0.726	0.711	0.712
345	0.545	0.626	0.527	0.726

The test results of 9 patient data sets in clinic setting: threshold method and sliding window method both improve the clustering results

5.4.3 Classification

The Support Vector Machine (SVM) [13] was introduced in 1992 by Boser, Guyon, Vapnik and has been greatly developed ever since. The first Artificial Neural Network (ANN) was developed by McCulloch and Pitts (1943) [70] and got much attention in early 1980s. Both SVMs and ANNs have been used in IMAR but ANN models better achieve two original IMAR goals: classifying accelerometer data into tasking vs non-tasking periods and functional vs non-functional periods. The move from SVM to ANN is another “tight loop” feedback as this notion is described in Section 5.1.

The ANN model used in IMAR is a two-layer feed-forward back-propagation model [53]. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by users. Theoretically, ANNs with biases, a sigmoid layer, and a linear output layer are capable of approxi-

imating any function with a finite number of discontinuities [67, 51, 27].

Similar to the clustering work, defining the input is very important. An ANN structure used in this study is depicted in Figure 5.8, where inputs are transformed accelerometer data and outputs are 0 or 1, where 1 stands for tasking or functional movement and 0 labels non-tasking or non-functional movement depending on the model's target.. Choosing the number of nodes in hidden layer is another issue, here the choice is 4 according to rules given in [66].

According to previous results by Uswatte et al., the classification depends not only on the present accelerometer recording, but also on several recordings before and after the present timestamp [102]. Movement, as is perception, is a continuous variable. A tactic used here is to apply a sliding window at each timestamp so that the continuity of movement is accounted for. In section 5.4.2, clustering worked better with several consecutive recordings.

It was found that the mean and the ratio of standard deviation to mean (as a percentage) of the affected arm are higher than those of the unaffected arm during tasking period [100]. Therefore, mean and percentage of a particular window (here the window size is 10 seconds) were used combined with the raw data as the input of the classifiers as discussed later.

5.4.4 *Classification results*

ANN classification results have been validated with different data sets in both home and clinic settings and returned solid accuracy. In this section, an example is shown with 56 data sets from 10 patients in clinic setting for classification of tasking

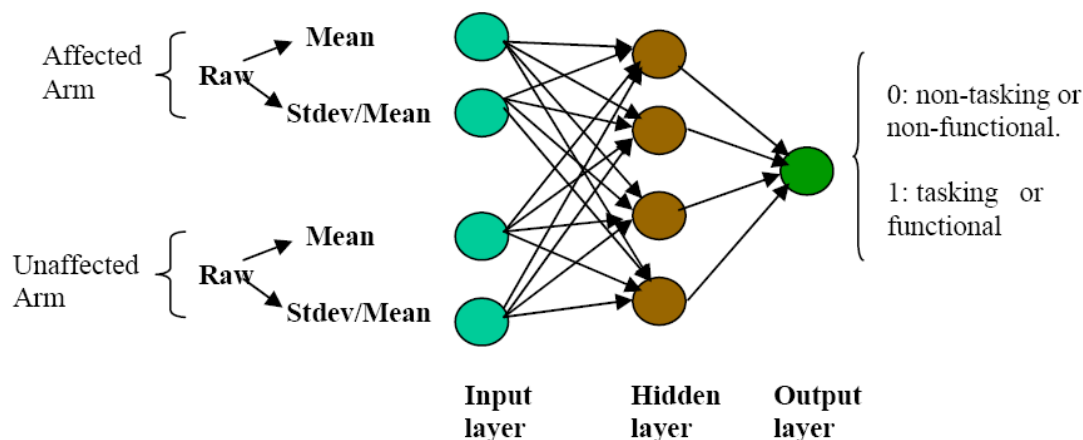


Figure 5.8: Two Layer Neural Network Structure Example in IMAR
 Four inputs are transformed from raw accelerometer data: means and stdev/mean of affected and unaffected arm. Outputs are 1 or 0 correspondingly. More nodes are needed if leg/chest data are taken into accounts.

vs. non-tasking period. Among 56 data sets, 44 are picked randomly as training data, and the remaining 12 as testing data to check the model accuracy depicted in Figure 5.8.

The results are illustrated in Table 5.2. From the table, the average testing rate is around 85%, which is deemed satisfactory for classification purposes. The classification error comes almost equally from false positives or false negatives.

Another way of assessing classification performance is by computing the correlation between ANN-predicted value and real (reported) value. Figure 5.9 shows a correlation plot with the same 56 data sets. In this figure, the real tasking period (reported by the therapists) is highly correlated with the predicted tasking period (by the neural net), i.e., points are distributed around a straight line. Actually, the correlation between subjects is around 0.86, again supporting the effectiveness of the ANN's predictions.

