# A VISUAL ANALYTICS APPROACH TO FEATURE DISCOVERY AND SUBSPACE EXPLORATION IN PROTEIN FLEXIBILITY MATRICES

by

Scott Anthony Barlowe

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2011

Approved by:

_____

Dr. Jing Yang

_____

Dr. Dennis R. Livesay

_____

Dr. Donald J. Jacobs

_____

Dr. Heather Richter Lipford

_____

Dr. Andrew Willis

©2011
Scott Anthony Barlowe
ALL RIGHTS RESERVED

# ABSTRACT

SCOTT ANTHONY BARLOWE. A visual analytics approach to feature discovery and subspace exploration in protein flexibility matrices. (Under the direction of DR. JING YANG)

The vast amount of information generated by domain scientists makes the transition from data to knowledge difficult and often impedes important discoveries. For example, the knowledge gained from protein flexibility data sets can speed advances in genetic therapies and drug discovery. However, these models generate so much data that large scale analysis by traditional methods is almost impossible. This hinders biomedical advances. Visual analytics is a new field that can help alleviate this problem. Visual analytics attempts to seamlessly integrate human abilities in pattern recognition, domain knowledge, and synthesis with automatic analysis techniques. I propose a novel, visual analytics pipeline and prototype which eases discovery, comparison, and exploration in the outputs of complex computational biology datasets. The approach utilizes automatic feature extraction by image segmentation to locate regions of interest in the data, visually presents the features to users in an intuitive way, and provides rich interactions for multi-resolution visual exploration. Functionality is also provided for subspace exploration based on automatic similarity calculation and comparative visualizations. The effectiveness of feature discovery and subspace exploration is shown through a user study and user scenarios. Feedback from analysts confirms the suitability of the proposed solution to domain tasks.

# ACKNOWLEDGMENTS

I would like to thank my advisor, Jing Yang, for allowing me to have great freedom to experiment with new approaches, for providing me with valuable feedback througout this process, and for never hesitating to work me into her schedule. I would also like to thank Dennis Livesay and Don Jacobs for their incredible patience when sharing their knowledge in the simplest terms, for taking the time to listen to my ideas, and for becoming active participants in this work. Without all of these faculty members, this collaborative effort would not have been possible.

This work also benefited from the effort of several others. I would like to thank Yujie Liu for help with the user study, Deeptak Verma for being a valuable source of feedback, and James Mottenon for supplying data.

Most of all, I would like to thank my wife. Becky, thank you for your patience all those many nights that I was either at school, traveling to and from school, or staying up until the next morning working on projects. Without your encouragement and your help, I would have fallen well short of my goal.

TABLE OF CONTENTS

LIST OF FIGURES viii

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: LITERATURE REVIEW 4

2.1 Current Systems for Protein Analysis 4

2.2 Matrix Visualization 8

2.3 Feature Extraction 9

2.3.1 Extraction Techniques 9

2.3.2 Feature and Change Visualization 14

CHAPTER 3: MODELS, DATA, AND TASKS 18

3.1 Protein Construction 18

3.2 A Computational Model 20

3.3 Data Formulation and Tasks 22

3.3.1 Model Outputs 23

3.3.2 High-level Tasks 27

CHAPTER 4: WAVEMAP – INTERACTIVE FEATURE DISCOVERY 35

4.1 System Components 38

4.1.1 Feature Extraction 38

4.1.2 Overview 42

4.1.3 Feature Exploration 46

4.1.4 Detailed Analysis 48

4.2 Scenario 49

4.3 Evaluation 52

4.3.1 User Study 52

4.3.2 Expert Evaluation 59

4.4 Discussion 62

4.4.1 Extensions 62

4.4.2 Limitations 66

CHAPTER 5: EXTENSIONS FOR SUBSPACE EXPLORATION 68

5.1 Task Refinement 68

5.2 Background in Subspace Exploration 69

5.2.1 Protein Subspaces 69

5.2.2 Subspaces in Visualization 71

5.3 Plot Carving 78

5.3.1 Grid Sections 78

5.3.2 Histogram Sorting 78

5.3.3 Interactions 83

5.4 Subpsace Clusters 85

5.5 Sliding Subspaces 86

5.5.1 Context Scans 86

5.5.2 Interactions 89

5.5.3 Outlier Detection 90

5.5.4 Subspace Similarity Types 92

5.6 Evaluation 93

5.6.1 Example Use 93

5.6.2 Expert Evaluation 98

CHAPTER 6: CONCLUSION AND FUTURE WORK 101

REFERENCES 103

# LIST OF FIGURES

FIGURE 1: VisAlign [41] 5

FIGURE 2: iVici [85] 6

FIGURE 3: Java Protein Dossier [72] and a lattice-based visualization 7

FIGURE 4: Wavelet families 11

FIGURE 5: Wavelet applications in information visualization 12

FIGURE 6: Wavelet lifting steps 13

FIGURE 7: Wavelet effect on a 1D signal 13

FIGURE 8: Protein contact map 15

FIGURE 9: Component plane array 16

FIGURE 10: Protein structure levels 19

FIGURE 11: Phi-psi angles for protein motion 20

FIGURE 12: Steps in the Distance Constraint Model 21

FIGURE 13: DCM outputs 23

FIGURE 14: Allosteric response plot 24

FIGURE 15: Cooperativity correlation matrix 26

FIGURE 16: Alignment example 28

FIGURE 17: Difficulty in flexibility analysis 29

FIGURE 18: Multiple correlation metrics 30

FIGURE 19: Early prototypes 31

FIGURE 20: CheY data set 32

FIGURE 21: System setup 35

FIGURE 22: Main interface with allosteric response 36

FIGURE 23: System framework 37

FIGURE 24: Wavelet iteration and outputs 40

FIGURE 25: Wavelet application to cooperativity correlation plots 41

FIGURE 26: Overview with MDS layout 43

FIGURE 27: Overview with jigsaw layout 43

FIGURE 28: Parameter labels 44

FIGURE 29: Pencil tool 46

FIGURE 30: Feature view 47

FIGURE 31: Detailed analysis 49

FIGURE 32: Scenario (part 1) 50

FIGURE 33: Scenario (part 2) 51

FIGURE 34: Scenario (part 3) 52

FIGURE 35: User study tasks 54

FIGURE 36: User study questionnaire 55

FIGURE 37: User study results 57

FIGURE 38: User study example 58

FIGURE 39: Other biological applications 63

FIGURE 40: Microarrays 65

FIGURE 41: Subspace and feature selection 73

FIGURE 42: Visa [2] 75

FIGURE 43: Web-based outlier subspace 76

FIGURE 44: Histogram view 79

FIGURE 45: MVE [4] 80

FIGURE 46: Shifted histograms 82

FIGURE 47: Trim tool 83

FIGURE 48: Effect of trim tool on subspaces 84

FIGURE 49: Subspace context dots 85

FIGURE 50: Bubble view 87

FIGURE 51: Bubble details 90

FIGURE 52: Free-hand tool 90

FIGURE 53: Outlier detection 91

FIGURE 54: Two distance types 92

FIGURE 55: Example Use (Part 1) 94

FIGURE 56: Example Use (Part 2) 95

FIGURE 57: Example Use (Part 3) 96

FIGURE 58: Streamlined bubble view 99

# CHAPTER 1: INTRODUCTION

Models describing physical or naturally occurring behavior can generate many possible outputs when different parameter settings are used. As models become more complex and more sensitive to parameter changes, the data produced becomes more difficult to analyze. This difficulty arises from both the amount and variation of produced data which must be considered quickly in parallel so that a model and the consequences of its outputs can be understood efficiently and accurately. Models are used in many applications but have become central in the prediction of protein behavior. One of the most fundamental predictors of protein behavior is *flexibility*, or a protein's ability to change shape under given circumstances. Accurate descriptions of protein flexibility are crucial for understanding the physiochemical mechanisms that underlie protein function [49] and could eventually be used to speed the drug discovery process. Making the prediction of protein behavior complex is that some portions of a protein are highly dynamic (flexible), whereas other regions are quite static (rigid). Compounding the problem is choosing the most appropriate set of parameters that accurately describe or predict protein flexibility.

These problems become evident when attempting to holistically analyze various types of flexibility and correlation plots. These plots are colored matrices representing either a flexibility index or residue to residue coupling behavior for varied proteins

and parameter settings. Small-scale examination impedes comparison among residues within plots possibly having several hundred variables. Scientists must not only identify abrupt differences, but subtle changes which easily escape manual inspection. This becomes more difficult as the similarities or differences among proteins becomes less dramatic, but no less important. In fact, the identification and comparison of small differences may prove to be crucial for learning how to alter a given behavior or explaining why two similar proteins behave differently. This difficulty increases as any insights gained must be placed in context across multiple proteins, environmental conditions, and correlation types.

The main contribution of this work is the design and development of a visual analytics prototype for model-based protein analysis. Two benefits emerged during the development of the prototype presented here. The first benefit is the formalization of data types and high-level tasks encountered in the course of using a particular protein flexibility model. Unfortunately, even public providers of biological data publish data sets with inconsistent formatting standards and require a great deal of preprocessing before software systems can be used in exploring them. Stein [81] discusses the challenges and current attempts for creating standardized data sets. However, attempts at standardization mostly address public repositories and often ignore the complexities introduced by scientists employing individualized workflows. Only when individualized workflows, including data types and high-level tasks, are exposed can common processes be identified and the level of standardization needed for fast discovery take place. Years of collaboration, sifting through much data, and several failed attempts at prototyping have resulted in the description of the data sets

for this particular model. This will not only aid in the development of new *processing tools* for this model but also for other domains encountering similar data structures both within and outside protein analysis. Furthermore, this work defines the high-level tasks learned from observing domain analysis which will help in the development of new *analysis tools* tools so that insights can be made more efficiently.

The second benefit is the construction of a pipeline and prototype that can serve as a model for future visual analytics tools. The prototype, called WaveMap [3] combines automatic analysis and human perceptual abilities to provide a more complete view of data. WaveMap utilizes automatic feature extraction through image segmentation to guide users to points of interest in the colored plots at varying levels of granularity. Items that exhibit coarse-grain patterns of interest can be selected so that features representing a given fine-grain behavior can be explored. Simultaneously, the data is reduced through both automatic techniques and user interaction allowing scientists to focus on fewer items of interest. Extracted features can be efficiently compared to the original data values so that the results of automatic analysis can be mapped back to their original values. WaveMap also includes techniques for interactive selection and comparison of subspaces that help in distinguishing local behavior from global behavior. The effectiveness of the system is confirmed by a user study, user scenarios, and feedback from domain scientists.

# CHAPTER 2: LITERATURE REVIEW

The solution presented in this work employs many techniques from a wide range of fields. In the following sections, current visualization systems used in the exploration of two-dimensional protein data are surveyed and their shortcomings are exposed. Techniques in matrix visualization are then examined. Finally, possible feature extraction methods for emphasizing areas of interest and ways of visualizing those features are discussed.

## 2.1 Current Systems for Protein Analysis

While limited, there have been multiple attempts for applying visual analytics to protein structure and function. Because of the complexities associated with handling biologically based data, most attempts have either originated in or been heavily influenced by the bioinformatics community. Several systems are presented now.

Keim et al [41] state the importance of combining automatic and exploratory techniques during protein analysis. The authors present VisAlign (Figure 1) which helps users view and explore the alignment sequence of proteins and the correlation to a selected basis column. The system is comprised of the Alignment Viewer, Parameter Window, Mapping Window, Properties Window, and a 3D viewer. The Alignment Viewer shows each alignment as a column where each amino acid is color coded. All cells not correlated to the basis column are gray. The user can input maximum and

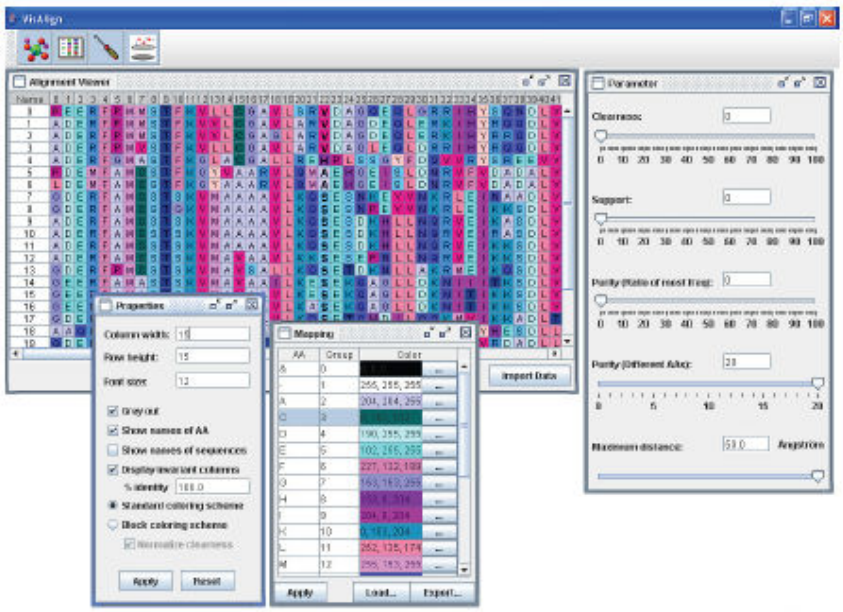


Figure 1: VisAlign [41].

minimum threshold values which are immediately available in the visualization. The Mapping Window allows the grouping of similar amino acids where similarity is based on an amino acid property or hypothesis of interest. The Properties Window controls visual properties such as zooming, fading, and cell size variation. The 3D Viewer shows a three-dimensional protein structure which is linked to the Alignment Viewer.

iVici [85] is a system for viewing protein-protein interactions encoded into symmetric, two-dimensional matrices (Figure 2). There are three modes. The general mode represents hierarchical clusters generated by an outside source. The comparative mode dissects symmetrical matrices into halves along the diagonal resulting in a triangular section. Matrices are compared by placing the triangular section from one matrix onto the top of a new matrix and the triangular section from the other matrix onto the bottom of the new matrix. The superimposed mode places one matrix onto

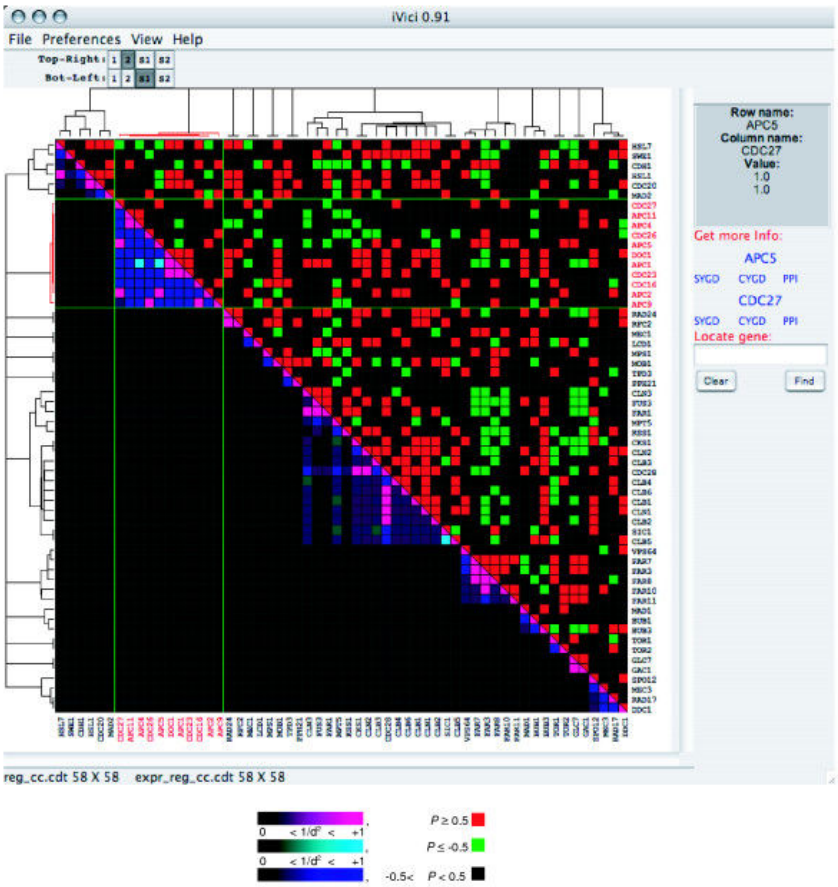



Figure 2: iVici [85].

another and the resulting color of each cell represents the intersection of values.

Java Protein Dossier [65] is shown in Figure 3(a). This system attempts to be a parameter and visualization warehouse for protein analysis. Java Protein Dossier accounts for many parameters that may need to be considered and employs elementary pixel displays for summary statistics. The authors claim that their molecular modeling capabilities include more than sixty parameters and can be deployed over the web. Windows showing protein sequences, structures, and parameters are coordinated.

One of the most complete tools specifically targeting protein structure and function exploration is based on lattice construction [72]. This system (Figure 3(b)) in-

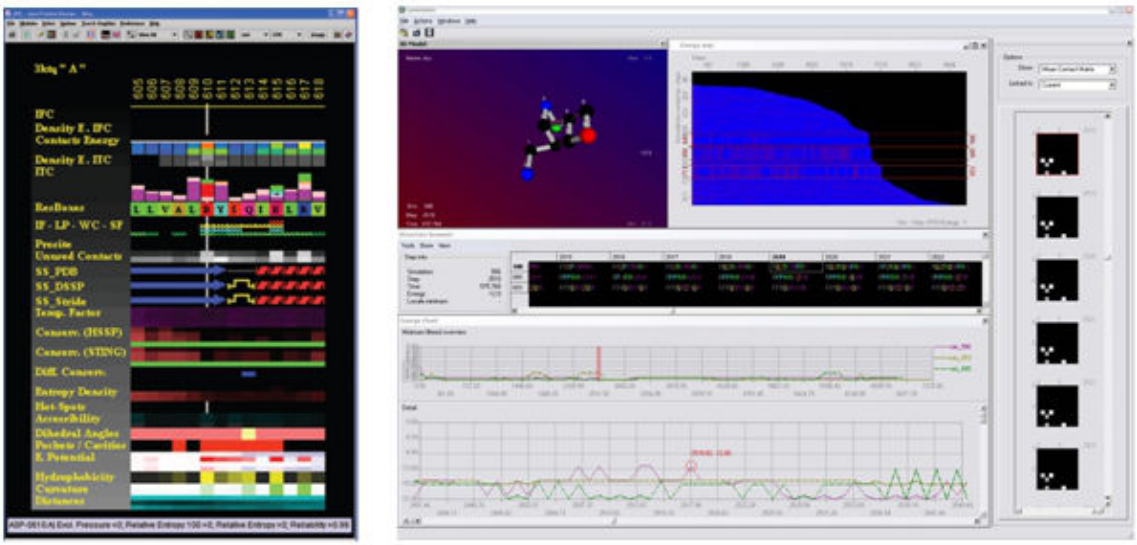



Figure 3: (a) Java Protein Dossier [65] and (b) lattice-based protein visualization [72].

cludes many of the techniques necessary for complete visual analysis. Focus+context, overview+detail, and multiple views are integrated into this platform. Energy landscapes can be examined through interactive line plots. Additionally, the work is connected to simulation models, provides a three dimensional lattice viewer, and includes contact matrix visualizations.

Although these representative systems attempt to utilize visualization of two-dimensional protein data, there are shortcomings. For example, most of these systems make little use of automatic analysis techniques which can ease the exploratory burden of users when searching for important data characteristics. In the cases where automatic techniques are present, highly-interactive tools are not available to guide the user to the places in the data which may be of greatest interest. Additionally, these and similar systems often only show summary information which can hide im-

portant relationships and subtle features. Such information is often crucial in spatial or temporal understanding. Finally, these systems (with the exception of [41]) only consider one or few proteins at a time and ignore the need for large scale analysis.

## 2.2 Matrix Visualization

Currently, scientists lack effective tools to conduct the above tasks for large flexibility data sets. Existing methods heavily depend on manual inspection of enlarged flexibility plots using Heatmaps [59], [58]. Subtle but important relationships and patterns may remain hidden even with zooming and distortion interactions. The large number of plots and the subtle differences both within each plot and among parameter sets are almost impossible to distinguish. These obstacles greatly hinder knowledge discovery.

Heatmaps are widely used in bioinformatics besides protein flexibility data visualization. Specifically, they are the most common representation for gene expression data [23]. Many of them can guide users to patterns such as clusters and outliers within the data. For example, HCE [78] and Java Treeview [76] enable users to identify clusters in microarray experiment data sets using hierarchical clustering algorithms and interactive visual exploration. Visualization methods developed for matrix data [84], [79] also allow users to find patterns through interactive visual exploration. However, the above techniques would not work for protein flexibility data visualization because of their heavy reliance on the grouping of similar rows and columns. Not only does each flexibility matrix value $i,j$ have a color-coded flexibility measure but also carries spatial significance reflecting residue ordering along the

three-dimensional protein structure. Any reordering or rearranging of rows, columns, or individual measures would disrupt the spatial context in which any flexibility measure occurs. Moreover, a large number of plots need to be examined simultaneously in our application while most of the above techniques only consider one matrix or array at a time.

## 2.3 Feature Extraction

### 2.3.1 Extraction Techniques

Feature extraction is an automatic analysis technique in signal and image processing used to draw out defining characteristics in a data set. Forlines and Balakrishnan [22] have shown that feature extraction through image segmentation can be helpful in visual search as target subimages become small and rare. This is a necessary aid since they note that search time is linearly correlated with the number of distractor objects. The authors perform user studies which differ in the presentation of targeted objects. Presentation differences include increasing target prevalence, re-layout, and space/time tradeoff. The authors found that all three of the image segmentation techniques improved search performance by reducing the false-negative error rates.

Many image analysis techniques for extracting features can be applied to flexibility plots so that difficult to detect patterns can be identified. For example, Principal Component Analysis (PCA) [38], [17] can be used to summarize features by finding the linear combinations of variables and then ordering the resulting components by variance [93]. It has been used in many image processing applications such as face recognition [69] and edge detection [73] but can be computationally taxing [88],

[18]. Additionally, the results can be difficult to interpret [96]. Fourier analysis [24] is another popular image analysis technique applied to many areas including feature extraction and dimension reduction [35]. Fourier frequencies can be linked to pixel value changes where low frequencies are associated with slowly varying pixel changes and high frequencies are associated with abrupt pixel changes [24]. A major drawback to this type of analysis is that frequency and spatial information cannot be conveyed at the same time.