Table 5.2: Neural Net Classification Results of IMAR

Subject	Test rate (%)	False Positive (%)	False Negative (%)	real tasking time (min)	NN tasking time (min)	difference (min)
616	88.2(86.5-90.8)	7.0(5.2-9.7)	3.8(3.3-5.9)	7.3(7.2-7.7)	9.3(7.8-11.2)	-2.1(-3.4-0.5)
621	87.6(86.1-88.6)	6.4(5.1-7.6)	6.3(5.0-6.4)	8.5(8.1-8.9)	9.2(7.3-9.6)	-0.7(-1.0-0.8)
615	84.2(82.6-86.1)	7.9(7.1-13.0)	6.8(4.4-7.9)	9.7(9.7-9.9)	9.9(9.6-14.2)	-0.1(-4.4-0.1)
614	90.5(89.6-91.7)	3.1(2.2-4.1)	6.5(5.1-8.2)	10.6(10.6-10.7)	8.9(7.8-10.3)	1.7(0.4-2.9)
623	87.4(86.1-88.4)	2.8(1.7-3.2)	9.5(8.7-12.2)	12.6(12.1-14.0)	8.4(7.6-9.0)	3.8(3.6-6.2)
622	89.4(87.1-91.8)	2.8(2.0-3.9)	8.3(5.1-10.1)	14.0(14.0-17.6)	13.0(10.6-15.3)	3.4(0.8-3.6)
617	81.7(79.4-86.8)	6.1(3.8-8.2)	12.4(6.8-14.5)	18.2(17.4-19.1)	14.3(12.9-17.0)	3.7(0.4-5.3)
619	82.4(80.1-84.6)	7.0(4.2-8.0)	11.8(7.4-13.8)	20.0(19.2-20.6)	17.2(14.5-20.1)	3.4(-0.4-5.6)
stdev	6.2(4.5-6.9)	3.0(2.5-4.1)	5.4(4.2-5.8)	5.1(5.0-5.2)	4.0(3.8-4.3)	3.4(3.0-3.7)
mean	86.0(85.8-87.2)	5.7(4.1-7.0)	8.3(6.3-9.0)	12.8(12.5-13.0)	11.5(10.2-12.7)	1.4(-0.2-2.6)

The test results of 56 data sets in clinic setting: the average testing rate is around 86%, NN predicted tasking time is similar to the reported tasking time.

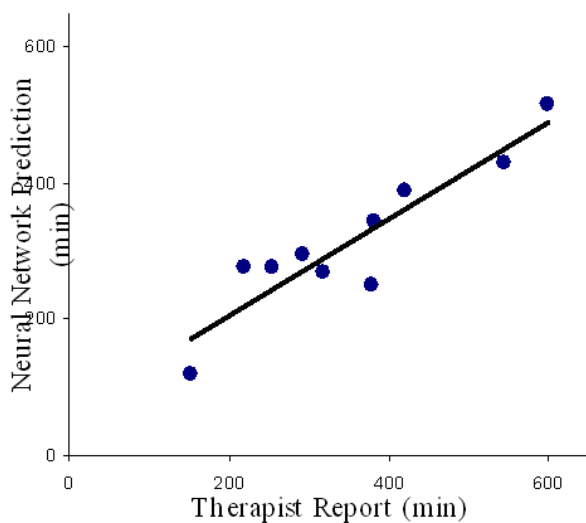


Figure 5.9: Neural Network Prediction vs. Therapist-Report of Average Daily Time in Arm Training for CI therapy Patients in Clinic. Data points are distributed around a straight line, signifying a high correlation.

5.5 Summary and Conclusions

In this chapter, primary concentration is placed on activity of a data mining and repository system. IMAR activity starts from data collection and preprocessing, extends into storage of data within a central data repository and ultimately leads to information retrieval and analysis (with feedback).

From an application point of view, the purpose of IMAR is to identify movement classes from accelerometer data in rehabilitation studies. IMAR's larger system context, a programmed environment, a data repository and mining system, constitutes the context in which all the activities described occurs. Clustering and classification are mainly used as illustrations for the data processing provided by this system. Both clustering and classification results show the IMAR effectiveness in data mining work.

Graphical User Interfaces of IMAR play important roles. Section 5.3 shows an

example how computer aids users go through stages of preprocessing phase. In Section 5.4, an example of retrieval action using the GUI is shown.

The appearance of graphics on the output side often takes the form of visualizations. In a simple case, data from the repository (both accelerometer readings and expert ratings, over time) are displayed in Figure 5.6. In a more sophisticated case, visualization provides features for clustering and classification.

Future plans and possible directions have been presented at various points in the text above. For example, wireless communications may be used in some plans for data collection, e.g., both to provide instructions to the subject and to collect some information on-line. Such communications most likely will lead to new and interesting problems in the pre-processing stage.

On the output side, there seem few limits on ways to mine the data. The most significant future items may well lie in developing and automating portions of the feedback mechanisms. Currently, output results are stored in files that conceptually reside within the repository and modelers can use these files in fashions that suit their immediate needs. Keeping track of common patterns of use can be first step toward, e.g., supplying data automatically with only a small hint as to what kind of need exists.

In the long run, the systems described here are expected to change as a result of analyses which compare them to other comprehensive systems views (as well as from changes due to new research aims). Data warehousing is one concept we've mentioned several times. It finds widespread use, i.e., across business, engineering and scientific domains. It shares features with several systems views where "intelligent systems"

notions are emphasized, e.g, knowledge based systems, simulation and modeling environments and decision support systems. As such, a rich source of information exists to guide progress toward more advanced automated systems for IMAR.

CHAPTER 6

CONCLUSIONS

In this thesis work, several aspects of clustering analysis are presented, with principal applications mainly from biological/biomedical data mining contexts. The dissertation provides discussion on clustering methods themselves, to include our a new clustering algorithm and adaptations of other algorithms to fit into an unified and coherent framework aimed at expediting use and fostering further development. This chapter briefly summarizes the thesis work with focus, first on clustering methods and then on system views. An important feature is that suggestions on future work are also included. Core contributions are also summarized at the end of the document.

6.1 Summary

How clustering works both in stand-alone style and in larger systems are studied in several ways. These include developing our own clustering algorithm in chapter 2, further discussed in “Methods and Techniques” (section 6.2). A next step in this research could be to develop a “testing environment” for such studies, sometimes called a “harness” in software engineering. Having several testing examples are shown in chapter 2.

Beyond the algorithm with its associated testing, most attention is directed to what is labeled “Systems Views” (section 6.3). In chapter 3, a protein sequence clustering system is built which automates the transformations of data and adaption of existing clustering methods. In chapter 4’s “systems study,” clustering is used as

an integral part of a complex sequence of analysis, in an effort to establish a possible model for communication within ion channels. The clustering was employed after the “coupling analysis” or “correlation analysis” for amino acid site co-occurrence was performed. In chapter 5, additional theoretical views of the work are also offered, motivated by research in data mining areas such as artificial intelligence, simulation environments, data warehouse, and decision support systems. This was done in part to stake out possible future directions for specific research on this project.

The data repositories across all these studies, the manner in which they are stocked, formatted and re-formatted for deep processing, is in principle what is done under the designation data warehouse. Data mining, which shares many features and processing setups of data warehousing, can be viewed as embedded within a warehouse. Data imported from the internet is featured in chapters 3 and 4, while local files are called upon in chapter 5, where data is stored in a repository for additional processing.

All told, clustering methods are tested and/or employed in most of the main classes of clustering schemes identified by researchers in the field, as we sought to make certain points, e.g., in our testing activities, or, as we sought to identify a suitable method according to how well it takes care of application needs. Having several algorithms in tow led to opportunities for flexibility in attack on problems.

The work is now reviewed at a more detailed level, starting with “Methods and Techniques” and extending to “System Views.” The chapter and the dissertation is concluded with statements often appearing in dissertation contexts: “The Contributions to the Field and Associated Publications.”

6.2 Clustering - Methods and Techniques

MABAC (MAtrix BAsed Clustering method), described in Chapter 2 is the first “Methods and Techniques” tract. It employs the idea of direct link between two clusters as one part of the computation, the other part being an indirect link through common neighbors.

The name, MABAC, indicates a view of the storage structure on which MABAC works, i.e., a matrix (of similarities). The actual details of computation include a slight variation on matrix multiplication, and, the matrix base of the method merits attention because of it.

So far, MABAC has been tested mainly on 2-D data sets, but the collection of tests includes some difficult cases such as a ring-and-core: a set of data arrayed as a ring surrounding an almost circular core with space between the two, and another example that may be viewed as “hamburger and hot dog” .

Comparisons have been made with other methods, including OPTICS, Matlab Fuzzy C-means, and CHAMELEON. None of the methods tried matches MABAC across all the cases, but there exists some possible explorations that are recommended for additional study. For example, Fuzzy clustering with an appropriate number of clusters, separates out the core in the ring-and-core example, and the fuzzy memberships suggest a possible means of connecting the remaining clusters, which cover the ring in such a way (using a transitive relationship) that at least an approximation to a single cluster can be claimed.

Although the principal purpose of SEQOPTICS (Chapter 3) was application-

oriented, the code was implemented on our own from OPTICS specifications, the algorithm underlying it. This code was developed to be compatible with that of MABAC, in the same programming language and open to a similar user interface. SEQOPTICS includes other features, especially for interacting with data taken from heterogeneous sources, editing and storing it in a repository, or a data warehouse, in future study.

A testing need arose for SEQOPTICS. The special features of this algorithm merited attention, i.e., its visualization features for cluster structure. As a representative of a “density based” clustering method, it is different from MABAC, and raised some interest on account of this. Tests indicated that SEQOPTICS might work for data of forms expected in subsequent real-world use.

A similar ‘Methods and Techniques’ study occurred with hierarchical clustering, prior to use in the analysis of the allosteric network in the protein family, Ligand Gated Ion Channels (chapter 4). In this case visualization is also an important feature and how it might serve the application was explored. Again, the work served as a possible stimulant for future work with MABAC in relation to a visualization subunit. Understanding internals of implementations could prove useful in assessing a future possibility wherein MABAC, SEQOPTICS, and some other clustering methods might in some fashion be blended, at an algorithm or a system level.

In chapters 2 and 5, Fuzzy C-means and MABAC were applied. The combination of work in the two chapters was instructive especially to the important complementary roles different clustering methods may play in direct application. A distributed agent with a mix and match of clustering methods might be helpful in this case. Clustering

in chapter 5 should be viewed broadly in its potential, as a preprocessing step leading to classification with neural networks, and as a co- or post- processor to the same classification and other statistical and modeling attacks.