Wavelet analysis is based on small signals (waves) of limited duration and varying frequency [24]. This type of transform allows the same frequency-based processing of pixel values as Fourier analysis. However, wavelets provide simultaneous frequency and spatial information with a multi-resolution approach that allows normally hidden features to be revealed. The wavelet transform (Equation (1)) results in a set of coefficients $W_\Psi(s,\tau)$ which represent the similarity between a function *f(x)* and a given wavelet transform $\Psi_{s,t}$. Similarity is measured as $s$, a scaling factor, and $\tau$, a translation factor, are varied resulting in a multiresolution view of *f(x)*.

$$W_\Psi(s,\tau) = \int_{-\infty}^{\infty} f(x)\Psi_{s,t}(x)dx \tag{1}$$

Different wavelets transforms, many of which can be grouped with others exhibiting similar characteristics to form wavelet families, can be substituted to extract desired characteristics. Examples of different wavelets are shown in Figure 4. Wavelets have been integrated into visualization tools for brushing applications [90] (Figure 5(a)), text analysis [55] (Figure 5(b)), and many scientific applications [19], [8].

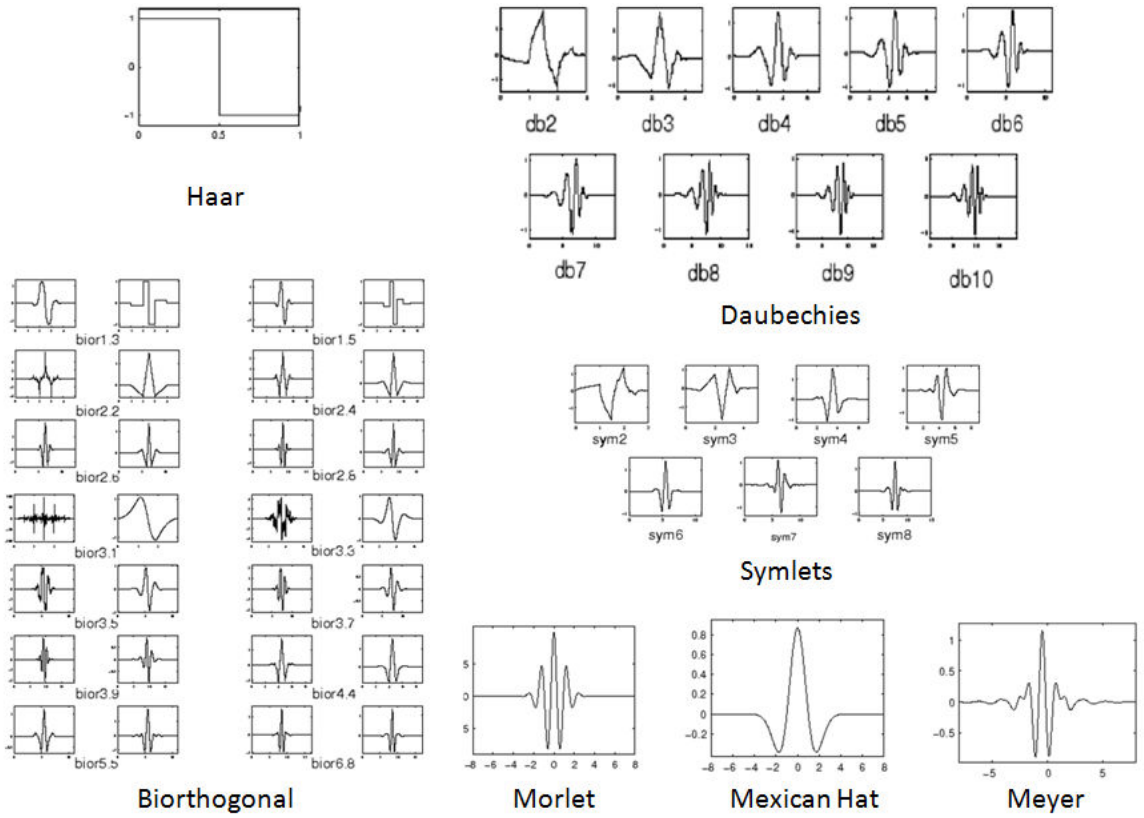



Figure 4: A sample of different wavelet families provided by a toolkit in MATLAB [53]

Wavelet lifting [82], [37] is an improvement to traditional wavelet analysis. Lifting is accomplished through repeated execution of a set of distinct steps. Steps for a one-dimensional signal include *split*, *predict*, and *update*. The first step, split, sorts the data into even and odd indices. Predict assumes that the correlation between a sample and its neighbors is high. In this step, the difference between the predicted value and the actual value is recorded in the odd entries. The update step uses the difference in the predict step to update the even entries. The even entries represent an approximation of the signal and the odd entries represent the details. Repeating these steps using the output as the input to the next sequence of split, predict, and update results in an increasingly coarse (if the approximations are used) or detailed

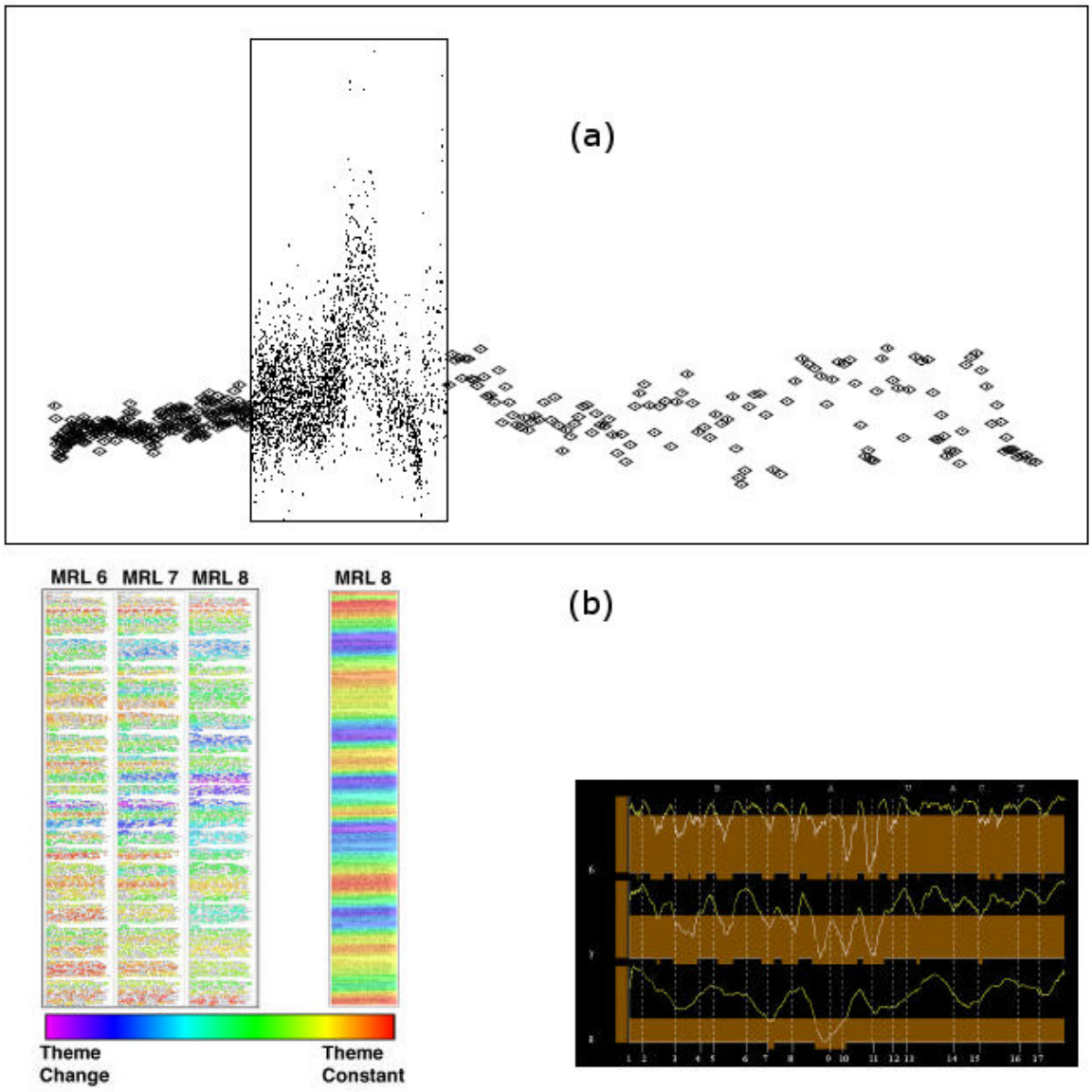



Figure 5: (a) Wavelet brushing shows the approximations (outside the brush) and details (inside the brush) [90]. (b) Wavelet energies superimposed on topics (left) wavelet energies on a line graph (right) [55].

(if the differences are used) view of the signal. The steps in lifting are shown in Figure 6 and the general effects of applying a wavelet to a one dimensional signal is shown in Figure 7.

The wavelet used can be changed by altering the predict and update stages. As with other wavelet implementations, different *filter banks* can be constructed by varying

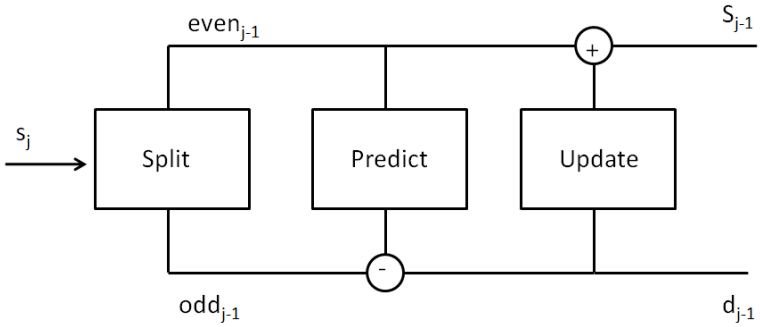



Figure 6: Steps in wavelet lifting are split, predict, and update. The result is a set of approximations, $S$ and a set of details, $d$.

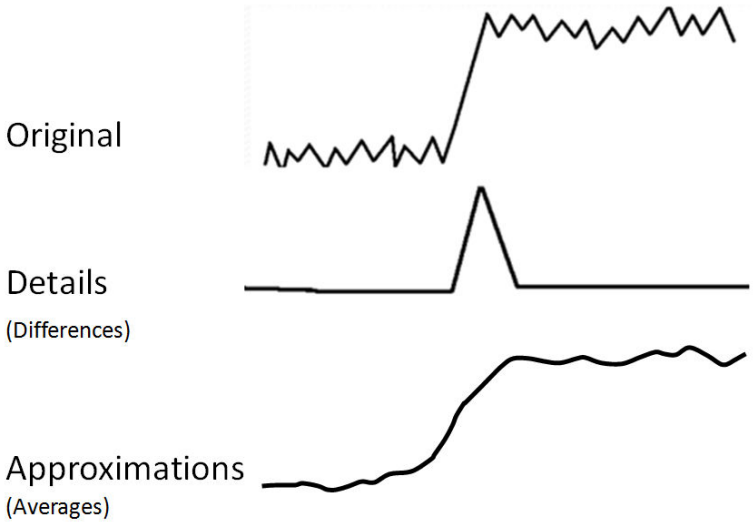



Figure 7: The general effect of applying a wavelet filter on an one-dimensional signal. Outputs include a set of approximations representing a coarse view of the original signal and a set of details representing a fine-grained view. Different variations of wavelts can be used to alter specific behavior.

what (either the signal approximation or detail) is taken as the input to the next step. A discrete, iterative process which uses no more memory than required for the original data matrix improves computational efficiency. Additionally, the results of lifting can be reversed by simply reversing the discrete steps used in transformation. This property enables the original data to be directly and quickly accessed from any level of decomposition.

There have been many cases of the application of feature extraction to data sets found in protein analysis [61], [12], [56]. The mining of protein contact maps is an excellent example where automatic feature extraction has been applied to protein image data. Protein contact maps are color images representing the chemical interactions for all amino acids in a protein and, because each map is unique, is a picture of protein structure [39]. Contact maps have also been used to inform scientists regarding a protein's secondary structure in addition to non-local features influencing the definition of its tertiary structure [32]. Characteristics which are color coded by Fernandes et al [39] include hydrophobic interactions, electrostatic interactions, and hydrogen bonds. Similar to the protein model described earlier (section 3.3.2), contact maps relate three-dimensional structure to a two-dimensional color image. Because the final output of the protein analysis is an image, the authors use content-based image retrieval (CBIR) as an automatic approach for similarity based searching. The authors report a successful grouping of similar structures based on their methods. Other activities in which scientists have been interested include pruning mined patterns and then clustering the results [32]. An example of a contact map is shown in Figure 8.

### 2.3.2 Feature and Change Visualization

Although most of the attempts for developing a complete system for modeling protein behavior have come from the bioinformatics domain, the visualization community has many techniques that can be applied to model-driven protein analysis. The most important contribution of the visualization community for the inspiration

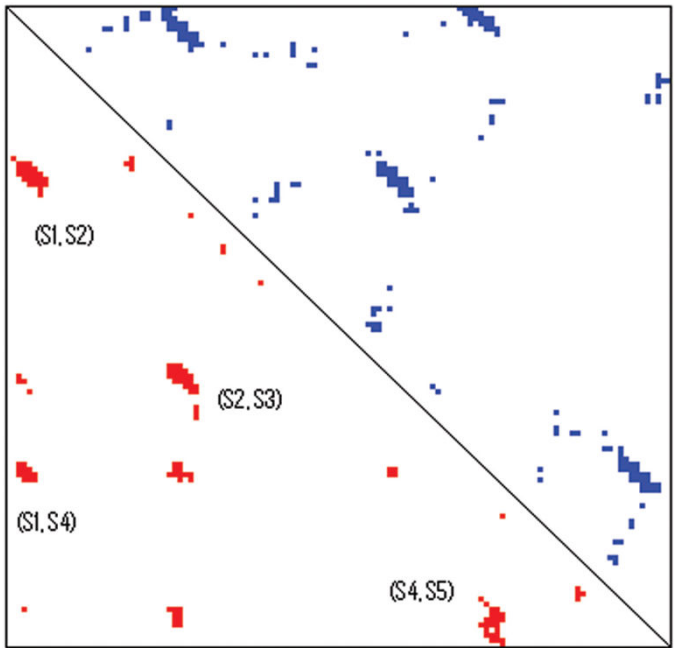



Figure 8: Contact map [86] color coding hydrophophic interactions, electrostatic interactions, and hydrogen bonds.

of this work is in feature visualization. Because the proposed system seeks to detect unexpected changes in residue behavior, works in anomaly visualization and visual change detection are included below.

Schreck et al [77] recognize the importance of visualization when analyzing features. The authors present an approach utilizing a self-organizing map (SOM) and visualization to help find high quality feature vectors. Feature vectors are descriptors of data characteristics and are important in clustering, classification, and similarity search. Although crucial in many data mining tasks, feature vectors which exhibit high discriminatory ability are often found only after much experimentation, benchmarking, and expert intervention. The authors cluster the feature vectors in the SOM which results in an unsupervised, compact feature space representation. Component planes are constructed by color-coding reference vectors at each SOM position. The

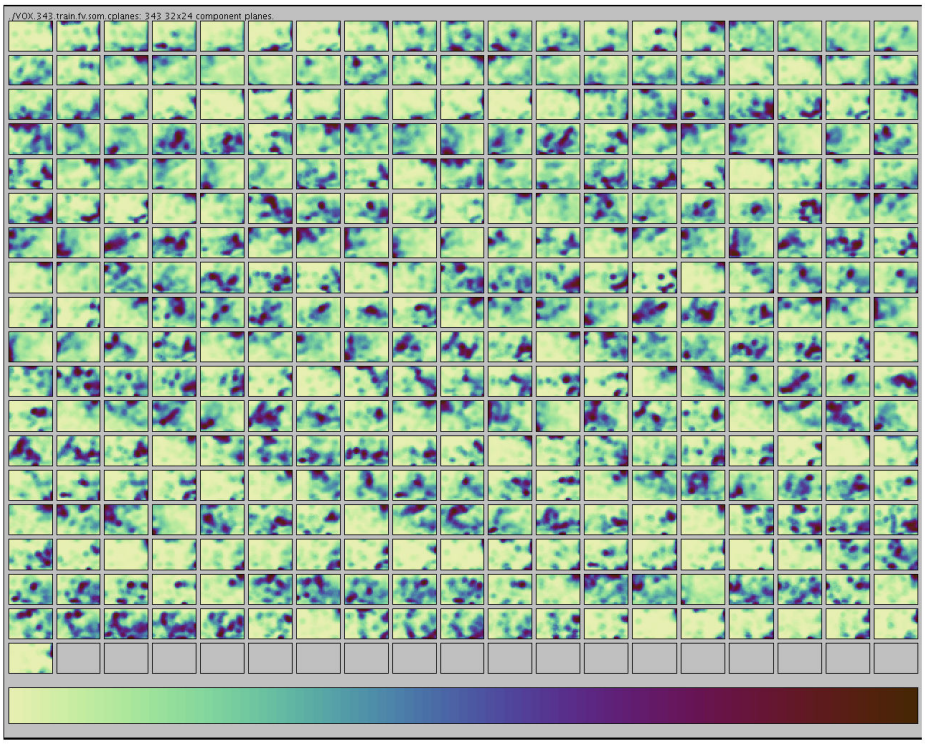


Figure 9: Component plane array for feature vector distribution from a self-organizing map [77].

distribution of the vectors is then visualized. Image processing techniques such as differencing and entropy functions are used to mine the component planes. An array of component planes is shown in Figure 9.

Oelke et al [68] present feature-based text visualization and illustrate how extracted features can provide patterns for desired text characteristics to which other documents can be compared. Characteristics include the importance of passages and the classification of opinions. Visual examination relies on the work in [40] where pixel displays representing documents relate feature importance through a color scale.

Visualizing features produced through image segmentation is one of the main functions of the Semantic Image Browser [91]. The Semantic Image Browser utilizes automatic image analysis to explore image datasets. Users can view the original

image or extracted image features which result from semantic image classification. Layout options include multidimensional scaling and ordering. Viewable low-level features include but are not limited to color histograms, extracted textures, and color variances. Interactions such as zooming, panning, and distortion aid exploration.

The detection of interesting regions or items allows more efficient comparison by allowing experts to isolate where trends or individual values deviate from expectations. Visualization has been shown to be beneficial in finding specific points of interest. The visualization of text passage importance has already been mentioned [68]. Layout generation has been used to reflect importance of numeric summaries and variance through display size and location [29]. Two-dimensional colormaps [95] and the integration of visualization with advanced interfaces [54] have been successfully used for detecting unexpected behavior in financial time series data. Maciejewski et al [51] use visual analytics to identify unexpected behavior change, or hotspots. Hotspots can occur in spatiotemporal data including health reports and terrorism and are available to aid analysts prevent disease spread or criminal attacks.

The work here differs from those mentioned above by providing feature extraction within a highly interactive visual analytics framework for guided discovery in model guided protein examination. Specifically, this approach utilizes image segmentation to detect regions of interest based on the degree of change in flexibility and correlation plots. This work goes even further by providing options for choosing which image characteristics in the protein data should be explored. Finally, this work provides links from the extracted features to the original data so that the features can be understood in context of the entire data set.

# CHAPTER 3: MODELS, DATA, AND TASKS

## 3.1 Protein Construction

The National Center for Biotechnology Information (NCBI) [62] defines bioinformatics as the "science in which biology, computer science, and information technology merge to form a single discipline." The NCBI goes on to state that bioinformatics has two goals. The first is a practical goal of enabling the discovery of new biological insights. The second goal, the progress toward which is much more difficult to measure, is to create "a global perspective from which unifying principles in biology can be discerned." One area of bioinformatics that can greatly benefit from approaches that combine the above components is the study of protein behavior. Complexities associated with protein insight include the simultaneous consideration of multiple protein families, the presence of multiple proteins within families, single proteins having many residues per protein, and a host of environmental conditions.

Proteins are composed of unbranched chains of amino acids connected by chemical bonds [72]. These chains can consist of 20 different possible amino acids and vary greatly in length and sequence across different proteins. The spatial arrangement of these chains determines the biological function of the protein. As Figure 10 illustrates, amino acids make up the primary protein structure and influence the more complex, upper level (secondary, tertiary, and quaternary) shapes. After polymerization, each

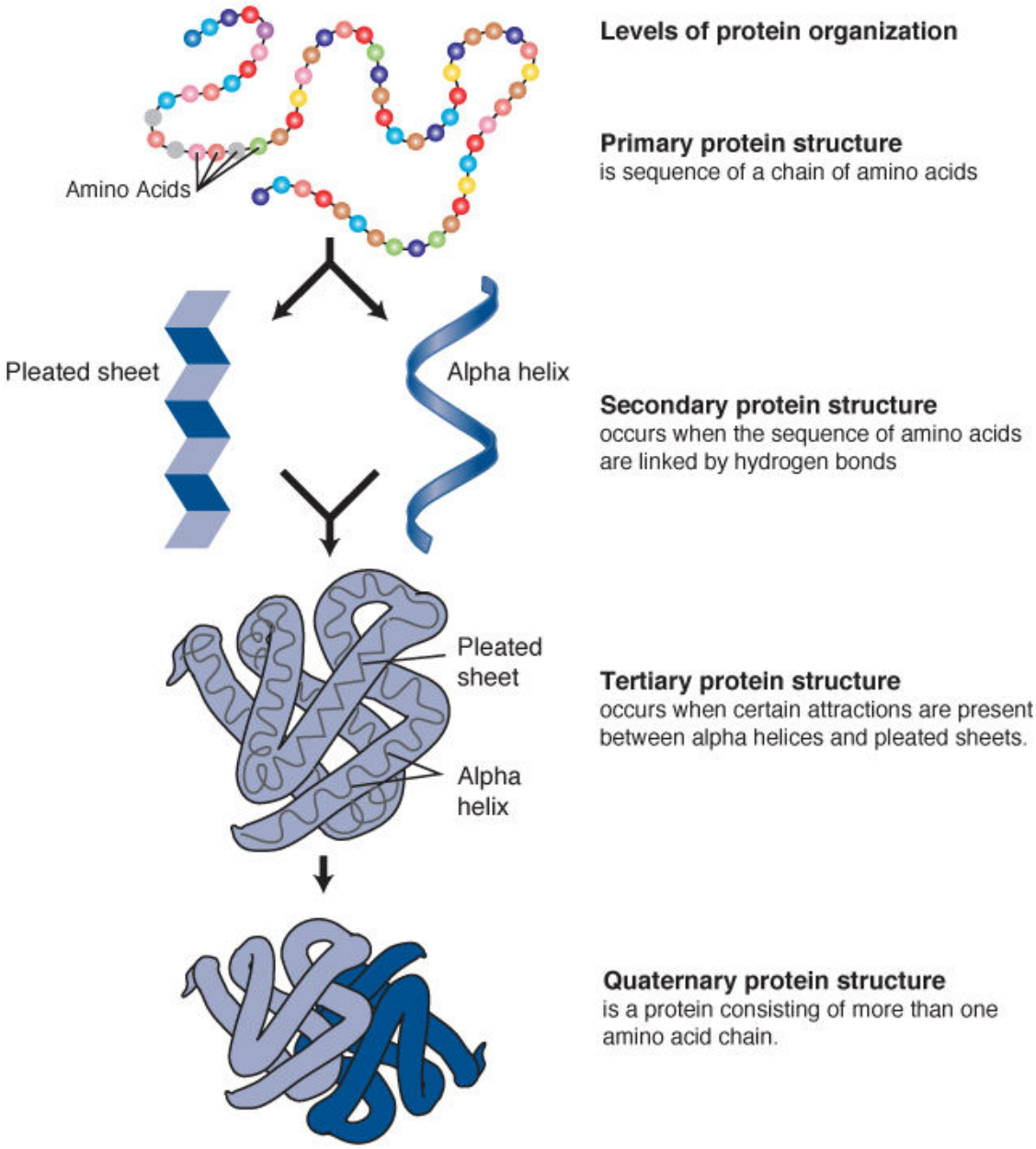



Figure 10: The four protein structures [63]. The primary structure consists of amino acids and influences the upper level structures.

amino acid building block in the protein chain is referred to as a *residue*. The overall 3D shape of the protein chain is defined by internal rotation angles within bonds that form the repeating unit along the protein *backbone*. Present in each residue, the rotatable angles phi and psi (shown in Figure 11) provide the degrees of freedom

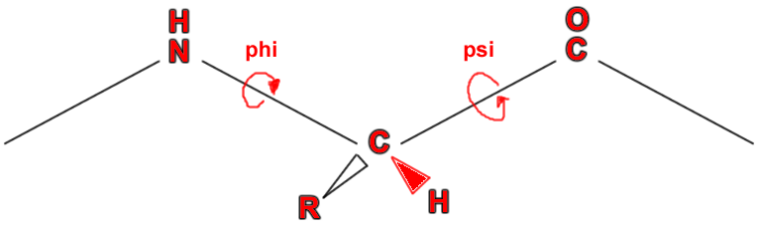



Figure 11: Phi-psi rotation angles that allow movement between chemical bonds in an amino acid [7].

that allow the structure to change. A proper understanding of this process is critical because protein function is defined by its structure, and fluctuations therein. Unfortunately, to date, there have only been a small number of connections that correlated specific structural changes to function [42].