6.3 System Views

Clustering methods and techniques in this dissertation ultimately get employed in systems contexts. The rationale for development and testing of a variety of clustering methods is rooted in the desire to embed the methods in larger systems aimed at solving real-world problems, e.g., protein structural analysis and patient rehabilitation study.

In chapter 3, SEQOPTICS adapted OPTICS clustering to produce a protein clustering system aimed at improved performance relative to some existing systems. Protein sequence data was extracted from large on-line data bases. A distance measure for every pair of sequences was computed according to a method employing a normalized Smith-Waterman score. The computed distance (score) matrix was then subjected to OPTICS clustering and results were compared with existing methods according to criteria including the *Jacquard coefficient*, *Recall* and *Precision*. Two other systems are compared in their default modes and SEQOPTICS exceeds their performances.

The whole system starting from data extracting and ending with evaluation offers future opportunities all along the path. It has been suggested to make input data collections larger. Using other methods in a cooperative mode could nicely complement the competitive mode used. Further explorations along these lines are among

recommendations for follow-up studies.

In chapter 4, the system embedded an adopted hierarchical clustering method, aiming at unveiling a network of Ligand Gated Ion Channels (LGICs) residues. Similar to the study with SEQOPTICS, the system started with protein sequences of LGICs extracted from on-line databases. These sequences were first aligned using an existing tool, ClustalW, using default parameters. The aligned profile was then further analyzed with statistical coupling analysis and correlation analysis, each of which creates a score matrix. These score matrices were subjected to clustering and coupled (correlated) residues were clustered together. These sites were then mapped onto a 3-D structural model of LGICs to reveal an allosteric network residues of LGICs.

A future here, from the clustering component point of view, might be to employ our generated and adapted systems, MABAC and SEQOPTICS, the former with added visualization capabilities. Larger or alternative samples are also indicated. The processing prior to and after clustering can be expected to evolve and at least indirectly affect clustering needs.

In chapter 5, a data repository system with clear data mining components applied both classification and clustering while addressing problems in stroke rehabilitation studies. The system started with raw data from patients (accelerometers, mainly) and therapists' assessments (from video recordings of the patients) and loaded them into the central repository (after certain cleaning and checking operations), rendering them available for further processing. The processing ranged from routine retrieval to complex multi-stage model-based probing. The modeling part of the effort is seen in the chapter's discussion of neural networks and clustering work. MABAC, the topic

of chapter 1, played a role in this work, specifically, as an integrated component into a real-world study. Fuzzy clustering played a supporting role. Considerable effort has been put into development and implementation work so that component systems can work together in complex and integrated ways. The array of tools and successes with neural networks demonstrate that all the major goals of this study have been at least potentially met.

Apart from future work on clustering methods and systematics, analogous to that in the other projects of this study, the schemes of collecting data, storing them, processing them and previously obtained information, and, feedback of results from processing, lay the claim that IMAR is a budding data warehousing system. The lessons learned from warehouse work on such systems can now be applied to IMAR for future study. One of the most important considerations, results feedback, is perhaps the weakest link in the chain of activity so far. A few words are provided on how other computer and information sciences can be most helpful. Information, in particular, from artificial intelligence, simulation (especially simulation environments) and decision support systems, has grown in importance, especially in computing science circles. Finally, hand-held devices, another topic of our research group, will be used in the computing loop. Together with rapidly evolving wireless communications, these may represent several opportunities, in data collection and otherwise helping to speed up the overall process from data collection to model results and associated feedback.

6.4 Contributions: Software Systems and Publications

A summary of core contributions of this thesis work is now presented, along with a precis of publication activity, this being taken as an important measure of contribution alongside an intended systematic and comprehensive attack on research topics. attention is called to:

- Design and implementation of MABAC, a clustering scheme whose augmented bond notion and its matrix base are of interest (chapter 2). Testing, though limited so far, adds to the interest. Two first author conference papers emerge from this topic [26, 25]. One of these is a regional conference and the other is international in scope; both involve refereeing and evaluations.
- Implementation of SEQOPTICS, a protein clustering method based on an existing method, OPTICS (chapter 3). The implementation, in the same programming language and with some similarities in design goals, adds potential for future blendings of activity at the algorithm-method level and at the systems level. A first author paper for part of this work was accepted by Symposium of Computations in Bioinformatics and Bioscience (SCBB06) in conjunction with the International Multi-Symposiums on Computer and Computational Sciences (IMSCCS—06), The paper will appear in the proceedings for the symposiums.
- Application of clustering incorporating a new, previously devised scheme for coupling analysis and correlation analysis to identify a network of residues in Ligand Gated Ion Channels (chapter 4). In this case clustering is viewed in a “deep” embedding in a surrounding complex biological modeling systems. An

additional interest lies in the fact that clustering is utilized more than once on two diverse data streams resulting in strengthened results. A first author paper for this part of the dissertation work has been accepted by Journal of Biological Chemistry for forthcoming publication [23].

- Design and implementation of IMAR, a data repository system supporting data mining activity, for aid in managing and processing data from stroke rehabilitation patients under a methodology pioneered by researchers at the UAB (chapter 5). Conference abstracts have been published on this topic [22, 100] and a first author journal paper is to be offered similar to the coverage in chapter 5.