One of the main obstacles in understanding how structural changes in proteins affect function is the vast number of possible spatial configurations allowed by various bond rotations occurring under changed environmental conditions. This set of possible configurations is referred to as a protein's conformation space [66]. Most prediction methods search this space for the structure having the lowest energy, making energy one of the primary variables to be examined. Although important in protein changes, the energy function is often difficult to explore. Exploration is complicated by multiple local minima and an energy function's dependence on many interrelated variables. Furthermore, differences in the underlying chemical structures may require examination of individual proteins.

## 3.2 A Computational Model

Because of the complexities associated with exploring a protein's conformation space researchers rely on models to understand and to predict changes in protein

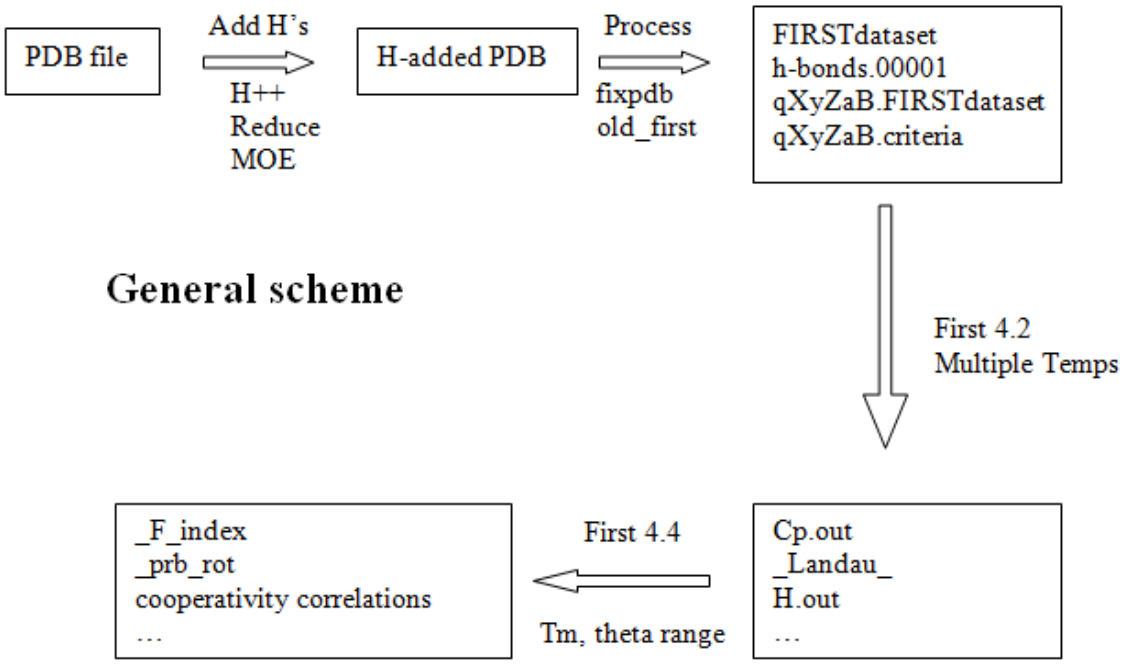



Figure 12: The Distance Constraint Model [34], [48] consists of numerous steps including third-party information such as protein repository files, 3d structure diagrams, several programs developed by domain scientists, manipulation of structures, and model parameters. (From James Mottonen, 2008).

structure. A variety of models exist which are suited to magnify the effect of important characteristics desired for study [57], [36], [28]. The Distance Constraint Model [34], [48] is an example of a model that has proved successful in predicting protein behavior and is the model used throughout this work. The Distance Constraint Model (DCM) is based on free energy decomposition and mechanical constraints. The premise of the DCM is to relate free energy and mechanical constraints with a graph topology which can be calculated in linear time. The strength and location of constraints represented by graph edges in the resulting topology help scientists determine the flexibility/rigidity of a protein. The details of workflow for the DCM are shown in Figure 12. The generalized steps [50] include the following:

1. Each vertex in the graph is assigned degrees of freedom.

2. Molecular interactions are represented by an edge in the graph and assigned enthalpy (depth of the energy curve) and entropy (width of the energy curve).

3. Constraints are added recursively from lowest to highest entropy.

4. When accessible degrees of freedom are present, the added constraint consumes 1 degree of freedom.

5. Enthalpic components are linearly consumed, entropic components are only summed over independent degrees of freedom.

6. The partition function is calculated and thermodynamic properties are determined.

7. Using the probabilities calculated from the thermodynamic data, mechanical properties are appropriately averaged.

Like any model that retains relevance, the DCM [34], [48] is constantly evolving and the need to quickly examine changes in parameters is crucial for efficient model development. Parameters considered by this model include but may not be limited to heat capacities, energy curves, pH measures, temperature, and torsion constraints.

## 3.3 Data Formulation and Tasks

The process of data formulation and the defining of high-level tasks for the model in this work occurred over a period of several years with the help of domain scientists. Data formulation and the definition of tasks were not discrete events but often

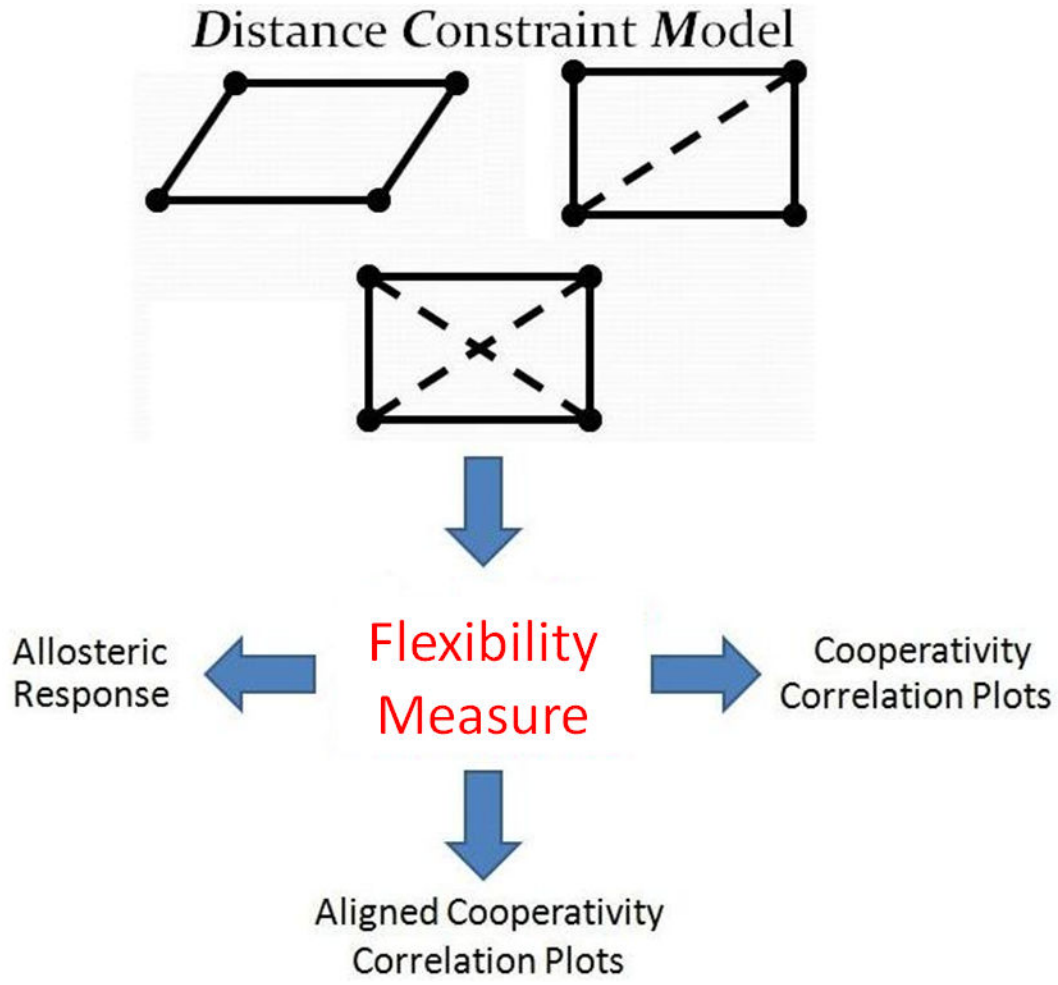



Figure 13: Model outputs can be used to form flexibility measures, psi-phi correlation measures, and aligned correlation measures.

occurred in small segments before, during, and after collaborative activities. The model outputs are described here first so that the high-level tasks, and the process for defining them, can be more clearly explained.

### 3.3.1 Model Outputs

Outputs from the DCM [34], [48] can take several forms depending on the needs of the scientist. They include, but are not limited to, raw flexibility values, correlation measures, and aligned correlation measures (Figure 13). They are now described.

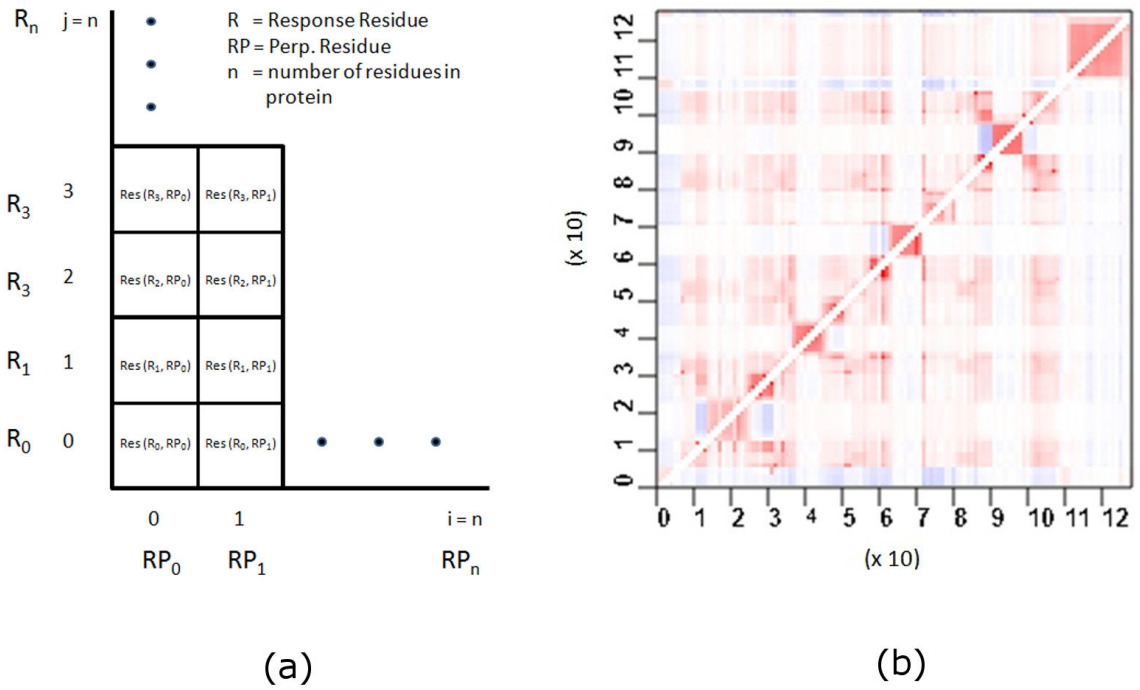



Figure 14: The allosteric response plot for one parameter set applied to the CheY protein [58] where a color index at $i$, $j$ is the response of residue $j$ occurring due to a perturbation at residue $i$. Columns represent the response for all residues when a single residue is perturbed. Rows represent all of the allosteric responses for a single residue as every residue in the protein is perturbed.

**Raw flexibility measures.** Raw flexibilty measures are determined by model parameters or other internal modifications and serve as a standard method for comparing residue flexibility. An example use of raw flexibility measures is the study of allosteric response. Allosteric response tracks how a protein change in one residue, referred to as a *perturbation*, affects the flexibility of other residues. Analyzing allosteric responses for many residue parameters and types will allow scientists to better understand how to achieve specific biomedical results. Allosteric responses can be represented by an $n$ $x$ $n$ asymmetric color plot where $n$ represents the number of residues for a given protein. Figure 14(a) shows the set-up for allosteric response plots and (b) shows the allosteric response of one parameter set when applied to the CheY protein

[58]. In each plot, the residues are ordered according to the three-dimensional protein structure so that local and regional characteristics have biological meaning. In the examples used here, a color index at *i, j* is the flexibility change of residue *j* occurring with a perturbation at residue *i*. Color values are used such that darker shades of blue indicate increased residue rigidity and darker shades of red indicate increased flexibility. White areas correspond to neutral residues.

**Correlation measures.** Scientists may be just as interested in how flexibility changes are coordinated as parameter sets vary just as much as a global flexibility measure. For example, Quantitative Stability/Flexibility Relationships (QSFR) describe a high dimensional range of model properties where regions of correlated flexibility and rigidity are of great interest [49], [59]. Domain scientists currently visualize correlated flexibility and rigidity through cooperativity correlation matrices (Figure 15) where each axis is also ordered by the sequence of amino acids (residues) that define a protein. Individual indices are correlations for one of two different types of rotation angles found on the residue structure such that for two residues $A$ and $B$

- i, j is the correlation between $A_{phi}$ and $B_{phi}$
- i + 1, j is the correlation for $A_{phi}$ and $B_{psi}$
- i, j + 1 is the correlation between $A_{psi}$ and $B_{phi}$
- i + 1, j + 1 is the correlation between $A_{psi}$ and $B_{psi}$

Correlations between pairs of residues provide insight into the nature of the dynamics of a protein. Patterns that emerge in these plots over families of proteins give

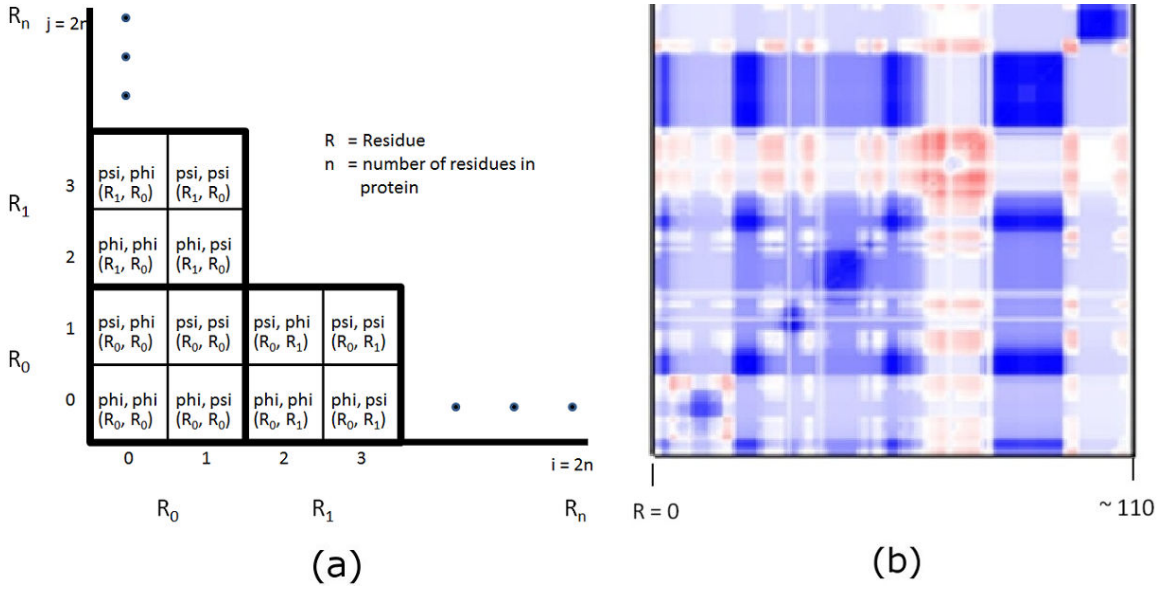



Figure 15: The length of each axis is twice the number of the residues in the protein. Each index in the matrix is one of the pair-wise correlation measures for a residue's psi-phi angles. In this cooperativity correlation matrix, blue indicates residue pairs that are co-rigid, red indicates coflexible, and white indicates no mechanical coupling. The three dimensional protein structures are colored according to a single strip of the matrix, meaning each highlights all pairwise coupling to a given reference residue. Some residues share similar correlations resulting in large, consistent regions. Other residue regions that differ from neighbors are much more difficult to detect. Three dimensional structures were visualized with PyMol [15].

insight into the mechanisms important for biological function. These include the size and location of similar or dissimilar regions, and any outliers where a given residue may unexpectedly differ from its neighbors, or from a consensus over the family. Descriptors, or metrics, include (but are not limited to) the probability of residues to rotate (PROB), the probability of residues to be in correlated motion (COR), a flexibility index (FLX), indicators of structure fluctuation (SUS), and measures of mechanical freedom (DOF). Taken together, these plots simultaneously provide both local and global descriptions of protein dynamics. Scientists can plot the correlation between any two metrics but this work only considers the correlation among mea-

sures of the same metric type (e.g. PROB-PROB, COR-COR, SUS-SUS, FLX-FLX, and DOF-DOF). The volume of plots produced and the number of residues present within a plot makes large-scale comparisons and exploration cumbersome and ineffective. Although some descriptions for a single protein are expected to show some common characteristics, any unexpected differences within a protein's set of metrics or across proteins for a single metric would be of great interest to domain experts.

**Aligned correlation measures**. Not all correlation measures being studied have the same number of residues because the underlying proteins are different lengths. This often occurs for individual proteins within the same family and makes comparison among those proteins more difficult. To better compare proteins of different lengths, alignment algorithms [64], [80] are applied. The sequence alignment is represented by a string of alpha characters (each character corresponding to a residue) and, in simple cases, alignment algorithms will shift the residues left or right. During the process of shifting, spaces or gaps may be inserted so that the optimal alignment is reached (Figure 16). (What constitutes "optimal" and how that condition is reached is an active area of research [70], [16] beyond the scope of this work.) Accounting for aligned residues when applying both automatic analysis and visual techniques is necessary for accurate comparison.

### 3.3.2 High-level Tasks

This work addresses the problem of identifying and exploring points of interest in allosteric response or QSFR [49], [59] correlation data resulting from the DCM [34], [48]. The process of gathering user requirements for solving those problems

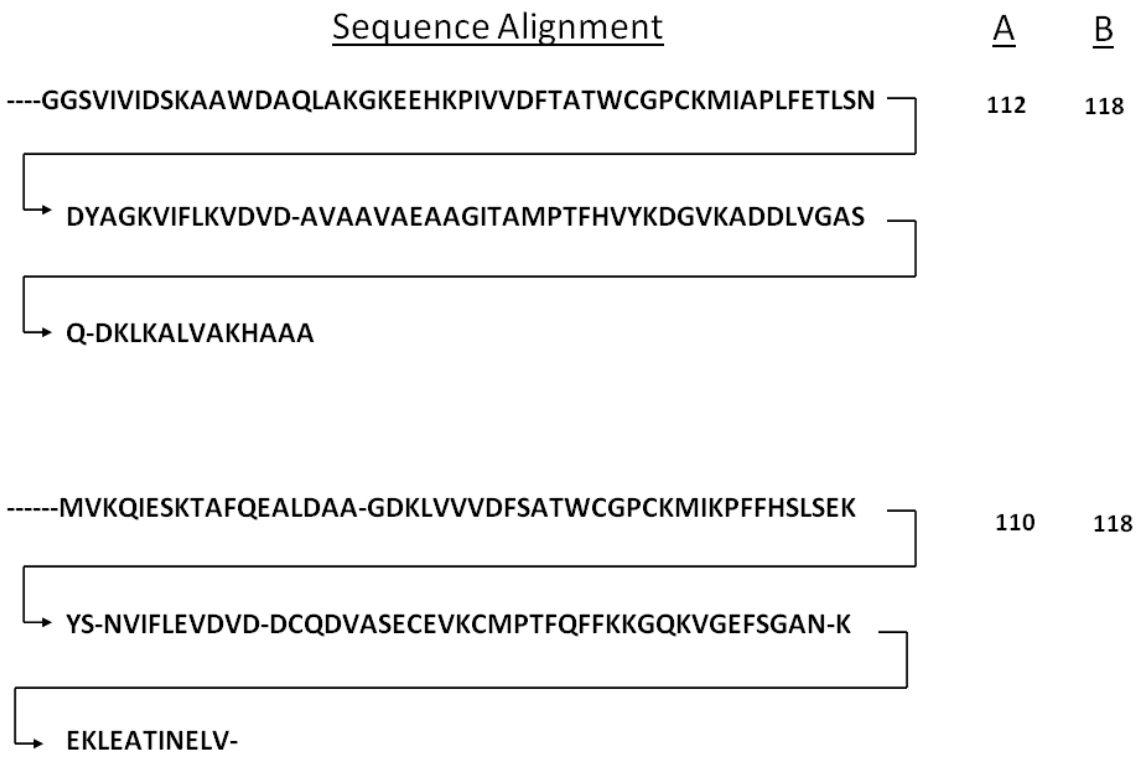



Figure 16: Alignment for two proteins. Dashes represent where the residues have been shifted to produce the best alignment. Column A shows the length of the protein and column B shows the adjusted length used during comparison.

included heavy involvement by domain scientists beginning in the summer of 2008 and is most adequately described as participatory design [60]. Domain experts had no choice but to be full partners in the design process because I knew very little about the correlation and flexibility data that was driving the collaboration. I began attending group research meetings with bioinformatics faculty members, graduate students, and a research scientist where the group would examine a single plot or, at best, a few plots at a time. I was a silent observer in the beginning and was, frankly, overwhelmed by the complexity and diversity of the workflow to produce even a single flexibility plot. The domain terminology describing the workflow and the resulting data were just as complex and diverse. I began experimenting with ideas, maintained

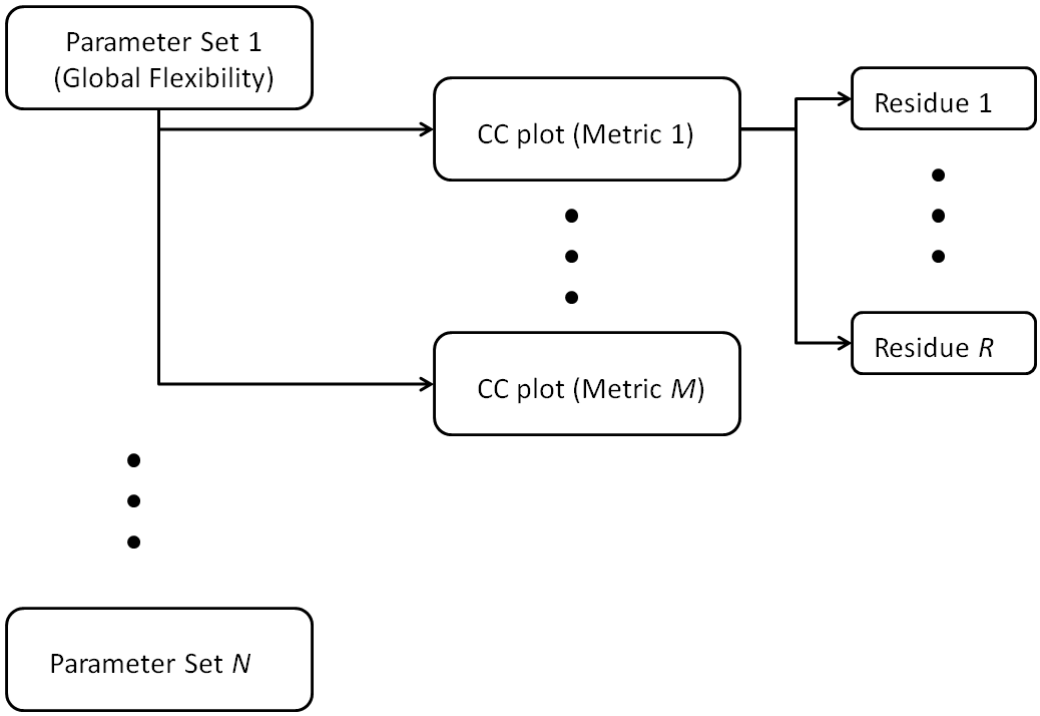



Figure 17: Residue comparison becomes difficult as the number of parameter sets increases. Each parameter set produces a set of flexibility which, in turn, can be used to calculate several types of correlation measures among many residues.

frequent communication (at least once every two weeks) with both bioinformatics and visualization experts, and regularly presented results. Discussion was primarily at meetings during summer sessions (of which there were well over 30), through email, and less frequent meetings during the academic year.

I quickly learned plots displaying raw flexibility measures for a single protein or parameter set can contain local and regional characteristics for perhaps several hundred residues that are difficult to identify and even more difficult to compare. Eventually, flexibility plots can be used to construct cooperativity correlation matrices. Acquiring insights from these matrices becomes more difficult as the number of parameters, correlation types, or residues increases. Figure 17 illustrates the multi-tiered prob-

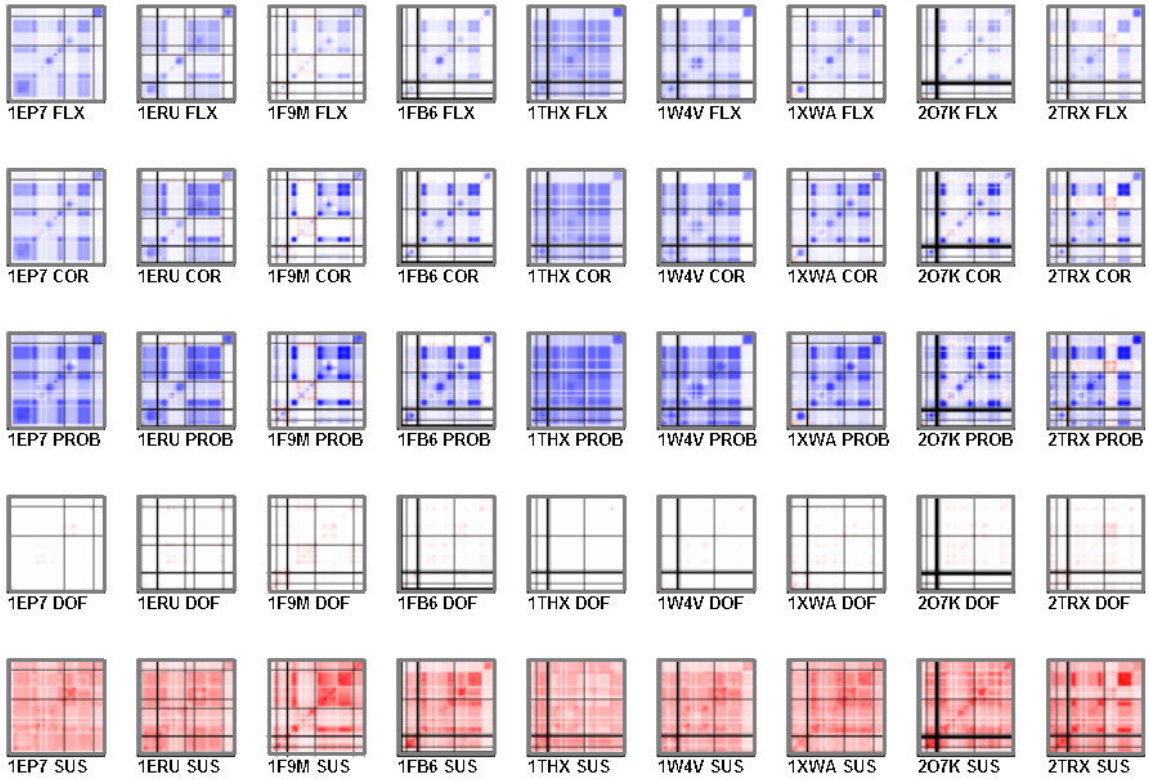
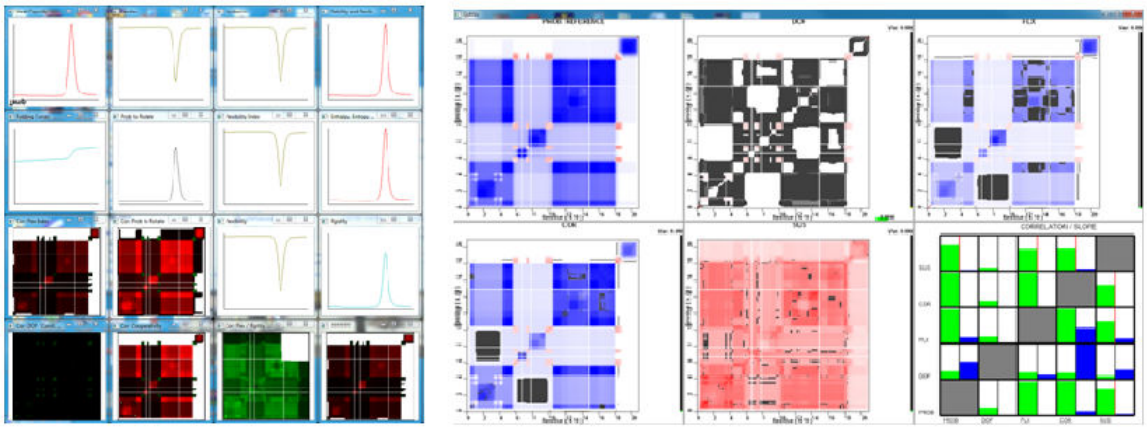



Figure 18: Multiple correlation metrics for the 2TRX protein family. In this display created by WaveMap [3] each row consists of a single correlation metric for all proteins in the family. From top to bottom the metrics are FLX-FLX, COR-COR, PROB-PROB, DOF-DOF, and SUS-SUS. Black lines represent gaps inserted into the sequence by an alignment algorithm.

lems challenging users of the DCM and QSFR data. Domain scientists have few tools available for examining plots and then pruning the possible choices to only those of interest. The lack of tools hinders biological insight.

There were several failed attempts before the solution presented here emerged. Those first attempts primarily targeted a small, but much studied data set comprised of QSFR [49], [59] correlation plots for nine related proteins (Figure 18). The first attempt consisted of multiple glyphs similar to a picture frame. The correlation type of interest was in the center of the frame and the remaining four measures for the protein made up the surrounding frame segments. This proved ineffective when



(a) (b)

Figure 19: Residue comparison becomes difficult as the number of parameter sets increases. Each parameter set produces a set of flexibility which, in turn, can be used to calculate several types of correlation measures among many residues.

multiple proteins were viewed because the domain analysts found the color mapping and glyph representation confusing. The second attempt plotted the cooperativity correlation plots on the bottom of the screen and line graphs for parameters relevant to QSFR such as energy curves and heat capacities on the top. A prototype is shown in Figure 19(a). The third attempt (Figure 19(b)) utilized animation to selectively gray items that fell below a correlation threshold for a single protein across the QSFR correlation types listed above. The aim in this case was to quickly identify regions that had high positive or negative correlation to a metric of interest chosen by the user. Different colors were eventually used to signify predefined bins of correlation. Both attempts proved ineffective for large-scale analysis since the number of plots were limited to five (one for each metric) so that neither the back-end parameters nor the shaded areas could be easily compared.

Most of the difficulties in the previous attempts were unsuccessful because they were not scalable to large data sets with many dimensions. Additionally, the mixing


Figure 20: The CheY data set [58] consists of 75 plots representing one protein's behavior for varying combinations of three parameters. Many of the differences among the plots are subtle and difficult to detect.

of correlation types often confused development and analysis efforts. Further complicating development was the fact that this data set contained several different proteins having residue sequences needing functions to adequately handle alignment results. The CheY data set [58] that had been developed by BMPG for studying allosteric response was much more suitable (Figure 20). It was significantly larger, all plots could be considered at once without confusion, and there was no significant sequence preprocessing required. Even though the data set was changed to ease development, the same high-tasks described below are the same for almost all flexibility data sets

used by BMPG.

**Analyzing spatial relationships and numeric trends of flexibility measures within proteins.** Protein dynamics can be altered by either a local group of residues, larger regional groups, or the concerted effort of multiple areas of varying sizes. Locating and identifying those regions of interest which contribute to change is necessary before the roles of individual subunits can be identified.

**Studying parameter influence and grouping parameter sets.** Parameter refinement within a model is a reflection of evolving expert knowledge for a specific protein and environmental condition. For a fixed parameter set, a comparative analysis between different proteins and/or environmental conditions can help discover new spatial relationships and numerical trends. Grouping model outputs by parameter sets will allow scientists to understand what combination of parameter settings result in the greatest or most unexpected change for a single or group of residues. From this knowledge, domain experts can refine the model or investigate ways to take advantage of these differences.

**Pruning parameter sets and residues.** Clearly defining relationships among residues and parameters of interest is best accomplished if redundant or uninteresting data items are excluded from consideration. This can take the form of excluding entire parameter sets, entire proteins, or individual residues based on domain knowledge or thresholding. Additionally, scientists need to be able to start with a well-studied individual residue, group of residues, or overall structure that is accurately reflected by model outputs and then eliminate parameter sets based on similarity (or dissimilarity) from the established item.

BMPG members confirm that these tasks are frequently encountered. Previous to this work, the lack of a tool in meeting them was a significant obstacle in securing biomedical advances. Because of the complexities associated with protein insight, approaches that address the above goals must combine the best of automatic techniques to guide users to interesting places in the data, the natural ability of humans to discern patterns, and the unique knowledge of domain scientitists.

# CHAPTER 4: WAVEMAP - INTERACTIVE FEATURE DISCOVERY

WaveMap [3] is a visual analytics prototype that integrates wavelet lifting [82], [37] with visualization to address the problems associated with protein analysis. Specifically, WaveMap was designed to help scientists find global, regional, and individual residue characteristics that may be of interest. The prototype is now presented.

The system is comprised of inputs, preprocessing, and the interface (Figure 21). Inputs include a file containing the sequence alignment for all proteins to be studied, one file for each raw flexibility/correlation data matrix, and a file for the parameter settings. Parameter settings include which part of the wavelet output is sent to the input of the next iteration, the number of proteins, the number of necessary wavelet

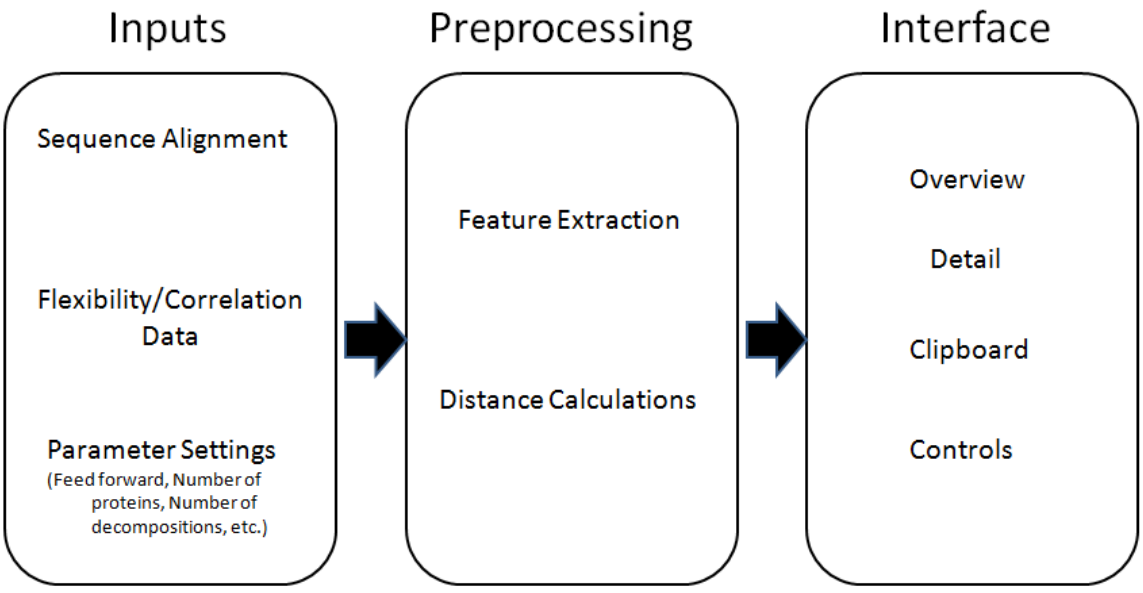



Figure 21: WaveMap consists of inputs, preprocessing, and the interface.

decompositions, and several other parameters used in system functions. Preprocessing includes extracting features (described below) and distance calculations. The interface consists of a control panel, an overview to display the entire data set, a feature window to examine selected plots, a detailed analysis window to perform closer examination, and a clipboard to carry plots of interest throughout analysis (Figure 22). The interface with allosteric response data is shown in Figure 22(a)-(d). Co-

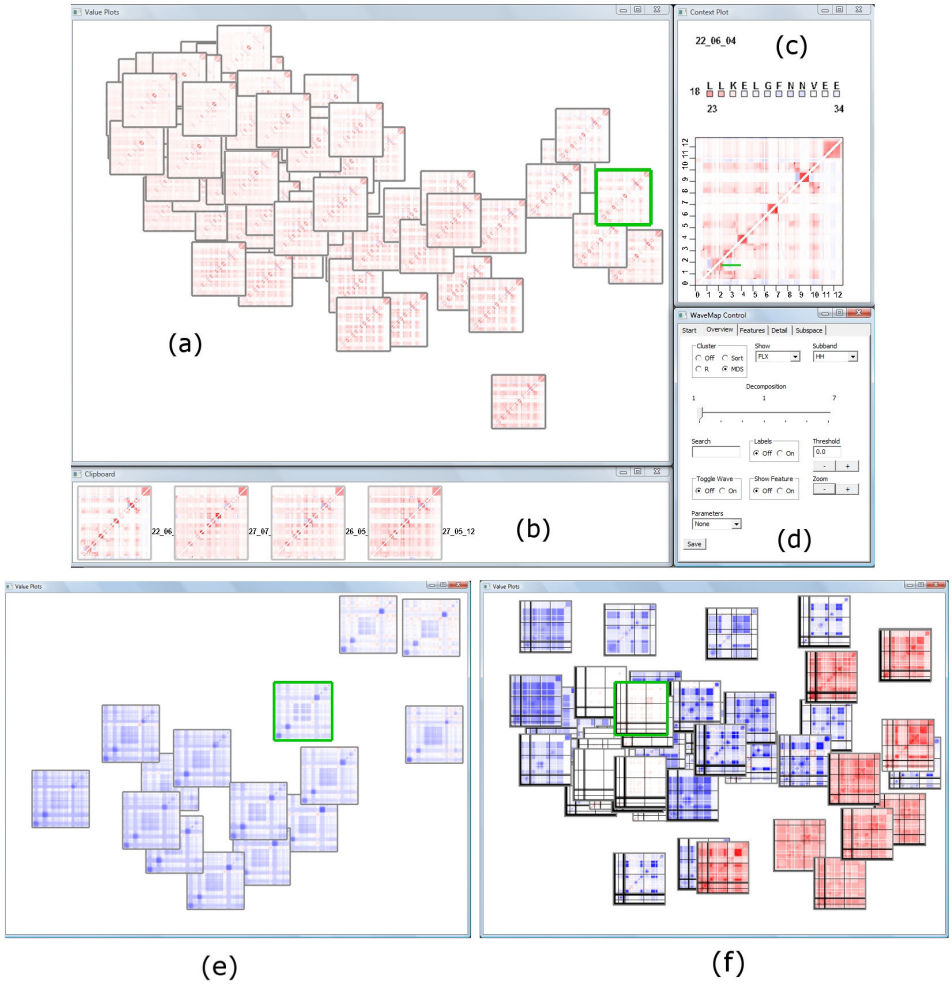



Figure 22: (a) MDS, Sorting, and Jigsaw layout space. (b) Clipboard. (c) Detail window. (d) Control panel. (e) Single cooperativity correlation measure. (f) Multiple correlation metrics after alignment. Black lines represent "gaps" inserted by an alignment algorithm. When multiple metrics are included, users can filter the display by metric type.

operativity correlation measures are shown in Figure 22(e) and aligned cooperativity correlation measures with multiple metrics are shown in Figure 22(f). In the case of aligned measures and multiple metrics, users can view the entire data set with all metrics available or filter the data set to one metric through a drop-box selection.

WaveMap [3] allows scientists to begin with the entire data set and continuously refine their analysis to individual residues. The workflow is shown in Figure 23. The first step in this process is to extract features by wavelet analysis. Extracted features are visually presented to users to help them locate global trends or local areas where trends are interrupted. Subtle trends that are hard to discover in the original data become visible in the feature space. To study parameter influence, a set of plots in the original data or extracted features can be viewed in a clustering or sorting layout from which global trends across parameter sets, as well as clusters and outliers of

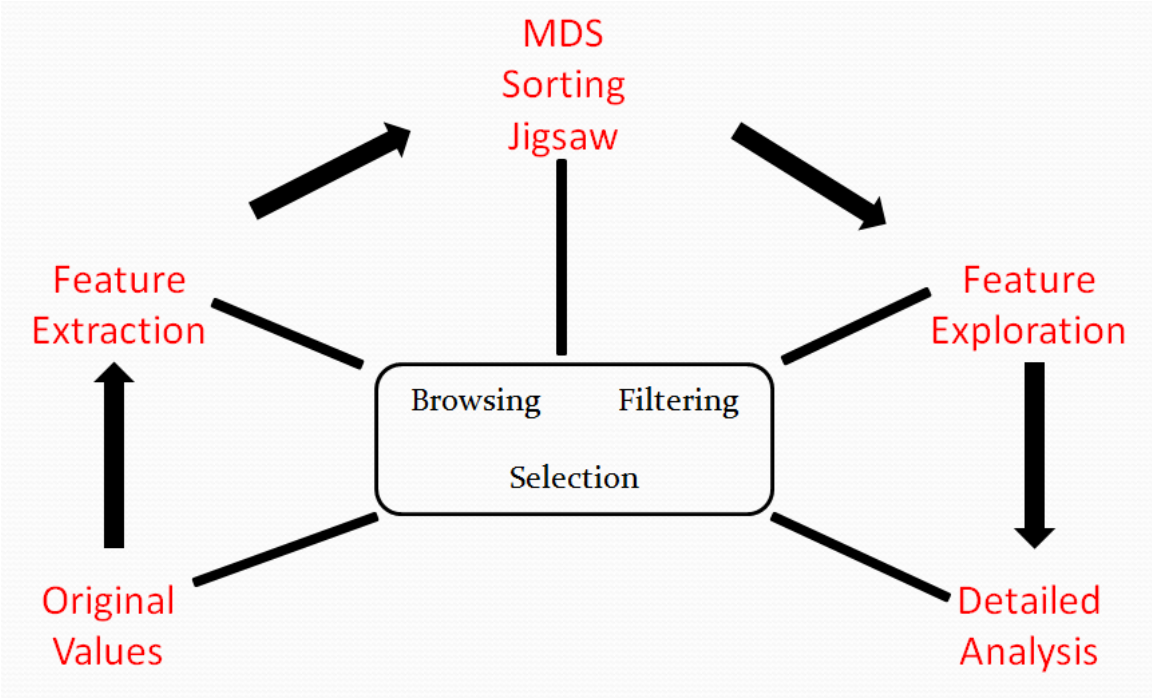



Figure 23: Feature-based framework for examining protein flexibility matrices.

plots can be observed. Users can interactively retrieve groups of interesting plots so that further examination and comparison can reveal the relationships among residues and parameters. Features can be filtered based on their type and magnitude and are intuitively mapped to the original data. The feature window allows examination of interesting regions for a subset of interesting plots. The detail window facilitates coordinated, residue-level analysis among multiple plots. In every view, specific regions for given parameter sets can be exported for insight management and exchange. The framework is now discussed.