LIST OF REFERENCES

- [1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data*, pages 94–105, Seattle, Washington,, June 1998.
- [2] S. Ahmed, N.E. Mayo, J. Higgins, N.M. Salbach, L. Finch, and S.L. Wood-Dauphinee. The Stroke Rehabilitation Assessment of Movement (STREAM): a comparison with other measures used to evaluate effects of stroke and rehabilitation. *Physical Therapy*, 83(7):617–30, 2003.
- [3] S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
- [4] SF Altschul, TL Madden, AA Schaffer, J Zhang, Z Zhang, W Miller, and DJ Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl. Acids Res.*, 25(17):3389–3402, 1997.
- [5] Jahanshah Amin. A Single Hydrophobic Residue Confers Barbiturate Sensitivity to γ -Aminobutyric Acid Type C Receptor. *Mol Pharmacol*, 55(3):411–423, 1999.
- [6] B. Amit and B. Baldwin. Algorithms for Scoring Coreference Chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, May 1998.
- [7] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jorg Sander. OPTICS: Ordering Points To Identify the Clustering Structure. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 49–60. ACM Press, 1999.
- [8] Amos Bairoch and Rolf Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucl. Acids. Res.*, 28(1):45–48, 2000.
- [9] R. Bakeman and J. M. Gottman. *Observing Interaction: An Introduction to Sequential Analysis*. Cambridge: Cambridge University Press, 1986.
- [10] Alex Bateman, Ewan Birney, Lorenzo Cerruti, Richard Durbin, Laurence Etwiller, Sean R. Eddy, Sam Griffiths-Jones, Kevin L. Howe, Mhairi Mar-

- shall, and Erik L. L. Sonnhammer. The Pfam Protein Families Database. *Nucl. Acids. Res.*, 30(1):276–280, 2002.
- [11] Dennis A. Benson, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and David L. Wheeler. GenBank: update. *Nucl. Acids. Res.*, 32(90001):D23–26, 2004.
- [12] Andrew J. Boileau and Cynthia Czajkowski. Identification of Transduction Elements for Benzodiazepine Modulation of the GABAA Receptor: Three Residues Are Required for Allosteric Coupling. *J. Neurosci.*, 19(23):10213–10220, 1999.
- [13] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, New York, NY, USA, 1992. ACM Press.
- [14] Cecilia Bouzat, Fernanda Gumilar, Guillermo Spitzmaul, Hai-Long Wang, Diego Rayes, Scott B. Hansen, Palmer Taylor, and Steven M. Sine. Coupling of agonist binding to channel gating in an ACh-binding protein linked to an ion channel. *Nature*, 430(7002):896–900, 2004.
- [15] Katjuša Brejc, Willem J. van Dijk, Remco V. Klaassen, Mascha Schuurmans, John van der Oost, August B. Smit, and Titia K. Sixma. Crystal structure of an ACh-binding protein reveals the ligand-binding domain of nicotinic receptors. *Nature*, 411:269–76, 2001.
- [16] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [17] Sudha Chakrapani, Timothy D. Bailey, and Anthony Auerbach. Gating Dynamics of the Acetylcholine Receptor Extracellular Domain. *J. Gen. Physiol.*, 123(4):341–356, 2004.
- [18] Chang-sheng S. Chang, Riccardo Olcese, and Richard W. Olsen. A Single M1 Residue in the $\beta 2$ Subunit Alters Channel Gating of GABAA Receptor in Anesthetic Modulation and Direct Activation. *J. Biol. Chem.*, 278(44):42821–42828, 2003.
- [19] YongChang Chang, Emmanuel Ghansah, Yonghui Chen, Jiawei Ye, and David S. Weiss. Desensitization Mechanism of GABA Receptors Revealed by Single Oocyte Binding and Receptor Function. *J. Neurosci.*, 22(18):7982–7990, 2002.

- [20] Yongchang Chang and David S. Weiss. Allosteric Activation Mechanism of the $\alpha 1\beta 2\gamma 2$ γ -Aminobutyric Acid Type A Receptor Revealed by Mutation of the Conserved M2 Leucine. *Biophys. J.*, 77(5):2542–2551, 1999.
- [21] Jean-Pierre Changeux and Stuart J Edelman. Allosteric Receptors after 30 Years. *Neuron*, 21(5):959–980, 1998.
- [22] Y. H. Chen, G. Uswatte, K. D. Reilly, and L. Hobbs. Data mining and simulation approaches in stroke rehabilitation programs. In *The Huntsville Simulation Conference: HSC 2005*, Huntsville, AL, October 2005.
- [23] Yonghui Chen, Kevin Reilly, and Yongchang Chang. Evolutionarily conserved allosteric network in the cys-loop family of ligand-gated ion channels revealed by statistical covariance analyses. *J. Biol. Chem.*, page M600349200, 2006.
- [24] Yonghui Chen, Kevin D. Reilly, Alan P. Sprague, , and Zhijie Guan. SEQOPTICS: Protein Sequence Clustering Method. In *Proceedings The International Multi-Symposiums on Computer and Computational Sciences 2006*, 06 2006.
- [25] Yonghui Chen and Alan Sprague. Exploration on the Commonality of Hierarchical Clustering Algorithms. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 246–247. ACM Press, 2004.
- [26] Yonghui Chen, Alan P. Sprague, and Kevin Reilly. MABAC - Matrix Based Clustering Algorithm. In *MSV/AMCS*, pages 439–443, 2004.
- [27] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.
- [28] Mark A. Danielson, Randal B. Bass, and Joseph J. Falke. Cysteine and Disulfide Scanning Reveals a Regulatory α -Helix in the Cytoplasmic Domain of the Aspartate Receptor. *J. Biol. Chem.*, 272(52):32878–32888, 1997.
- [29] Maria Jose De Rosa, Diego Rayes, Guillermo Spitzmaul, and Cecilia Bouzat. Nicotinic Receptor M3 Transmembrane Domain: Position 8' Contributes to Channel Gating. *Mol Pharmacol*, 62(2):406–414, 2002.
- [30] C.L. Donald, L. Jennifer, Pinkham, and Charles F. Stevens. Role of a key cysteine residue in the gating of the acetylcholine receptor. *Neuron*, 6(1):31–40, 1991.