## 4.1 System Components

### 4.1.1 Feature Extraction

Wavelet lifting [82], [37] is first applied to extract varying plot features at many resolutions. During each application to a discrete, two dimensional signal (or decomposition), the data is separated into the high and low frequency components. The result is a series of four data matrices each of which is one-quarter the size of the original data matrix. The results include the high frequency components in both directions (HH), low frequency components in both directions (LL), low frequency along rows and high frequency along columns (LH), and high frequency along the rows and low frequency along columns (HL). One of the components is chosen to be fed to the input of the next stage and the process repeats.

Wavelets come in varying families and can be designed to extract desired features [83]. A lifting implementation of the widely-applied Debauchies 4 wavelet [37] is the starting point chosen here but other wavelets can be used. Because the data is

halved along the rows and columns during each application, the original data set is linearly interpolated so that each row and column is a power of two. (It should be noted that the suggestion given by domain analysts for handling sequences with gaps was followed. Their suggestion was to remove any row or column across the data set if one protein or parameter set had a gap inserted.) The interpolated data is only used in application of the wavelet algorithm and is never visible to the user. After each decomposition during wavelet analysis, any given feature represents a larger neighborhood in the original data. Feature magnitude can be mapped to color so that the degree of change between adjacent locations can be visually represented. The components, or subbands, and the characteristics emphasized in the data that are important for our work are illustrated in Figure 24 and include

- **LL:** Averages along rows and columns
- **LH:** Averages along rows and differences along columns
- **HL:** Differences along rows and averages along columns
- **HH:** Differences along rows and columns

It may seem that the total amount of data has been significantly increased because each original data plot is now represented by four different subbands. However, each decomposition results in each subband being only one-quarter of the input data size. Additionally, the subbands and multiple levels of resolution produced are different perspectives of the original data. This allows experts to choose the appropriate prism through which domain knowledge can be applied.

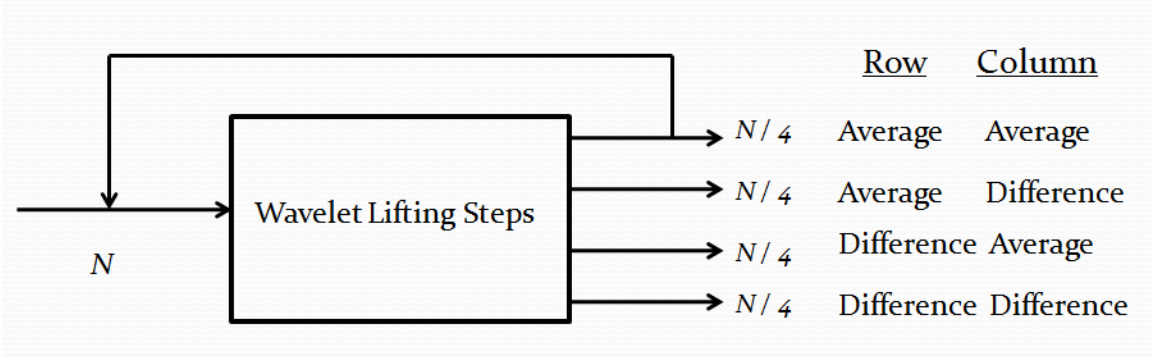



Figure 24: During a wavelet iteration four data matrices are produced, each one-fourth the size of the input matrix. Each resulting matrix contains features representing different data characteristics.

Figure 25 illustrates how features for each subband emphasize various characteristics present in cooperativity correlation flexibility plots for a TRX protein [59]. Each index is the correlation between two flexibility measures for two rotation angles. The top row is the original data and the bottom row is the transformed data after wavelet analysis and filtering. Original plot values (top row) are displayed with the red-white-blue scheme currently in use by domain scientists. The features for all subbands except for the averaging features (LL), use a different color scheme because features capture the change occurring between plot regions and not the original plot values. We chose a feature color scheme ranging from green (negative changes) to white (no change or below a threshold) to orange (positive changes). The red-white-blue scheme was kept for the averaging features because they visually relate original data information at varying resolutions (25(c)). In Figure 25(a) coarse-grain behavior along rows is preserved and differences along each column are detected so that general residue behavior is preserved along the rows but changes in residue behavior between adjacent column are detected. Figure 25(b) shows the detection of changes

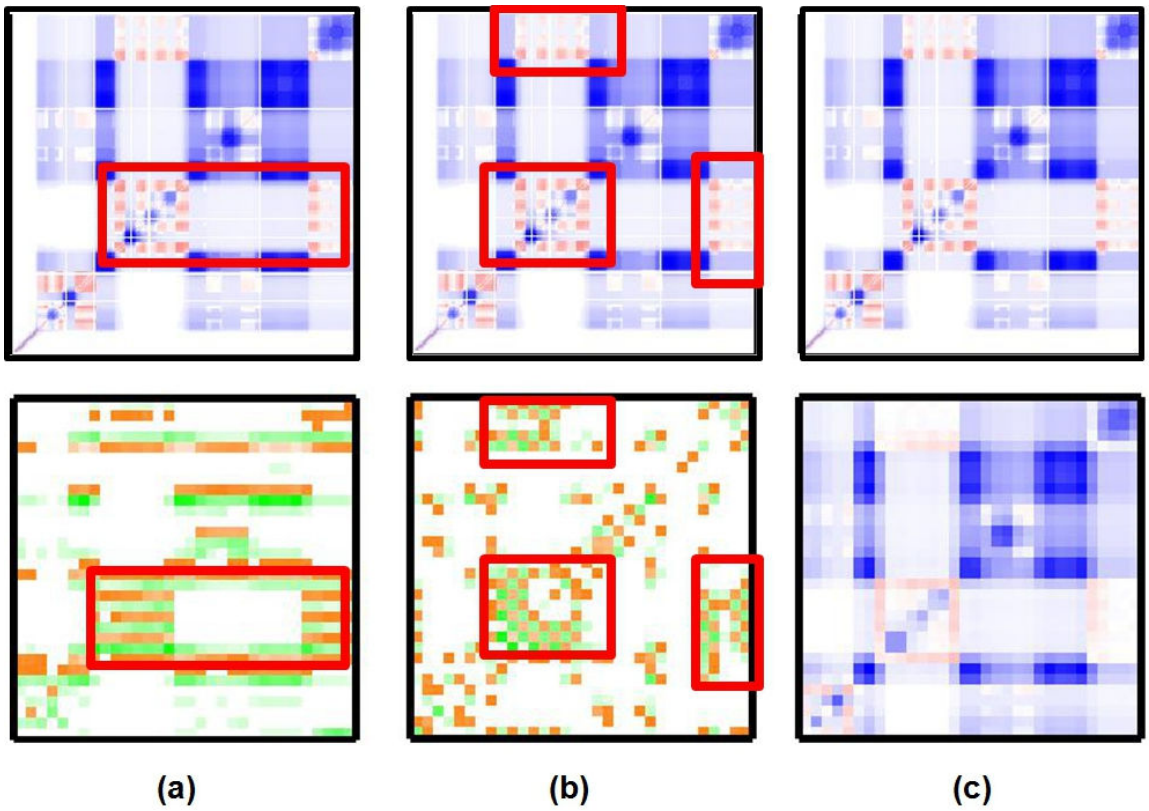



Figure 25: Three cooperativity correlation plots [49], [59] depicting the correlation of flexibility changes. Original correlation values are shown on the top row. Blue indicates co-rigid regions and red indicates co-flexible regions. The data after wavelet transformation (3 decompositions) are on the bottom row and corresponding regions are bounded in red. Orange highlights positive correlation changes and green highlights negative correlation changes in all except the last pair. (a) Row patterns are preserved while indicating changes along columns. Because this data set is symmetric, the LH and HL subbands are redundant. Only the LH subband is shown here. (b) Areas of change along both rows and columns are detected. (c) Coarse-grain characteristics are preserved for the entire data set.

along both rows and columns revealing where residue behavior changes from adjacent residues in both the row and column direction. Figure 25(c) preserves coarse-grain behavior in both directions. Although the features highlighted in the top row are easily detected and represent symmetric data, such features can be much harder to detect in other datasets without the help of the transformed data. An application to the asymmetric and much more subtle allosteric response data is discussed later.

### 4.1.2 Overview

After feature extraction, either the original values or the features can be displayed in the overview. Original data plots and extracted feature plots can be browsed with a MDS layout (Figure 26), a jigsaw layout (Figure 27), or according to similarity based sorting. MDS allows viewers to interpret data similarity as visual distances [9], [14]. Plot size can be interactively reduced to minimize overlap. The jigsaw layout [89], [92] allows users to examine plot clusters and outliers without overlap. It is a grid layout where similar plots are placed close to each other and boundaries among clusters of similar plots can be detected. Sorting enables users to examine similarity in relation to a selected plot. MDS, jigsaw, and sorting configurations depend on whether the original or feature plots are chosen. If features are explored, the MDS, jigsaw, and sorting layouts can futhermore depend on the subband and level of decomposition chosen. Providing access to all subbands and decomposition levels allows users to view the plot arrangement or the features which detect desired characteristics and simultaneously reducing the number of points being considered. Users have much flexibility when setting display properties. The data can be displayed according to features but display the original plot values. Likewise, the data can be viewed according to the original values but display features.

When parameter set labels are turned on, analysts can see if a relationship exists between any parameter sets and if any parameter combinations result in outliers. WaveMap [3] has the ability to read a set of filenames with the format ParameterTypeA-ParameterTypeB-ParameterTypeC (format of the allostery data

Figure 26: Clustering features for a section of the MDS layout. The HH subband after 2 decompositions identifies places of change along both rows and columns while simultaneously reducing the number of data points. the data is separated into areas with plots having many points of change (left side of each layout) and plots with fewer points of change (right side of each layout).

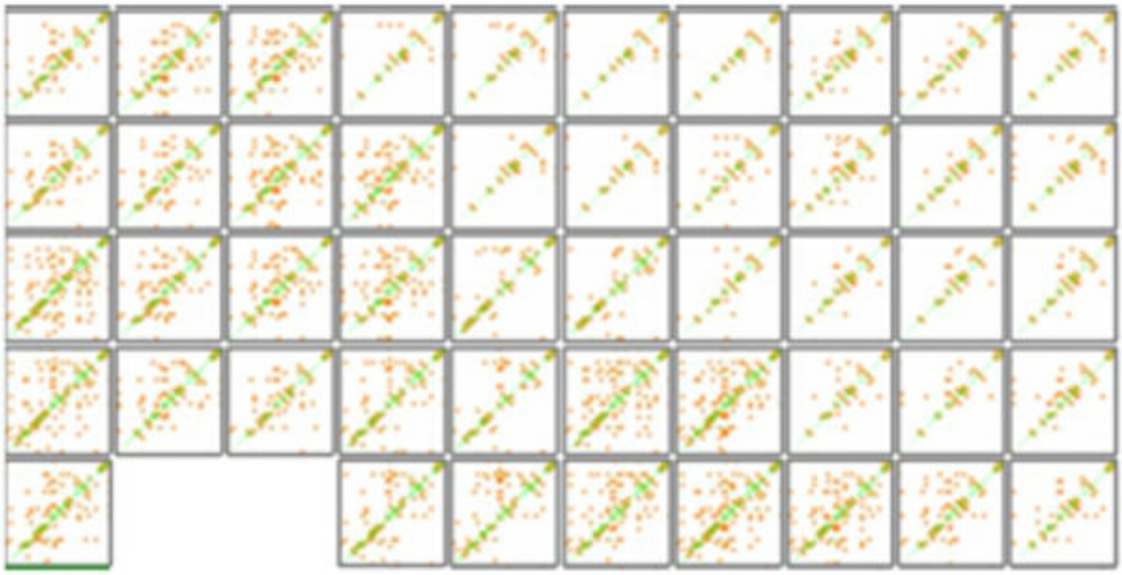


Figure 27: Clustering features for a section of the jigsaw layout also showing the HH subband after 2 decompositions.

file names)and then dynamically count the number of variants per type, up to 16 variants per type. The user can then choose through the interface which parameter type to apply border highlights. Figure 28 illustrates this functionality. Figure 28(a)

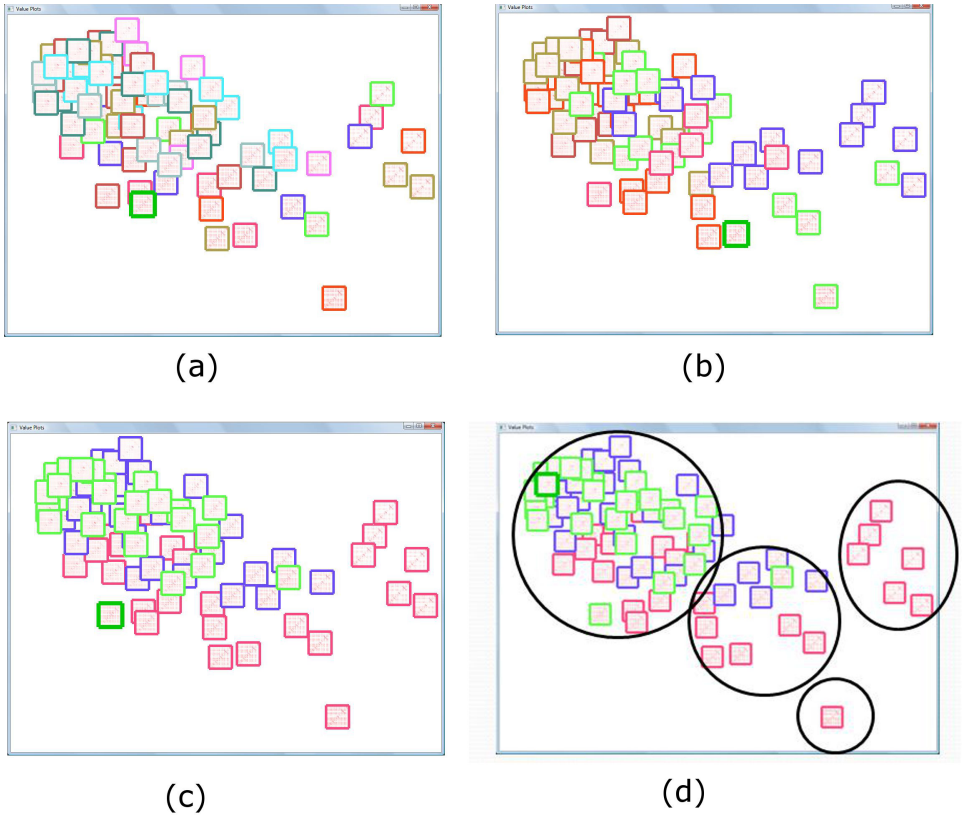



Figure 28: Parameter variants in (a) and (b) are evenly distributed and mixed. Parameter variants in (c) generally cluster by type. Clusters are identified in (d) and show varying degrees of consistency.

shows that variants of the first parameter type are evenly mixed. The variants of the second parameter type in 28(b) are mixed but seem to exhibit more consistency than 28(a). Variants of third parameter type shown in Figure 28(c) show even more consistency. In the third parameter type, each color represents a variant of an entropy parameter referred to as *d-nat*. The three possible values are 0.4 (red), 0.8 (blue), and 1.2 (green). Figure 28(d) indicates that some of the 0.4 items cluster in a small homogeneous cluster (far right), a less homogeneous cluster with mostly one other color (center), and a very heterogeneous cluster (far left). Domain analysts reported that the configuration confirmed their suspicions that the data set would at least

roughly cluster by this variable.

Many interactions exist for the overview display. They include filtering, decomposition level and subband change, searching, selection, and drawing.

**Filtering.** Features within plots can be filtered according to a user-defined threshold changed by buttons that trigger incremental increases or decreases or by direct entry into a textbox. Thresholding allows only the most important features to be shown by eliminating plot items that have an absolute value less than the user-defined value. Entire plots can also be filtered by metric type through a drop-down box.

**Decomposition Change.** The decomposition level is changed through a slider giving smooth transition between various levels of resolution. Decomposition values are propagated to feature exploration to aid in view continuity.

**Subband Change.** The feature type displayed can be changed through a drop-down box.

**Searching.** Individual plots can be located by entering the protein-metric combination. Found plots are highlighted in bright green.

**Selection.** Any plot left-clicked in the main display will be shown with greater detail in the context window. A middle-mouse click on a plot will add the plot to the clipboard. The clipboard is used to maintain an evolving list of interesting plots to be investigated further.

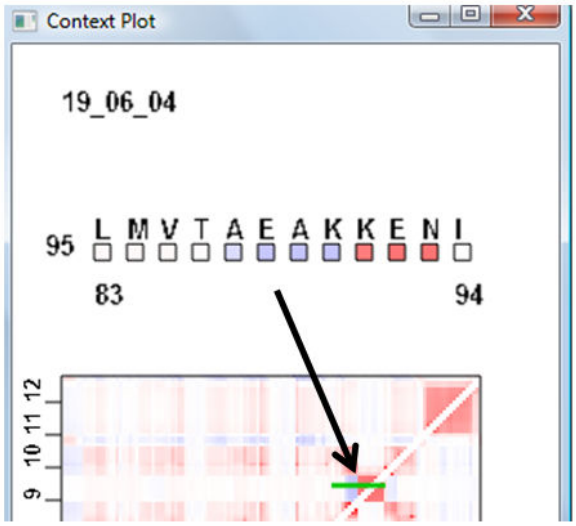



Figure 29: Drawing a line on the plot in the detail window reveals the residues at the line location along with the plot coordinates and residue name abbreviations.

**Drawing.** Detailed residue information is always available. Drawing a horizontal line on the plot in the context window displays corresponding residue flexibility/correlation colors and residue name abbreviations (Figure 29).

### 4.1.3 Feature Exploration

After viewing and selection, features can be examined in detail. Users can either view all proteins for a single metric or a mixture of protein-metric combinations that have been placed on the clipboard. A coordinated lens Figure 30(a) allows users to view the averages (LL) at the current decomposition level as the user sweeps over the features. The LL subband was chosen to be the center of the lens so that as sweeps are performed, users can associate features with the original values at the given level of decomposition.

Reconstruction (Figure 30(b)) further extends the effective and efficient association of extracted features with the original values. In this system, users are allowed to

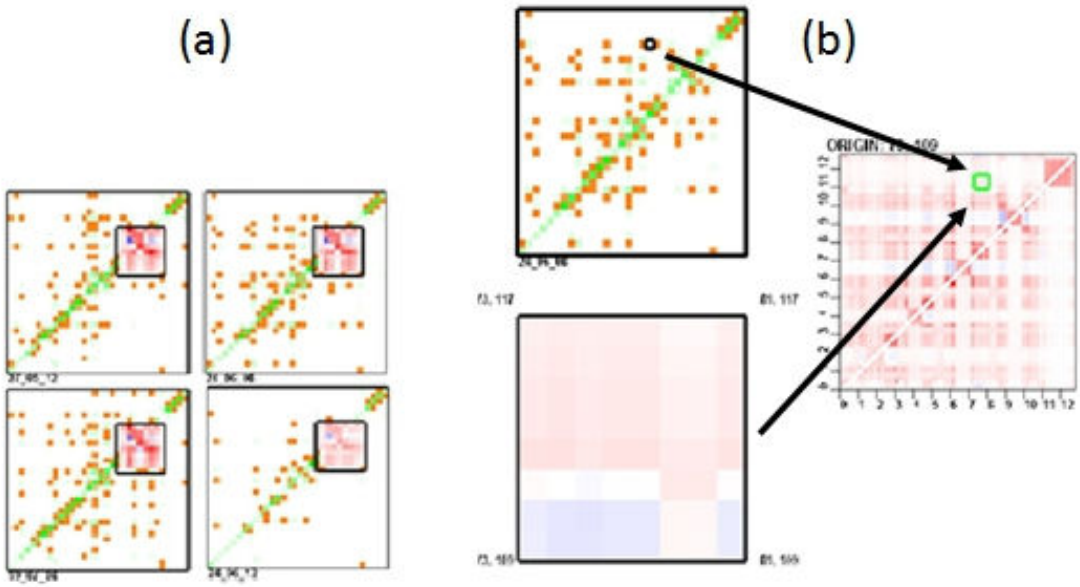



Figure 30: (a) A coordinated lens allows simultaneous examination of multiple plots while relating features to the original data. (b) Reconstruction further bridges the feature and original data. Coordinates are marked in the large plot showing the original data and in the context window. A bounding box marks where the features occur in the data before transformation. Coordinates are propagated to the detail view.

select a plot and then manually inspect features. Navigation is through directional buttons and users can choose to visit each feature or snap to features with values above the threshold. Once a feature is accessed, the original data values responsible for that feature value are bounded in the context window and in a resizable plot in the main display. Bounding box size increases along with the number of decompositions reflecting a decrease in the number of features present but an increase in neighborhood size. Boundary conditions can be problematic in wavelet analysis and developing techniques to appropriately deal with this case is an active area of research beyond the scope of this work [11], [27]. However, an initial step towards informing the user of boundary effects has been incorporated into this system by filling the reconstructed

bounding box that falls outside of the plot with gray.

Once a feature is visited, plot coordinates of the bounding box are displayed in the main display and in the context window. Main display coordinates represent absolute coordinates for that particular plot before any gaps are inserted during alignment. Coordinates in the context window reflect the bounding box origin relative to gaps inserted by alignment algorithms. Gaps are included in the context window bounding box so that plot coordinates for the entire data set can be normalized. The normalized coordinates are propagated to the next view for detailed analysis across multiple proteins.

### 4.1.4 Detailed Analysis

Once a subset of protein plots are placed on the clipboard, the chosen plots are available for detailed analysis (Figure 31). In this view, a column segment is shown for each protein. For raw flexibility values, each column is one rectangle wide. For correlation measures, each segment is composed of two horizontally adjacent color rectangles that represent the correlation values of the two rotation angle pairs for each residue in the column. Any inserted gaps appear black. The current column and row numbers are shown to the left of the series of correlation cells. Exploration in this view can begin at the plot origin or from the context coordinates propagated from the previous view. After a suitable beginning point is found, residue columns are navigated by directional buttons or by entering known coordinates into text fields so that a detailed sweep across selected plots can be performed. Clicking in each column sends the entire plot to the context window and a vertical green bar indicates current

Figure 31: Detailed analysis occurs for a column section across multiple plots. Left, right, up, and down buttons facilitate navigation during reconstruction and detailed analysis.

plot location. Removing an item from the clipboard removes it from the display and allocates the extra space to the remaining column segments. Normalization in this view only includes clipboard plots and is recalculated as plots are added and removed. Users can also toggle residue name abbreviations to further connect alignment information with correlation values.