- [31] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucl. Acids. Res.*, 30(7):1575–1584, 2002.
- [32] Anton J. Enright and Christos A. Ouzounis. GeneRAGE: a robust algorithm for sequence clustering and domain detection. *Bioinformatics*, 16(5):451–457, 2000.
- [33] L. Ertz, M. Steinbach, and V. Kumar. Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach. In *Text Mine '01, Workshop on Text Mining, First SIAM International Conference on Data Mining*, 2001.
- [34] J. Fahrenberg, F. Foerster, M. Smeja, and W. Muller. Assessment of posture and motion by multichannel piezoresistive accelerometer recordings. *Psychophysiology*, 34(5):607–12, 1997.
- [35] Anthony A. Fodor and Richard W. Aldrich. Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. *Proteins: Structure, Function, and Bioinformatics*, 56(2):211–21, 2004.
- [36] Anthony A. Fodor and Richard W. Aldrich. On Evolutionary Conservation of Thermodynamic Coupling in Proteins. *J. Biol. Chem.*, 279(18):19046–19050, 2004.
- [37] Chris Fraley. Algorithms for Model-Based Gaussian Hierarchical Clustering. *SIAM J. Sci. Comput.*, 20(1):270–281, 1998.
- [38] Gad Getz, Erel Levine, and Eytan Domany. Coupled two-way clustering analysis of gene microarray data. *PNAS*, 97(22):12079–12084, 2000.
- [39] U. Gobel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins*, 18(4):309–17, 1994.
- [40] C.V. Granger. The emerging science of functional assessment: our tool for outcomes analysis. *Arch Phys Med Rehabil.*, 79:235–40, 1998.
- [41] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*. MIT Press, 1994.
- [42] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.

- [43] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, 25(5):345–366, 2000.
- [44] Martin J. Gunthorpe and Sarah C. R. Lummis. Conversion of the Ion Selectivity of the 5-HT_{3A} Receptor from Cationic to Anionic Reveals a Conserved Feature of the Ligand-gated Ion Channel Superfamily. *J. Biol. Chem.*, 276(14):10977–10983, 2001.
- [45] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.
- [46] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, San Francisco, 2001.
- [47] J. Hartigan. *Clustering Algorithms*. Wiley, 1975.
- [48] Mark E. Hatley, Steve W. Lockless, Scott K. Gibson, Alfred G. Gilman, and Rama Ranganathan. Allosteric determinants in guanine nucleotide-binding proteins. *PNAS*, 100(24):14445–14450, 2003.
- [49] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994.
- [50] Francisco Hernandez, Purushotham Bangalore, and Kevin Reilly. End-user tools for grid computing. In *WEUSE I: Proceedings of the first workshop on End-user software engineering*, pages 1–5, New York, NY, USA, 2005. ACM Press.
- [51] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feed-forward networks. *Neural Networks*, 3(5):551–560, 1990.
- [52] A. Jaccard. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaudoises Sci. Nat.*, 44:223–270, 1908.
- [53] Anil K. Jain, Jianchang Mao, and K.m. Mohiuddin. Artificial Neural Networks: A Tutorial. *Computer*, 29(3):31–44, 1996.
- [54] J. S. R. Jang and N. Gulley. *Fuzzy Logic Toolbox*. The Mathworks Inc., 24 Prime Park Way, Natick MA 01760-1500, January 1995.

- [55] Arthur Karlin and Myles H. Akabas. Toward a structural basis for the function of nicotinic acetylcholine receptors and their cousins. *Neuron*, 15:1231–1244, 1995.
- [56] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar. Chameleon: Hierarchical Clustering Using Dynamic Modeling. *Computer*, 32(8):68–75, 1999.
- [57] H. Kawaji, Y. Yamaguchi, H. Matsuda, and A Hashimoto. A graph-based clustering method for a large set of sequences using a graph partitioning algorithm. *Genome Informatics*, 17:93–102, 2001.
- [58] Andreas Keil, Thomas Elbert, and Edward Taub. Relation of Accelerometer and EMG Recordings for the Measurement of Upper Extremity Movement. *Journal of Psychophysiology*, 13(2):77–82, 1999.
- [59] K. Kiani, C.J. Snijders, and E.S. Gelsema. Recognition of daily life motor activity classes using an artificial neural network. *Arch Phys Med Rehabil.*, 79(2):147–154, 1998.
- [60] Sun Kim and Arvind Gopu. BAG: A Graph Theoretic Sequence Clustering Algorithm, 2004.
- [61] Teuvo Kohonen, editor. *Self-organizing maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [62] Antje Krause. *Large Scale Clustering of Protein Sequences*. PhD thesis, der Universitat Bielefeld, 2002.
- [63] Antje Krause, Jens Stoye, and Martin Vingron. Large scale hierarchical clustering of protein sequences. *BMC Bioinformatics*, 6(1):15, 2005.
- [64] Evgenia V Kriventseva, Margaret Biswas, and Rolf Apweiler. Clustering and analysis of protein families. *Current Opinion in Structural Biology*, 11(3):334–339, 2001.
- [65] Silke Lankiewicz, Nicole Lobitz, Christian H. R. Wetzell, Rainer Rupprecht, Gunter Gisselmann, and Hanns Hatt. Molecular Cloning, Functional Expression, and Pharmacological Characterization of 5-Hydroxytryptamine₃ Receptor cDNA and Its Splice Variants from Guinea Pig. *Mol Pharmacol*, 53(2):202–212, 1998.
- [66] S. Lawrence, C.L.Giles, and A.C. Tsoi. What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation. Technical

Report UMIACS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, University of Maryland, College Park,, April 1996.

- [67] Richard P. Lippmann. An introduction to computing with neural nets. *SIGARCH Comput. Archit. News*, 16(1):7–25, 1988.
- [68] Steve W. Lockless and Rama Ranganathan. Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. *Science*, 286(5438):295–299, 1999.
- [69] Joseph W. Lynch, Sundran Rajendra, Kerrie D. Pierce, Cheryl A. Handford, Peter H. Barry, and Peter R. Schofield. Identification of intracellular and extracellular domains mediating signal transduction in the inhibitory glycine receptor chloride channel. *EMBO J.*, 16(1):110–120, 1997.
- [70] W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- [71] Zbigniew Michalewicz. *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1994.
- [72] S.J. Mihic, Q. Ye, M.J. Wick, V.V. Koltchine, M.D. Krasowski, S.E. Finn, M.P. Mascia, C.F. Valenzuela, K.K. Hanson, E.P. Greenblatt, R.A. Harris, and N. L. Harrison. Sites of alcohol and volatile anaesthetic action on GABA(A) and glycine receptors. *Nature*, 389(6649):385–389, 1997.
- [73] Angela Miko, Elena Werby, Hui Sun, Julia Healey, and Li Zhang. A TM2 Residue in the β 1 Subunit Determines Spontaneous Opening of Homomeric and Heteromeric γ -Aminobutyric Acid-gated Ion Channels. *J. Biol. Chem.*, 279(22):22833–22840, 2004.
- [74] A. Miyazawa, Y. Fujiyoshi, and N. Unwin. Structure and gating mechanism of the acetylcholine receptor pore. *Nature*, 423(6943), 2003.
- [75] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247(4):536–540, 1995.
- [76] Koichi Nishikawa and Neil L. Harrison. The Actions of Sevoflurane and Desflurane on the γ -Aminobutyric Acid Receptor Type A: Effects of TM2 Mutations in the α and β Subunits. *Anesthesiology*, 99(3):678–684, 2003.

- [77] Richard W. Olsen, Chang-Sheng S. Chang, Guodong Li, H. Jacob Hanchar, and Martin Wallner. Fishing for allosteric sites on GABAA receptors. *Biochemical Pharmacology*, 68(8):1675–1684, 2004.
- [78] Z. H. Pan, X. Zhang, and S. A. Lipton. Redox modulation of recombinant human GABAA receptors. *Neuroscience*, 98(2):333–338, 2000.
- [79] Zhuo-Hua Pan, Dongxian Zhang, Xishan Zhang, and Stuart A. Lipton. Agonist-induced closure of constitutively open γ -aminobutyric acid channels with mutated M2 domains. *PNAS*, 94(12):6490–6495, 1997.
- [80] W R Pearson and D J Lipman. Improved Tools for Biological Sequence Analysis. *Proc. Natl. Acad. Sci. U.S.A.*, 85:2444–2448, 1988.
- [81] P. Pipenbacher, A. Schliep, S. Schneckener, A. Schonhuth, D. Schomburg, and R. Schrader. ProClust: improved clustering of protein sequences with an extended graph-based approach. *Bioinformatics*, 18(90002):182S–191, 2002.
- [82] K. D. Reilly. Agents and Lightweight Use of Logic: Combined Simulation, Logic and Neural Network Nodes. In *HSC - The Huntsville Simulation Conference*, pages 132–137, 2001.
- [83] K.D. Reilly, N. Bray, J. Drake, M. E. Golding, D. Chen, and W. Pan. Intelligent System Modeling using Neural Networks and Animation within A Distributed Programming Context. In *Proc. 1997 Summer Simulation Conference - Society for Computer Simulation, Int'l.*, pages 735–740, 1997.
- [84] Kevin D. Reilly. Agent computing themes in biologically inspired models of learning and development. *International Journal of Developmental Neuroscience*, 20(3-5), 2002.
- [85] Kevin D. Reilly, Norman W. Bray, and Michael Jackson. Approaches to Cognitive System Simulation: Architectures and Animations. In *Proc. Annual Simulation Symposium*, pages 198–207, Los Alamitos, CA, USA, 2000. IEEE Computer Society.
- [86] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Min. Knowl. Discov.*, 2(2):169–194, 1998.
- [87] Jinwook Seo and Ben Shneiderman. Interactively Exploring Hierarchical Clustering Results. *Computer*, 35(7):80–86, 2002.