## 4.2 Scenario

We now present an example scenario that illustrates how WaveMap [3] can be utilized for better understanding of flexibility data. It specifically highlights the utility of our approach in detecting a small but significant area among a set of similar allostery response plots. A good reason for performing such analysis is to find model parameter sets which result in similar overall response but exhibit a small difference

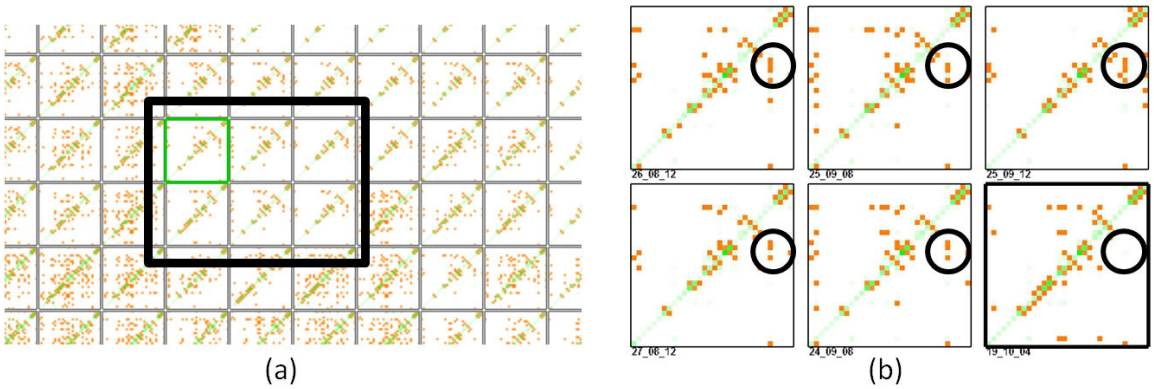



Figure 32: (a) A section of the Jigsaw layout is identified for further analysis based on similar global features (2 decompositions). A known plot is highlighted in green. (b) Closer examination of features reveals noticeable differences. Circled regions indicate a point of difference in one parameter set (lower right).

which could explain subtle variations in behavior. The data set used in the example represents flexibility response measures from the CheY [58] protein.

A protein scientist pre-selects a set of parameters (ie a plot) which exhibit a desired global behavior. The analyst suspects that other parameter sets globally similar to the selected plot have differences which may explain subtle variations in behavior. However, the values and residue region resulting in this behavior are unknown. The analyst searches for the known plot in the overview by entering the identifier into a search box and it is highlighted in green. Places of flexibility trend changes within this parameter set indicating possible differences are difficult to locate in the context of other similar plots. The analyst moves the slider which changes the level of decomposition and examines the resulting features (Figure 32(a)).When the wavelet features are displayed after two levels of decomposition and after the elimination of features having a small magnitude, the analyst sees a pattern of interest within the known plot indicating the changes in flexibility. He/she uses the Jigsaw layout [89],

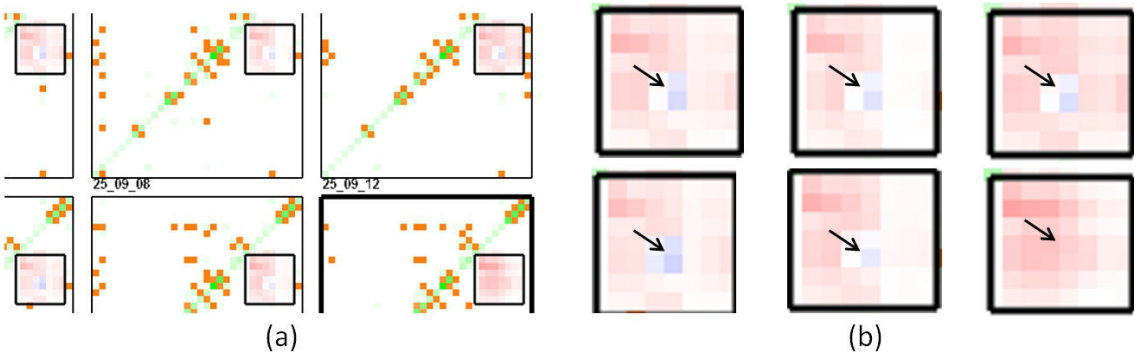



Figure 33: (a) The coordinated lens reveals the pattern that the features emphasize. (b) A larger view of the lens shown for clarity.

[92] and identifies several adjacent plots with similar features to the known plot to ensure global similarity (Figure 32(b)). The group of plots are placed on the clipboard for further examination.

After pruning the parameter sets, the analyst proceeds to the feature exploration window. In this window, the analyst easily compares corresponding regions of the selected plots with the help of the coordinated lens (Figure 33). It is confirmed that the feature plots are similar but have noticeable differences. Of particular interest is the feature present in all but the bottom, far-right plot in Figure 32(b). The coordinated lens (33(a) and 33(b)) reveals that the area of interest highlights a sharp change in flexibility (a small blue section in the middle of red) except for the one parameter set.

To more accurately define the region of difference, the location is visited (Figure 34(a)) and the residue numbers marking the area of change are revealed. The coordinates are propagated to the detail analysis window (Figure 34(b)) and the differences in response can be mapped to the specific residue numbers. The analyst can now

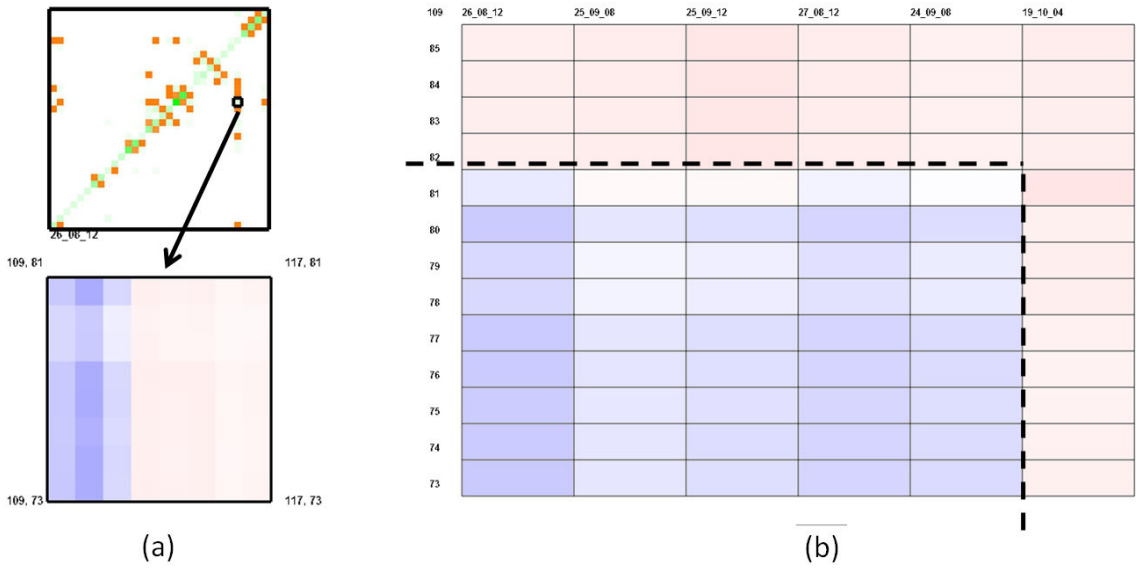



Figure 34: (a) Plot coordinates for the region of interest are found during reconstruction. (b) Detailed analysis allows further mapping of flexibility values to residue numbers among multiple plots.

further examine the physiochemical properties found in the plot lacking the small blue area to see if this region is perhaps responsible for variations in behavior or if the model should be modified.

## 4.3 Evaluation

### 4.3.1 User Study

A formal user study has also been conducted to confirm that discovering similarity or dissimilarity can be eased by feature analysis. Eight student subjects conducted the study one by one. Experience in protein analysis was not a requirement since we were interested in the ability of this technique in easing visual examination and not interpretation of the data. Each subject was given three tasks on paper using three different subsets of the allostery data. Each of the three tasks consisted of 12-15 plots (Figure 35). Subjects were asked to identify the most similar plot according to the

preselected sub-region denoted by a rectangle. (The "correct" match was determined during the design of the study and was based on the visual similarity of the HH wavelet features.)

The tasks each consisted of two subtasks. One subtask only showed the original values and another subtask set showed the original values accompanied by feature plots after three decompositions of the HH subband. Both original values and features below a common magnitude were eliminated to ease the comparison for both sets. The same plots were used for each subtask but the order was shuffled between the original layout and the feature layout subtasks. Half of the subjects were given the original plots first and the other half were given the feature plots first. Additionally, the plot that was the correct match for one layout was the starting plot for the other layout (Figure 35). We did this so that the same plots would be common across a given task while eliminating any biases that may arise by starting with the same plot in both subtasks. For the task shown in Figure 35 users were given the raw data plots 35(a) and asked to find the plot that they thought most closely matched the outlined section. After the selection, the subject was given 35(b) and asked to do the same. The matching pair (35(c)) was shuffled before the second subtask. This was unknown to the subject. One task allowed the subject to browse enlarged plots for additional assistance (one per page) during matching.

Users were asked to rate task difficulty and the confidence that they had chosen the match using a 5 point scale. The accuracy of the results was analyzed after the study. Subjects were also asked for comments after the tasks were completed. The questionnaire is shown in Figure 36 and the cognitive guidelines [30] are listed below.

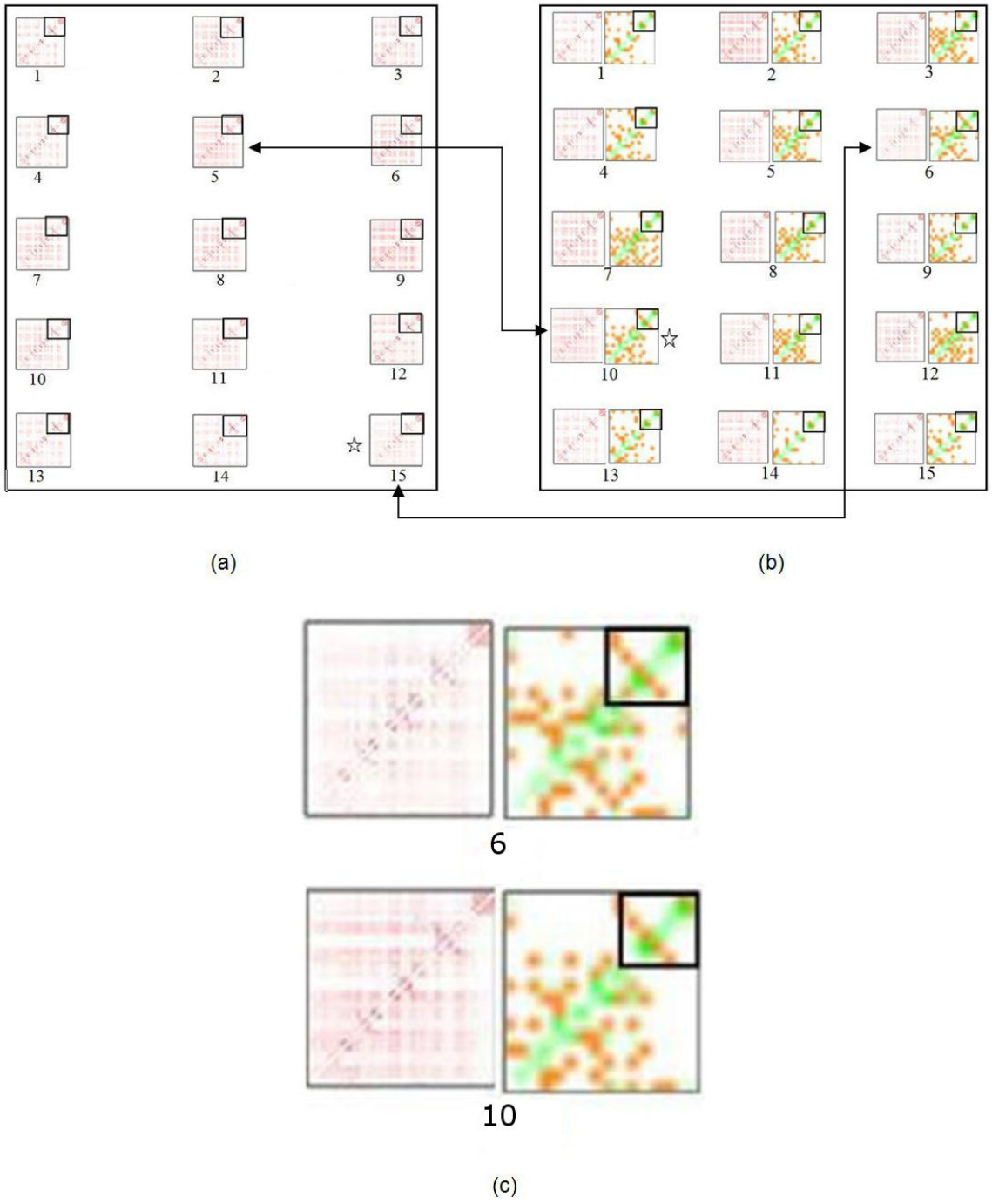



Figure 35: In this task the user was given the raw flexibility plots in (a) and asked to select the most similar plot to the one denoted with a star according to the outlined section. The user was asked to do the same with the feature values accompanied by the original values as shown in (b). The matched pair consists of plots numbered 6 and 10 in the feature view. The raw flexibility plots correspond to plots 5 (10 in the feature view) and 15 (6 in the feature view). They are shown in (c).

**Task ___ - ___** Time Task Started _____

**Subject #___** Time Task Ended _____

***Matching Plot #*** _____

***Question 1: Is it easy to make the drawing?***

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Very easy | Easy | Neutral | Difficult | Very Difficult |

***Question 2: Is it easy to make the match?***

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Very easy | Easy | Neutral | Difficult | Very Difficult |

***Question 3: Is it easy to find pattern differences within the corresponding regions?***

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Very easy | Easy | Neutral | Difficult | Very Difficult |

***Question 4: Is it easy to find pattern similarities within the corresponding regions?***

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Very easy | Easy | Neutral | Difficult | Very Difficult |

***Question 5: How sure are you that the most similar pattern for the corresponding region was found?***

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Very Sure | Sure | Neutral | Unsure | Very Unsure |

***Questions 6: Rate the following for step 2 of this task (finding the match). Descriptions can be found in the document "Cognitive Load Description".***

| *Demands* | *Ratings for this task:* | | | | |
|---|---|---|---|---|---|
| *MD* | *Low 1* | *2* | *3* | *4* | *5 High* |
| *PD* | *Low 1* | *2* | *3* | *4* | *5 High* |
| *TD* | *Low 1* | *2* | *3* | *4* | *5 High* |
| *OP* | *Excl 1* | *2* | *3* | *4* | *5 Poor* |
| *EF* | *Low 1* | *2* | *3* | *4* | *5 High* |
| *FR* | *Low 1* | *2* | *3* | *4* | *5 High* |

Figure 36: User form given to subjects during the study.

- **Mental Demand (MD)**: How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?

- **Physical Demand (PD)**: How much physical activity ws required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or de-

manding, slow or brisk, slack or strenuous, restful or laborious?

- **Temporal Demand (TD)**: How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?

- **Performance (OP)**: How successful do you think you were in accomplishing the goals of the task set by experimenter? How satisfied were you with your performance in accomplishing these goals?

- **Effort (EF)**: How hard did you have to work (mentally and physically) to accomplish your level performance?

- **Frustration Level (FR)**: How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

The results were overwhelmingly in favor of wavelet analysis and are shown in Figure 37. The wavelet feature tasks scored lower than the original data for the amount of demand (MD, PD, TD), effort (EF), and frustration (FR). Users thought that they performed better on the experiment during the feature tasks (OP). Furthermore, all subjects were able to correctly match the most similar plots when features were provided for all tasks. However, only one plot was correctly matched in the original plots only subtasks (out of eight subjects over three tasks). Additionally, the perceived difficulty was much greater for the original plots only subtasks and subject confidence that they had chosen the correct plot was higher for subtasks with the feature plots.

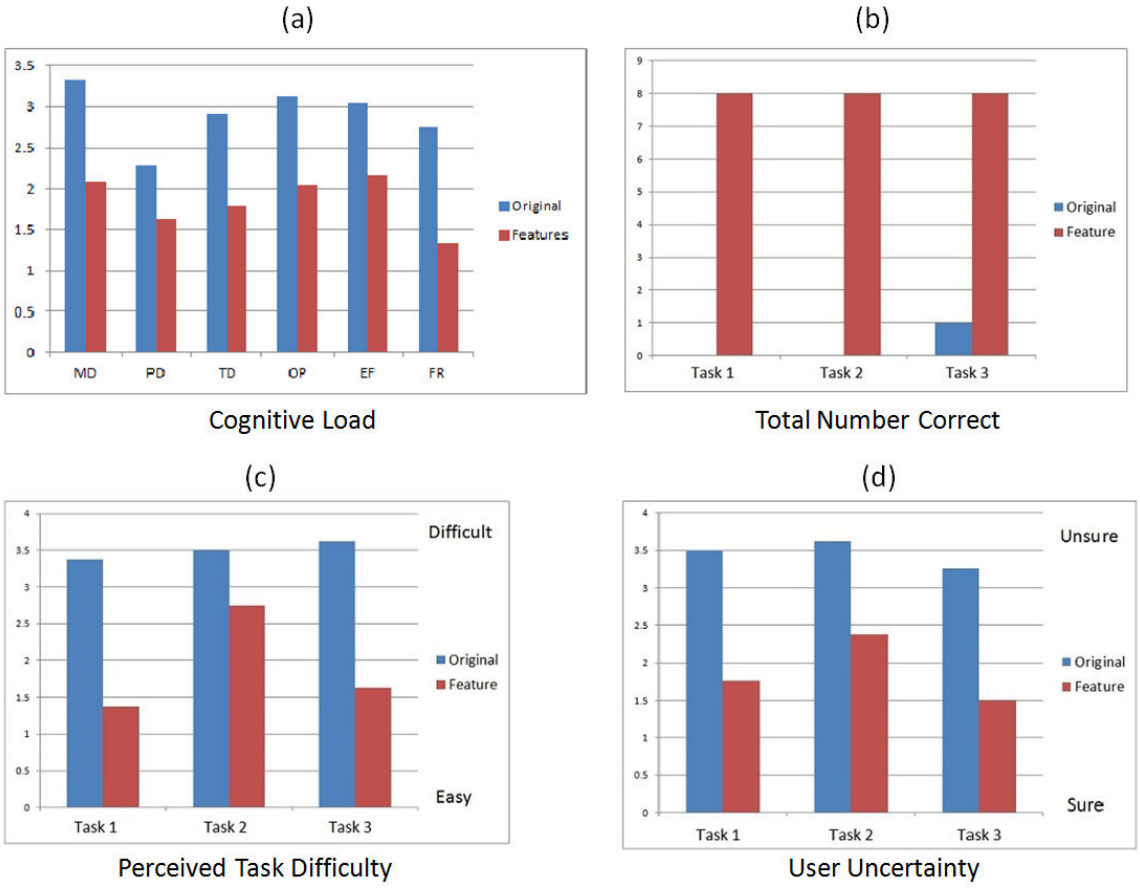



Figure 37: Wavelet features allowed users (a) to find more correct matches with (b) less perceived task difficulty and (c) less uncertainty.

One subject became so frustrated with the original plot only subtasks that he gave up on finding a match. Several subjects said in the post-test that they preferred the feature plots because the patterns were much easier to identify. In task 2, enlarged plots were provided to the subjects but did not aid the users in more accurately finding matches. This shows that zooming alone may not be very effective in assisting plot exploration.

The tasks in the user study provided a concrete example why the feature plots can effectively help users. Although this example may not be biologically significant, it illustrates how the methods presented here can be used to find spatial trends from

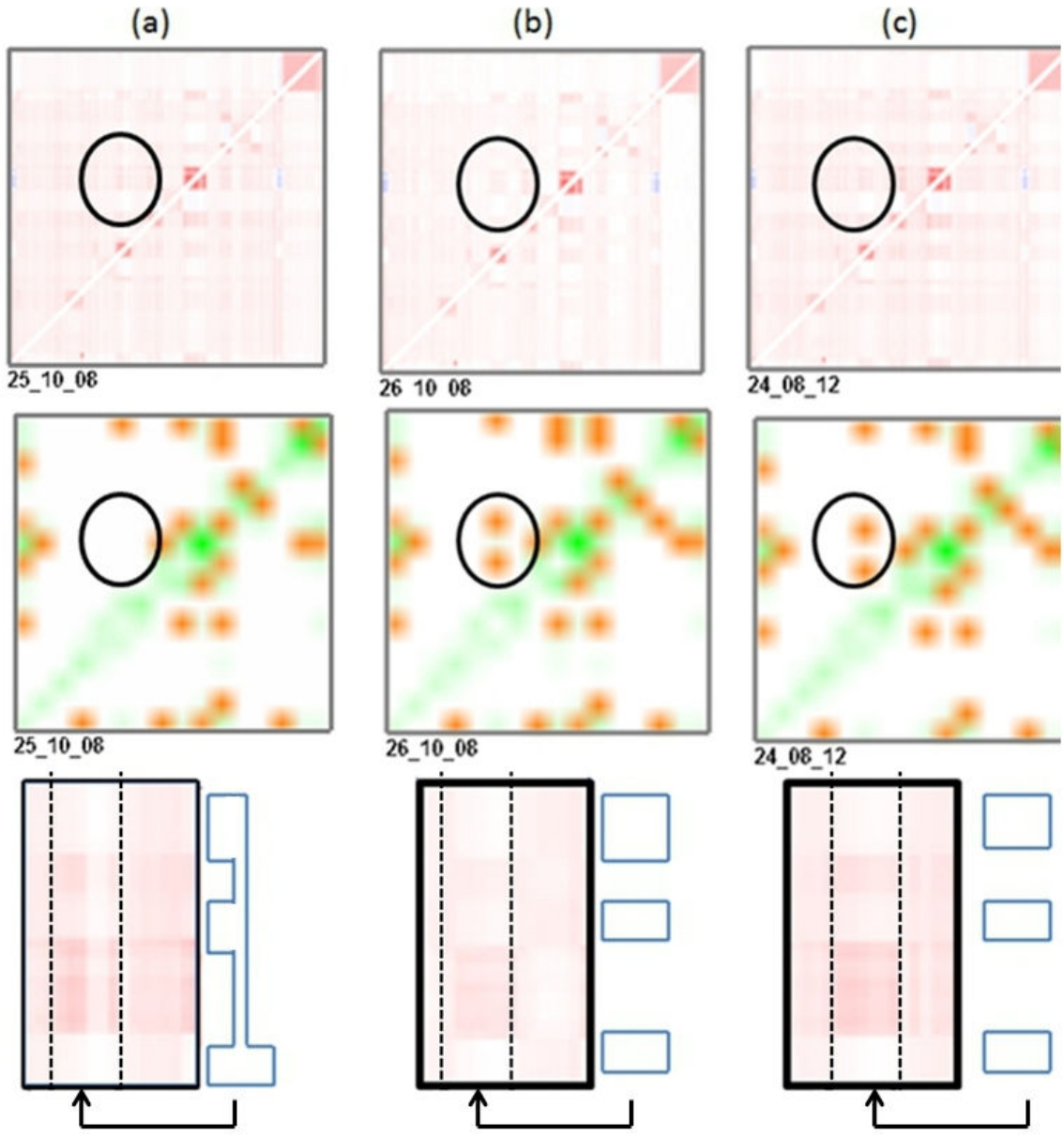



Figure 38: Three parameter sets (a, b, c) used in one of the tasks illustrating that the features make identification and comparison of subtle patterns easier. Top row shows the original data plots and the middle row shows the features which correspond to the circled region. The bottom row displays the very light, subtle patterns that resulted in the middle row features. For clarity, the outline of the dominat pattern is shown to the immediate right of each section and this pattern highly resembles the features. The areas of change in (a) are much more difficult to define and do not survive the threshold increase and decomposition change for this subband.

protein prediction data. The target plot for task 1 is shown in Figure 38(b) and its matching plot is shown in Figure 38(c). Seven out of the eight subjects chose

the plot shown in Figure 38(a) which did not match the target plot. The circled regions highlight the major difference between the plot in Figure 38(a) and the plots in Figures 38(b) and 38(c). It is difficult to distinguish in the original plots but the features provide much clarity and can be easily identified. The section corresponding to the region of interest is enlarged in the third row of Figure 38. The bounds of the section were found by examining the features in our tool as described in the case study. Upon closer examination the checkerboard like sections for 38(b) and 38(c) in the third row resemble the pattern formed by the features. The data plot in 38(a) also has a checkboard-like pattern similar to 38(b) and 38(c). However, this pattern is interrupted by a significant white column making the pattern much smaller and more ill-defined. Features in the selected region do not survive the threshold or are not formed in this level of decomposition.

### 4.3.2 Expert Evaluation

The scenario described above is a general-purpose example of how system components are able to meet the high-level needs of domain scientists. Domain analysts have further identified two specific scientific tasks that parallel the above scenario and illustrate the effectiveness of this approach. The scientific tasks are listed below. (The following was communicated by domain analysts via email.)

- **Identifying the QSFR feature(s) that lead to conservation of function across a protein family and the QSFR differences that lead to known functional differences.** *In this scenario we are going to compare a single QSFR metric of interest...across a number of evolutionarily related proteins.*

*While these proteins are found in different organisms, they all have (basically) the same function, meaning that at a high level function is conserved, but the low level details might vary. The size of these families in our planned studies will range from 50 to 200 proteins. Note that model variables (i.e., parameters) will be strictly controlled for in this scenario, and are not part of what WaveMap is trying to understand.*

*So, what properties are conserved across the family that mediate this conserved function? WaveMap will thus be used to identify features from within the FLX-FLX that are most conserved across the family. The conserved properties will thus represent likely candidates for mediating function because evolution conserves things that are important.*

*In addition, there will be known (and unknown) small functional differences that arise due to divergence. So, for example, assume that we can a priori classify the protein family into three different functional sub-families. This information is based on known differences from experiment. Then, we use WaveMap to cluster the family based on the features. Those features that best reproduce the functional clustering represent likely mechanistic features leading to functional diversity.*

*Finally, not all of the proteins will be experimentally characterized due to the cost of such efforts. Therefore, once WaveMap has classified the known experimental space, we can infer what type of functional variants the uncharacterized proteins fall into.*

- **Identifying conserved allosteric response across a single protein using delta-QSFR metrics.** *In this scenario, we are going to construct delta-QSFR metrics based on a difference between the unperturbed (normal) and perturbed variant of the same protein...That is, we introduce a hypothetical clamp to a single amino acid that restricts its flexibility, and then we recalculate the QSFR properties, which are now presented as difference maps. So, the FLX-FLX plot is now a delta-FLX-FLX. One at a time, we systematically move the "clamp" along the protein sequence to interrogate the effects of rigidifying each position. Thus, we end up with N-aa delta plots, where N-aa is the number of amino acids in the protein of interest. As above, all model details are strictly controlled for, and are not part of the WaveMap investigation.*

  *The scientific question being asked is, "Which amino acid clamps lead to similar results of functional interest?" Thus, the natural thing to do is to simply cluster the N-aa delta plots based on their overall composition. However, this is likely not to be sufficient because it describes all possible pairwise couplings. Rather, there may be a sub-section of the FLX-FLX plot that is believe to be most related to function, and thus changes to this sub-section would be of highest interest. Therefore, we again cluster using WaveMap features within this region. Therefore, we have now answered the question posed above.*

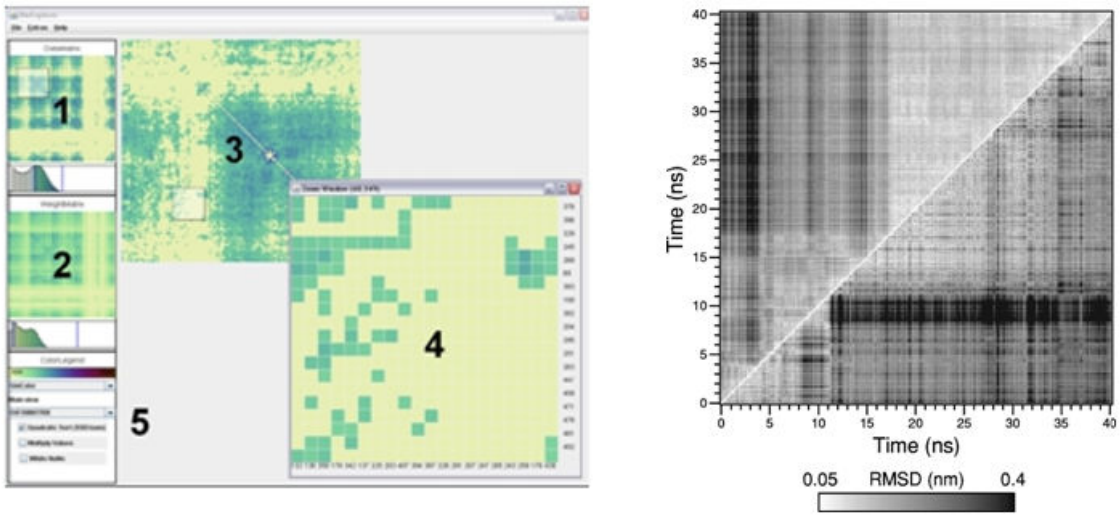
## 4.4 Discussion

### 4.4.1 Extensions

The initial purpose of the proposed approach was to help computational biologists investigate outputs from the Distance ConstraintModel [34], [48]. Domain analysts were impressed with the ability of the wavelet extraction technique in providing multi-resolution views and simultaneously reducing the data. They are very interested in extending the system so that one-dimensional data can be analyzed as well. One-dimensional data of interest includes the data that appears earlier in the DCM pipeline (such as energy curves) as well as one-dimensional outputs (such as individual column or rows in the two-dimensional plots). These are both important avenues of future work so that inputs and outputs can be more tightly coupled.

As the approach was refined and the data set was better understood, we realized that biologists often need to visually describe data in context of spatial arrangement. Our approach helps locate, identify, and evaluate both local and regional factors influencing global behavior. However, any data set where variables of interest can be arranged in a row/column layout consistently across many items and has spatial or other meaning can benefit from this approach. In the general case, the index $i$, $j$ would represent the value of an output $i$ for a parameter $j$. Examining rows (or columns) is useful for evaluating many model or experimental responses for a single variable and each column (or row) reflects a single result for the combination of interdependent variables.

As a first step in showing how our technique can be applied to other fields, several




(a) (b)

Figure 39: (a) A recently developed tool for analyzing molecular dynamics with zooming, sorting, and weighting [10]. (b) Plotting molecular dynamics versus time [31].

other examples in bioinformatics will be examined. For example, other molecular models than the one that is the focus of this work can use the row/column layout described above. Figures 39(a) and (b) show other applications that can benefit from our approach. Figure 39(a) is a matrix visualization tool for finding uneven distributions among molecular dynamics. Like the correlation measures presented earlier (section 3.3.1), Bremm et al [10] map these correlations to a three-dimensional protein sequence. The authors note that changes in dynamics often influence both local and global characteristics. The tool relies heavily on zooming and sorting and still suffers from the lack of tools that are provided with WaveMap [3]. Primarily, the tool lacks automatic guidance techniques during initial viewing and from the lack of contextual preservation during analysis.

Another example of protein analysis that would benefit from WaveMap [3] can be

found in [31]. In this case, scientists are interested in plotting molecular dynamics versus time (Figure 39(b)). The dynamics of a single atomic structure is tracked across time creating a space in which few of the data items will greatly influence global behavior. The authors do not mention any software tools used in analysis. The work presented here can help 1) find those few places of great influence by using fine-grain analysis and 2) preserve any meaning associated with time.

The final example from the bioinformatics community which can benefit from WaveMap [3] is microarray analysis. Microarrays are created by placing thousands of genetic material samples on a chip and allows the simultaneous analysis of an organism's genetic response for a single experiment [1], [94]. The process includes *hybridization* to bind the genes to a small, thin structure. Red and green dyes are used to *express* the level of activitiy of a given gene. The expression level is mapped to intensity and the results are summarized in an matrix organized by genes (rows) and experimental sample (column). A microarray is shown in Figure 40(a).

Zhang et al [94] discuss the visualization techniques that have been applied to microarray analysis. These include scatterplots, heatmaps, parallel coordinates, dendrograms, hyperbolic views, and an assortment of graph drawing methods. The authors also list several common needs associated with microarray analysis. Among those listed, WaveMap [3] is a suitable fit for two of those items. First, the authors list the need for data reduction since much information can be hidden by clutter. One way to perform data reduction is through wavelet analysis [45]. Since the application of wavelet analysis to microarrays has been extensively studied [47], [5], [44], [87] there would be little effort required for extending this tool to gene expression data.

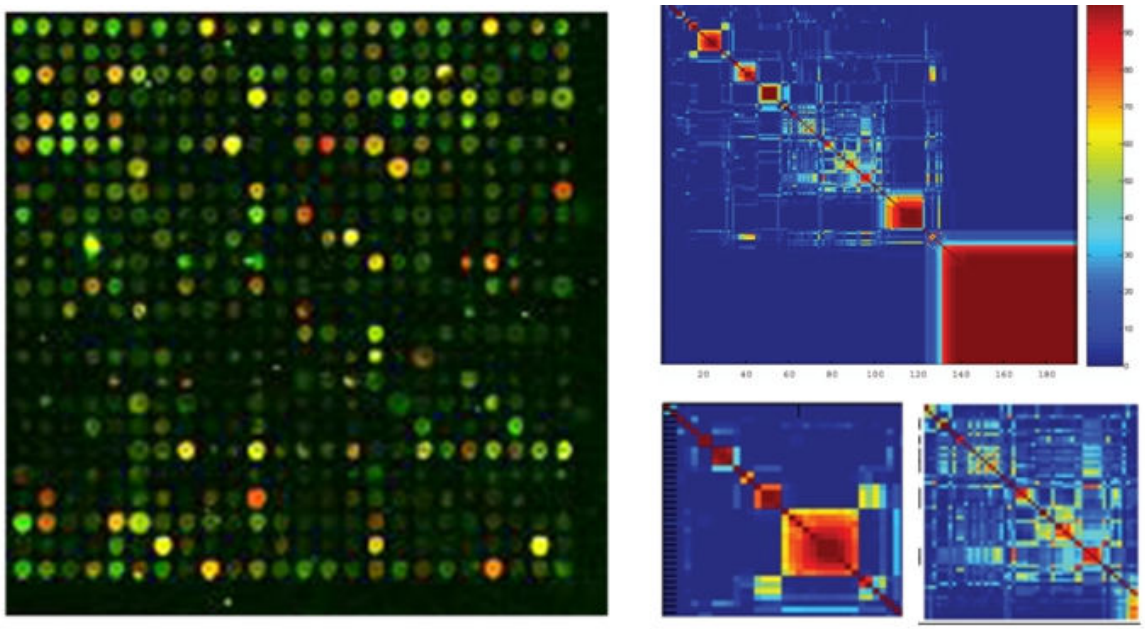



Figure 40: (a) Spot intensities in a microarray indicating gene expression [1]. Visualizing cluster uncertainty in microarray data [74].

Second, the authors note that time series experimentation for genetic expression is becoming more important. The method presented here will preserve the time sensitive value (either global or anomalous behavior) of any variables being measured as experimentation progresses.

Although many visualization and automatic analysis techniques have been applied to mining microarray data, there still exists a great deal of uncertainty in the results even before they are visualized. Among the chief contributors to uncertainty is deciding on the proper clustering algorithm and the most appropriate parameter settings for those algorithms. Rasmussen et al [74] use Dirichlet process mixture models, a non-parametric Bayesian approach, to form similarity measures among gene expressions. The authors visualize the co-occurrence probabilities (Figure 40)(b) such that

the resulting plot is similar to the outputs of the model used by the BMPG. In this display, each index represents the probability that the gene at row $i$ and the gene at column $j$ are in the same cluster. WaveMap [3] would be able to represent the global cluster structure and reveal when similar genes cluster together, leading to insight regarding both the clustering algorithm and the behavior of gene pairs.

### 4.4.2 Limitations

During experimentation a several limitations were found. First, the effectiveness of this approach is influenced by the wavelet used and the current level of decomposition. Each wavelet feature is the result of a correlation between two signals (data signal and wavelet signal) [45] and changing the wavelet alters its correlation to the data signal. This means that, for a different wavelet, the interpretation of the extracted features in relation to the underlying data may be different. Our initial implementation relied on a well-known wavelet technique and serves only as an initial attempt. Refinement and study are needed to develop custom wavelets so that more specific feature sets can be targeted and better defined in terms of biological significance. Furthermore, finding the best level of decomposition that reveals the most useful features for a given plot is accomplished through browsing the results of the entire range of decompositions. The situation becomes much more complicated if other data plots within the same set have interesting features revealed at other levels of decomposition.

Second, the original framework is meant to identify localized neighborhoods which cumulatively result in a feature. However, extensive exploration of corresponding neighborhoods for an entire data set would be quite tedious. In the original approach,

users are allowed analysis of a neighborhood for a single plot (feature view) before placing that information in a limited contextual setting considering only one, fixed height column containing a small number of data plots (detail analysis). Depending on the subband used and the level of decomposition investigated, data yielding insight yet contributing to the feature pattern may not be spatially close to the center of the feature. Furthermore, an accurate representation of the original data is also assumed. This may not be the case because of shifting during wavelet transformation and artifacts from noise. As discussed before, boundary conditions can also lead to a distorted representation of the underlying data near plot extremities and is an active area of research in signal processing [11], [27].

Third, a user could suffer from change blindness [67] in the workflow. For example, the clustering algorithm starts from a random seed and will result in different layouts each time it is executed. However, scientists wanted layouts that would persist across sessions. To stabilize the layouts, all cluster coordinates (raw and feature values) are calculated the first time the data set is processed. During program execution, the coordinates are read ensuring consistency but any desire for stability during dynamic clustering would be problematic. Another example is the change in decomposition. When the slider is moved, there is not a visual relationship bridging the layouts before and after the change. Any subtle changes in the feature location or values within a plot are lost. Solutions to these problems, including alternate layout methodologies, should be explored.

# CHAPTER 5: EXTENSIONS FOR SUBSPACE EXPLORATION

Shortcomings of the original approach were listed earlier (section 4.4.2). One of the disadvantages is the lack of tools for efficient exploration of corresponding sub-regions across an entire data set. The following sections show how this shortcoming was addressed. First, a refinement of the original tasks are presented. Second, related work in protein subspace analysis and previous attempts at visually conveying subspaces are given. Third, the proposed extensions are described. The extensions consist of a plot carving view, a subspace cluster view, and a sliding subspace view. Finally, an example illustrates how the proposed extensions can aid analysis.

## 5.1 Task Refinement

Problems faced in protein subspace analysis include organizing and separating sub-regions across the entire data set, adequately describing differences among corresponding neighborhoods, and facilitating exploration of neighborhood border areas. As solutions were sought for these challenges, the high-level tasks mentioned earlier (section 3.3.2)were further refined to include the following subspace tasks. They contain elements of all three original high-level tasks but better represent the specialized nature of subspace exploration. The refined subspace tasks are listed below.

**Categorizing a given sub-region and its components as rigid, flexible, neutral, or mixed.** Flexibility for a region can be difficult to visually compare

because flexible/rigid regions may take place at different parts in the sub-region. Traditional distance measures, such as correlation, may not always be appropriate. For example, regions having a similar total flexibility may have a low correlation value. This can occur when local areas have the same number of flexible/rigid residues of similar magnitude but in different locations. Furthermore, analysis of neighborhoods can be difficult because highly flexible(rigid) regions can overshadow/hide rigid(flexible) residues which may not be noticeable if interwoven within the flexible(rigid) regions.

**Analyzing the changes in conservation for a sub-region across multiple parameters within context of its neighborhood.** Knowing how shifting the sub-region or changing its size across the entire data set will allow scientists to know how extensive the effects of parameters are. Parameters may only affect a small region or they may result in a more concerted effort accross many residues. Additionally, the extent of the effect may be different for various parameter sets. Knowing the extent of a given group of parameters will aid scientists in developing methods to more accurately model or predict behavior.

## 5.2 Background in Subspace Exploration

### 5.2.1 Protein Subspaces

Finding important subspaces allows analysts to identify local patterns and features which contribute to global behavior [71]. Subspace construction and exploration is common in protein analysis. For example, subspaces have been used in the synthesis and analysis of experimental proteins [52]. Often, these type of *mutant* proteins differ from a *wild type* by varying a single amino acid. Subspace modeling has helped alle-

viate some of the difficulties encountered with numerous mutants that make training and applying machine learning problematic.

Henk and Liere [33] divide the conformational space into two subspaces: essential and physically constrained. The essential degrees of freedom describe functionally important motions and physically constrained describes local variations that are not relevant to functionally important motions. From these spaces, a protein analyst can construct and visualize a three-dimensional ribbon diagram. Interactions for the diagram include dragging an atom to a new position resulting in a new displacement, monitoring the hinge properties with an angular widget, and a measuring stick widget for deciphering distances between atoms.

Interesting clusters may be difficult to find in noisy or in high dimensional data [2] and protein analysts lack the appropriate tools for overcoming these difficulties. Current methods for exploration are insufficient in two ways. First, many of the methods employed for constructing and evaluating subspaces rely almost exclusively on computational methods, ignoring the abilities of humans to establish and compare patterns. Second, if visualization is used as part of the solution such as described above, only a few proteins or environmental conditions can be evaluated at a time. Solutions that effectively address the need for subspace exploration during protein analysis will utilize both computational methods and visualization for large scale data sets.

### 5.2.2 Subspaces in Visualization

Visual methods for handling large data sets can provide an alternative to using computational methods only. There are multiple works that either explicitly or implicitly address interactive analysis or construction of subspaces. Several of those works are now presented.

Many existing approaches for detecting and finding subspaces rely on quality metrics and reordering of dimensions. For example, Ferdosi and Roerdink [21] note that problems exist in many current techniques used in the visualization of high-dimensional data sets. Scatterplot matrices can fall victim to crowding and parallel coordinates require the proper ordering of items to reveal interesting data spaces. Ferdosi and Roerdink present three approaches based on subspace clustering and ranking to help alleviate the shortcomings found in parallel coordinates and scatterplot matrices. The first is *structure-based full ordering* for parallel coordinates. The first part of this step is to rank all one-dimensional subspaces by a combination of density distribution multimodality, mode significance, and mode separability. The top performing subspace reappears first in the reordered sequence. Next, the two-dimensional subspaces are ranked by quality. The subspaces considered in this step only include high-quality subspaces from the one-dimensional ranking. This process continues for higher-order subspaces except that the highest ranking subspaces with a known number of clusters will be considered in the next iteration. If no subspaces exist containing the given number of clusters, then the highest ranking subspace will be used in its place. *Structure-based partial ordering* for parallel coordinates is sim-

ilar to the full ordering scheme except that sequence creation is stopped when there are not any subspaces with the known number of clusters. *Structure-based simple ordering* is useful for filtering noise in scatterplot matrices and is based on density image quality. The filtering threshold is automatically set to the average quality of the one-dimensional subspace. User interactions include the ability to change the ordering of the scatterplots by dragging and dropping dimensions and the ability to swap dimensions in the parallel coordinates. Although these methods improve some of the most widely used visualizations, it does not fit the problem described in this work. If used for exploring plot subspaces, the above methods would not easily allow the inclusion of raw data values which are important in this work. The techniques would also not facilitate the exploration of rows and columns within plot subspaces. Like many of the matrix visualization techniques, finding and analyzing subspaces by Ferdosi and Roerdink rely on reordering. As previously described, reordering would not be able to handle dimensions or subspaces which are separate but may have highly interconnected effects.