- [88] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. WaveCluster: A Wavelet Based Clustering Approach for Spatial Data in Very Large Databases. *VLDB J.*, 8(3-4):289–304, 2000.
- [89] R. Sibson. SLINK:an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.
- [90] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.
- [91] Heidi Sveistrup. Motor rehabilitation using virtual reality. *Journal of Neuro-Engineering and Rehabilitation*, 1(1):10, 2004.
- [92] Lise Talbot, Chantal Viscogliosi, Johanne Desrosiers, Claude Vincent, Jacqueline Rousseau, and Line Robichaud. Identification of rehabilitation needs after a stroke: an exploratory study. *Health and Quality of Life Outcomes*, 2(1):53, 2004.
- [93] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.
- [94] P.T. Tangeman, D.A. Banaitis, and A.K. Williams. Rehabilitation of chronic stroke patients: changes in functional performance. *Arch Phys Med Rehabil.*, 71(11):876–80, 1990.
- [95] Edward Taub, Sharon Landesman Ramey, Stephanie DeLuca, and Karen Echols. Efficacy of Constraint-Induced Movement Therapy for Children With Cerebral Palsy With Asymmetric Motor Impairment. *Pediatrics*, 113(2):305–312, 2004.
- [96] Edward Taub, Gitendra Uswatte, Johanna H. van der Lee, Gustaaf J. Lankhorst, Lex M. Bouter, and Robert C. Wagenaar. Constraint-Induced Movement Therapy and Massed Practice Response. *Stroke*, 31(4):983–991, 2000.
- [97] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 11(22):4673–4680, 1994.
- [98] G.M. TSuel, S.W. Lockless, M.A. Wall, and R. Ranganathan. Evolutionarily conserved networks of residues mediate allosteric communication in proteins. *Nat Struct Biol.*, 10(1):59–69, 2003.

- [99] Nigel Unwin. Refined Structure of the Nicotinic Acetylcholine Receptor at 4Å Resolution. *Journal of Molecular Biology*, 346(4):967–989, 2005.
- [100] G. Uswatte, Y. H. Chen, and K. D. Reilly. Remote objective monitoring of adherence to a home rehabilitation program for upper extremity hemiparesis. In *8th Annual Rehabilitation Psychology Conference*, Reno, NV, March 2006.
- [101] G. Uswatte, W. Miltner, H. Walker, S. Spraggins, S. Moran, J. Calhoun, C. Beatty, and E. Taub. Accelerometers in rehabilitation: Objective measurement of extremity use at home. *Rehabilitation Psychology*, 42(139), 1997.
- [102] Gitendra Uswatte, Wolfgang H. R. Miltner, Benjamin Foo, Maneesh Varma, Scott Moran, and Edward Taub. Objective Measurement of Functional Upper-Extremity Movement Using Accelerometer Recordings Transformed With a Threshold Filter. *Stroke*, 31(3):662–667, 2000.
- [103] Gitendra Uswatte-Aratchi. *Accelerometry: a new technique for objectively measuring the actual real-world extremity use in rehabilitation patients*. PhD thesis, The University of Alabama at Birmingham, 2000.
- [104] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A model-theoretic coreference scoring scheme. In *MUC6 '95: Proceedings of the 6th conference on Message understanding*, pages 45–52, Morristown, NJ, USA, 1995. Association for Computational Linguistics.
- [105] Ke Wang, Chu Xu, and Bing Liu. Clustering Transactions Using Large Items. In *CIKM*, pages 483–490, 1999.
- [106] Wei Wang, Jiong Yang, and Richard R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 186–195. Morgan Kaufmann Publishers Inc., 1997.
- [107] Cathy H. Wu, Hongzhan Huang, Leslie Arminski, Jorge Castro-Alvear, Yongxing Chen, Zhang-Zhi Hu, Robert S. Ledley, Kali C. Lewis, Hans-Werner Mewes, Bruce C. Orcutt, Baris E. Suzek, Akira Tsugita, C. R. Vinayaka, Lai-Su L. Yeh, Jian Zhang, and Winona C. Barker. The Protein Information Resource: an integrated public resource of functional annotation of proteins. *Nucl. Acids. Res.*, 30(1):35–37, 2002.
- [108] Golan Yona, Nathan Linial, and Michal Linial. ProtoMap: automatic classification of protein sequences and hierarchy of protein families. *Nucl. Acids. Res.*, 28(1):49–55, 2000.

- [109] Oren Zamir and Oren Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998.
- [110] Chengcui Zhang and Xin Chen. Region-Based Image Clustering and Retrieval Using Multiple Instance Learning. In *CIVR*, pages 194–204, 2005.
- [111] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the SIGMOD 96*, pages 103–114, 1996.