Guo et al [25] present a platform for finding subspaces based on interactive feature selection and easily examining hierarchical clusters. For automatic guidance to interesting subspaces, an entropy matrix is constructed. In the matrix, the cell's color represents a measure between two dimensions. The maximum conditional entropy (MCE) is displayed above the diagonal and $\chi^2$ below the diagonal. Brighter colors represent either low MCE or high $\chi^2$ values and those values can be accessed by moving the mouse over the cell. Clicking on the diagonal allows the user to add or subtract a variable from a subspace and other system components are updated

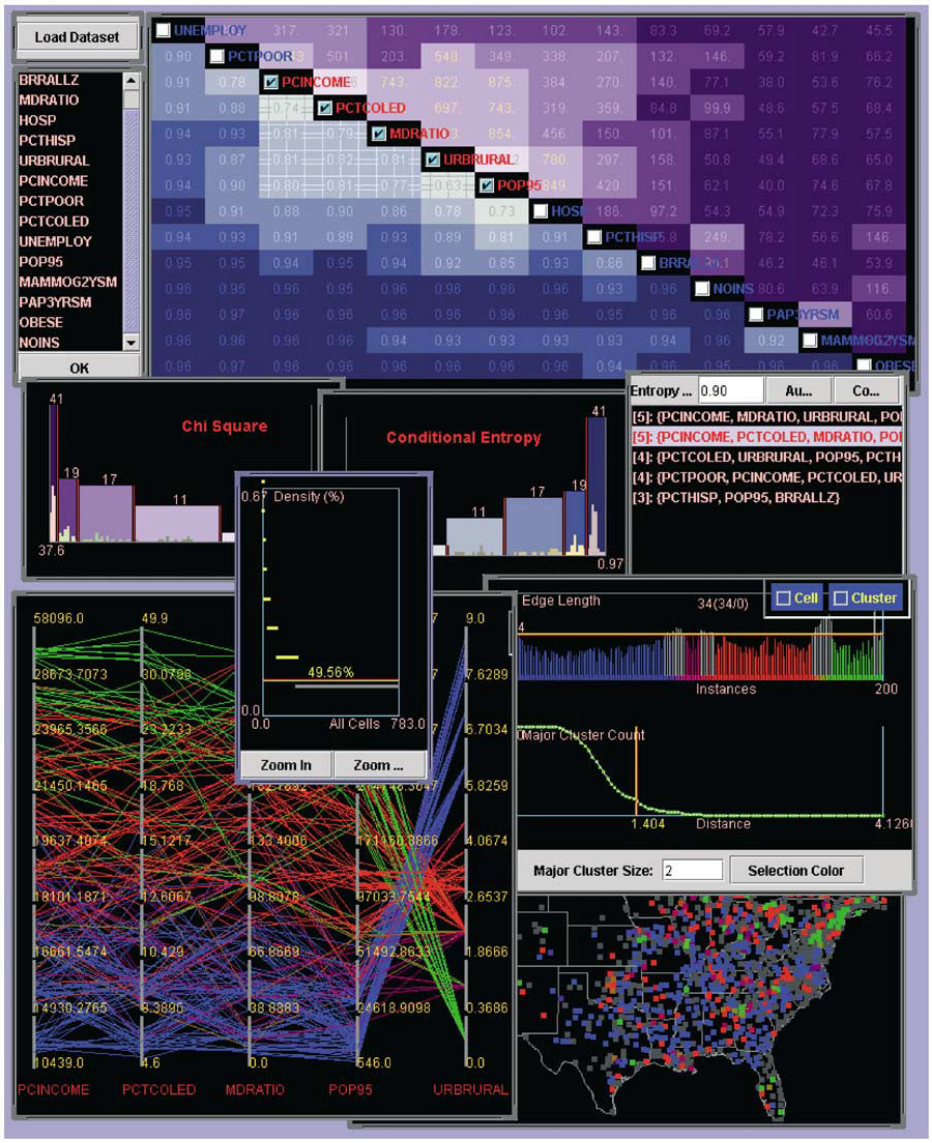



Figure 41: a) Subspace and feature selection [25]. Subspace entropy map (top) and linked visualizations for subspace and cluster exploration.

according to the change. Capabilities exist so that automatic searches are based on entropy values. Figure 41 shows the subspace feature selection platform. The entropy matrix is at the top of the display and is placed among other linked tools for cluster examination. Although the interactive feature selection methods show how automatic techniques and visualization can be combined, the interface shown in Figure 41 is very complicated and could easily hide important information or overwhelm

scientists. Additionally, this example lacks the ability to drill-down into specific data items and compare that data item's attributes across the data set.

Another plaform allows users to isolate subsets of data according to linear trend discovery [26]. The system includes a data space view consisting of a scatterplot matrix. In this view, users are allowed to choose which variables are dependent and which are independent. The model space view consists of linear trend management (viewing, selection, and deletion) as well as linear trend measurement views. A color coded sample measurement map allows parameter partitioning and tuning. Although a positive step forward in the subspace examination of model variables, it presents drawbacks for the application presented in this research. For example, techniques such as parallel coordinates and scatterplots are used which may not scale to the number of conditions being studied by protein scientists. Most importantly, domain scientists may not only be interested in the parameters which exhibit similar linear trends but may be more interested in exploring those parameters whose behavior cannot be easily defined across studies.

VISA (Visual Subspace Clustering Analysis) [2] attempts to occlude redundant spaces with efficient representations a nd user feedback (Figure 42). VISA allows subspaces to be browsed. The greatest obstacle in this first step is being able to compare insights in separate or overlapping subspaces in the same view. To overcome the problem of subspace comparability, a normalized distance function is constructed. Detailed information such as mean and variance can be shown for a selected subspace. Bracketing is a series of views for different parameter settings and is helpful to the user when setting the appropriate parameters that minimize redundancy. Subspace

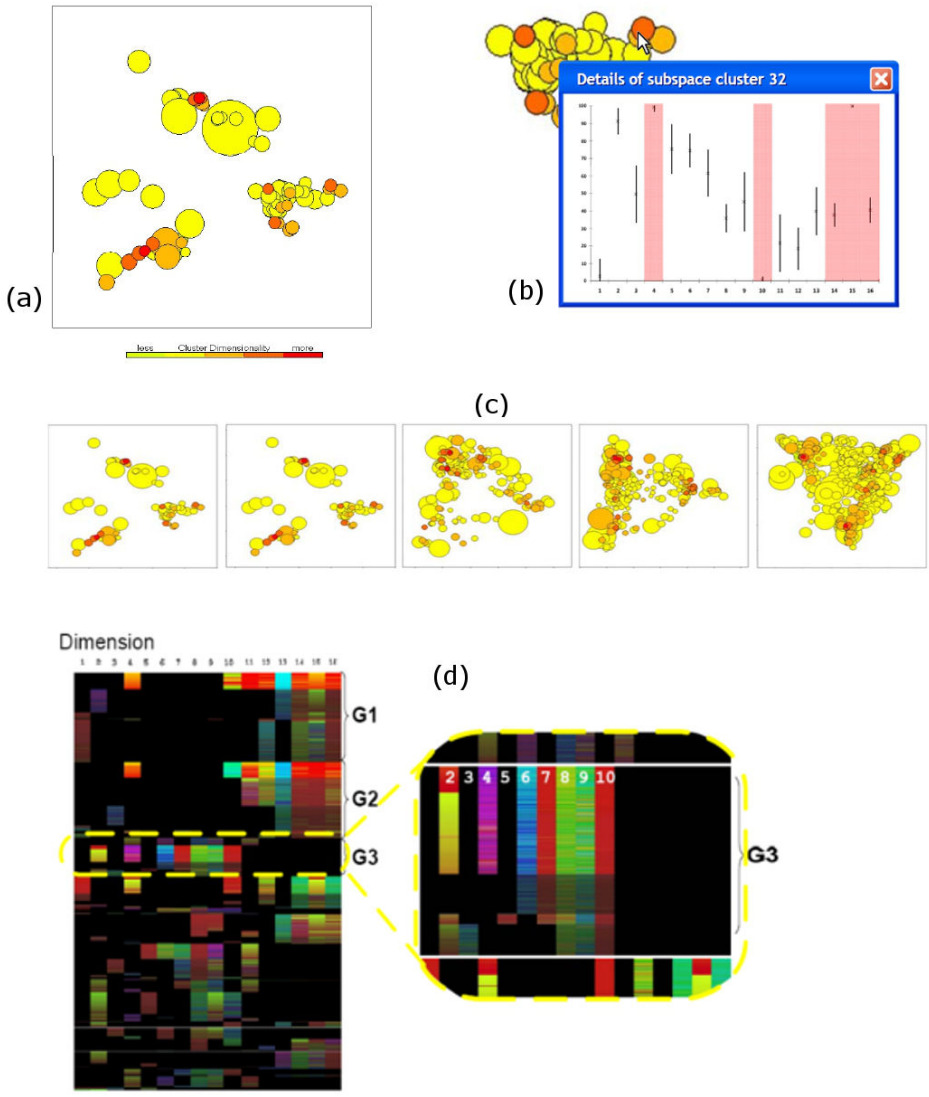



Figure 42: Visa [2]. a) Subspace cluster overview b) Detail view showing statistics for clusters c) Bracketing redundancy d)Matrix of cluster subspaces

matrices can be viewed where rows represent clusters and columns represent columns. Saturation denotes subspace importance through which a subspace *anchor* and dimension values are mapped to color. *Groups* are defined as a set of dimensions within a distance of the anchor. Individual subspaces can be examined even further through the use of a scrolling zoom.

Finding and examining outliers is an important activity for protein scientists. A

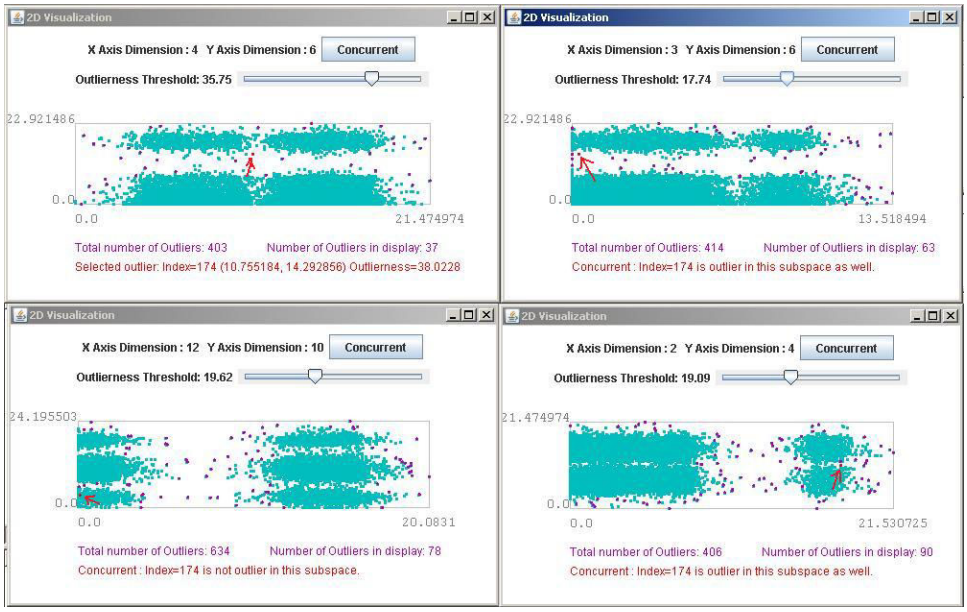



Figure 43: Web-based outlier subspace [46].

system for examining outlier subspaces is shown in [46]. There are many outlier detection methods [13] and this platform allows the simultaneous visual verification and comparison of detection methods. Additionally, users can visualize distributions of the outlier space. The two-dimensional display (Figure 43) outputs the total number of outliers, the number of outliers in the display, and summary information regarding outliers. Additionally, the user can change the threshold for defining an outlier and view a three-dimensional display.

The above is a sampling of the approaches combining visualization and subspace analysis. The major shortcomings of the above works and many other subspace approaches are listed below.

- **Reliance on reordering.** As discussed already, domain scientist may not want to rely on reordering of their data set. Each residue does not occur in isolation and may have large influence on the behavior of other residues which may or

may not share neighborhoods on the three-dimensional protein.

- **Utilization of complex algorithms and quality metrics.** Many systems for subspace analysis rely on complex algorithms and quality metrics which may not be designed to locate subspaces of true interest to domain scientists. Any visualization in these systems is constrained by the understanding of these algorithms by domain scientists and how much overlap there is between what the metrics and domain scientists consider a quality subspace.

- **Visualization of summary clusters** Ferdosi et al [20] note that too many visualization approaches only aid in presenting cluster analysis and do not aid in exploring individual subspaces. Because of the contextual significance of the residues on the three-dimensional behavior of proteins, the inability to explore neighboring regions on a residue-level basis is a critical shortcoming.

This work proposes a solution for finding subspaces of interest in protein flexibility that overcomes the above shortcomings by combining visualization with appropriate automatic analysis. Additionally, this approach will provide a more streamlined approach and greater flexibility than the systems presented here. Specifically, the approach in this work will allow domain analysts to use their unique knowledge within an exploratory environment to decide which subspaces need to be examined and then investigate individual members of those subspaces.

The views developed for subspace exploration in protein flexiblity matrices are now presented and described. Capabilities for subspace analysis are integrated into WaveMap [3] to complement the functionality already presented.

## 5.3 Plot Carving

### 5.3.1 Grid Sections

The plot carving view allows users to employ several techniques in support of exploring and categorizing subspaces. In this view, the entire data set can be divided into corresponsing sections and histograms allow each section to be categorized into its rigid and flexible components. Users begin exploration by dividing plots into smaller chuncks via a method we termed *carving*. Input boxes specify the number of horizontal grids and the number of vertical grids. If the plot cannot be divided evenly into the length or width given, an error message is displayed. After the neighborhood size has been specified, an interactive grid is drawn on the plot in the detail window. The grid dimensions can be changed by entering the new length/width (Figure 44(a)) and then clicking on a refresh button. Later, a free-hand tool is described for greater freedom for choosing subspaces.

Once the grid sections are defined the user can click on the "carved" plot (Figure 44(b)). Scientists can choose a grid three ways. First, scientists may be interested in a certain section because of prior domain knowledge. Second, they may use the previously described wavelet analysis to guide them to areas of interest. Third, they may systematically click on adjacent grid sections until a region of interest is found.

### 5.3.2 Histogram Sorting

After finding a grid section of interest, the main display shows information about the section occurring across all plots. Histograms for the corresponding grid section are shown for the entire data set (Figure 44(c)) and are utilized to categorize the

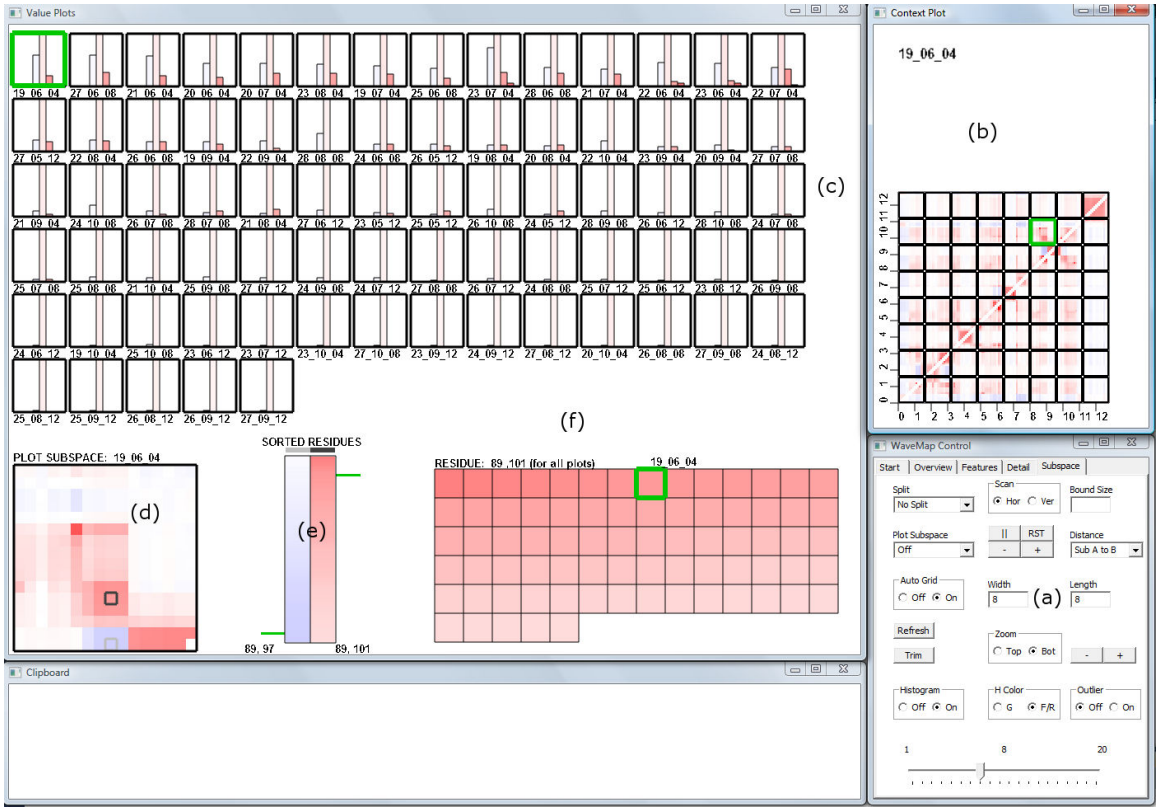



Figure 44: (a) Input boxes for changing the length and width of grid sections. (b) Grid sections dividing the data set into sub-regions. Sections can be selected by clicking. (c) Histogram view with bins colored according to average flexibility of its members. (d) Enlarged view of the selected plot with the currently selected residue outlined in black and the previous residue outlined in gray. (e) The currently selected residue is on the right side of the columns and the previous residue is on the left. Green lines indicate the place of each residue when all corresponding residues are sorted and the residue coordinates are shown below. (f) The current residue sorted across the data set allows users to select a plot of interest. Clicking on a square changes the target plot which is highlighted in green.

components in each individual plot. Histograms have been successfully integrated into visual analytics systems. Guo et al [26] used histograms to display distributions of distances for finding linear trends in model variables. Color is used to match the histogram to data in other views and colored vertical lines show boundaries where any bins outside of the boundaries contain trend outliers. In some instances histograms have played a more central role. For example, Barlowe et al [4] used histograms to

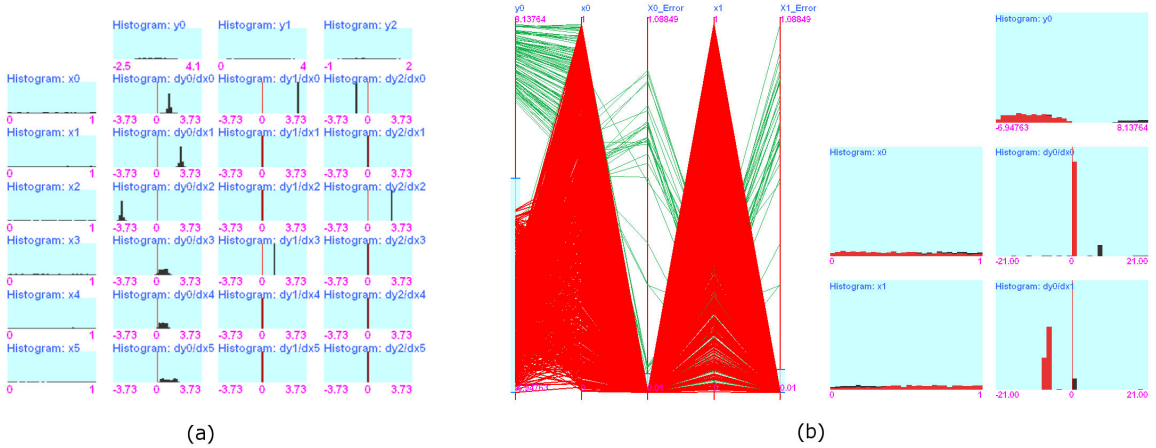



Figure 45: Multivariate Variable Explanation (MVE) [4]. Histograms are an integral part of MVE. (a) Histograms of partial derivatives help locate variables with high correlation. (b) Histograms are linked to parallel coordinates through brushing.

visually explore partial derivatives so that highly correlated variables can be detected and used in interactive model building. A histogram layout linked to parallel coordinates is shown in Figure 45(a) and (b). For the work presented here, histograms provide scientists a way to categorize groups of residues within a plot as flexible, rigid, neutral, or mixed. This compartmentalizes each subspace by varying degrees of flexibility. Bin color is determined by the average flexibility of its members. Data items are normalized for the entire set but the bin size is normalized individually. This allows users to compare bin color across all histograms but accurately maintain the proportion of each bin to the total subspace members.

After each subspace is compartmentalized, sorting the entire space of histograms allows users to categorize and compare whole histograms. There are many algorithms for calculating histogram similarity. Surveys of histogram bin measures can be found in [75] and [43]. Most types of histogram distance measures can be classified as either bin-by-bin, cross-bin, or a hybrid of these. Bin-by-bin distances are calculated

using only bins that correspond to the same position in the histogram. Examples of this type include Minkowski-form, histogram intersection, Kullback-Leibler and Jeffrey divergence, and $\chi^2$ [75]. Bin-by-bin distances have several notable disadvantages. First, they do not consider the similarity of any adjacent bins and may suffer information loss. Second, bin-by-bin measures are sensitive to bin size. Large bins may not distinguish between a large range of data items and small bins may place items within a narrow range into different bins where the distance between them will not be matched. Third, bin-by-bin measures may not produce *perceptual similarity* where histograms considered similar are visually similar.

Cross-bin distances calculate distance among pair-wise bins [75]. Information loss is minimized because more adjacent bins are taken into account when deriving distances. Common distances include the quadratic-form distance, parameter-based distance, and earth mover's signature distances. Cross-bin distances also suffer from several disadvantages. For example, quadratic-form distances do not ensure perceptual similarity and in the case of image retrieval can result in false positives. Because of pair-wise comparisons, cross-bin distances are generally more computationally expensive than the bin-by-bin distances. For example, [6] lists $\chi^2$ with a low complexity, quadratic form distance with a medium complexity, and earth mover's distance with a high complexity. (The only cross-bin measure listed as having low computational complexity by [6] is a measure for cumulative histograms which will not be considered here.)

The histogram similarity measure chosen was $\chi^2$ (Equation (2)) where $d$ represents the distance between two histograms $H$ and $K$, $h_i$ is the number of data items in the

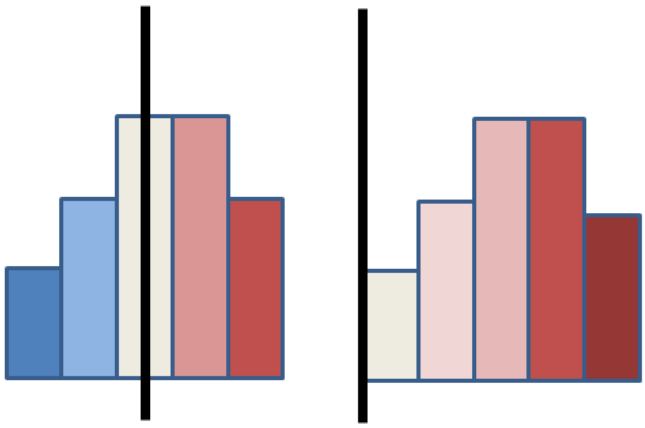


Figure 46: The left and right histograms are identical in shape. When bins are colored according to flexibility, the interpretation of the histogram drastically changes even though perceptual similarity is maintained. (The vertical black line marks the zero point between flexible and rigid.)

$i^{th}$ bin of histogram $h$, and $m_i$ is the average size of corresponding bins.

$$d_{\chi^2}(H, K) = \sum_i \frac{(h_i - m_i)^2}{m_i} \tag{2}$$

A bin-by-bin distance was chosen for three primary reasons. First, users should be able to order plots by similarity during program execution and bin-by-bin measures offer the efficiency that would be required doing this for large data sets. Second, a method for determining similarity among bins having the same flexibility range is desired. A method for pair-wise comparison could have been chosen to preserve perceptual similarity. However, perceptually similar histograms that are identical but differ only by a shift left or right would destroy any meaningful flexibility information encoded by bins (Figure 46). Third, the sensitivity to bin size can be controlled by letting the user interactively choose the number of bins into which the subspace can be divided.

### 5.3.3 Interactions

There are many interactions available for examining histograms. Clicking on a subspace will make that subspace the target. The remaining histograms will be sorted based on similarity to the target. Slider movement allows the number of bins in a subspace to be increased or decreased. Users can also interactively trim the histograms so that a subset of bins can be compared (Figure 47). For instance, scientists may

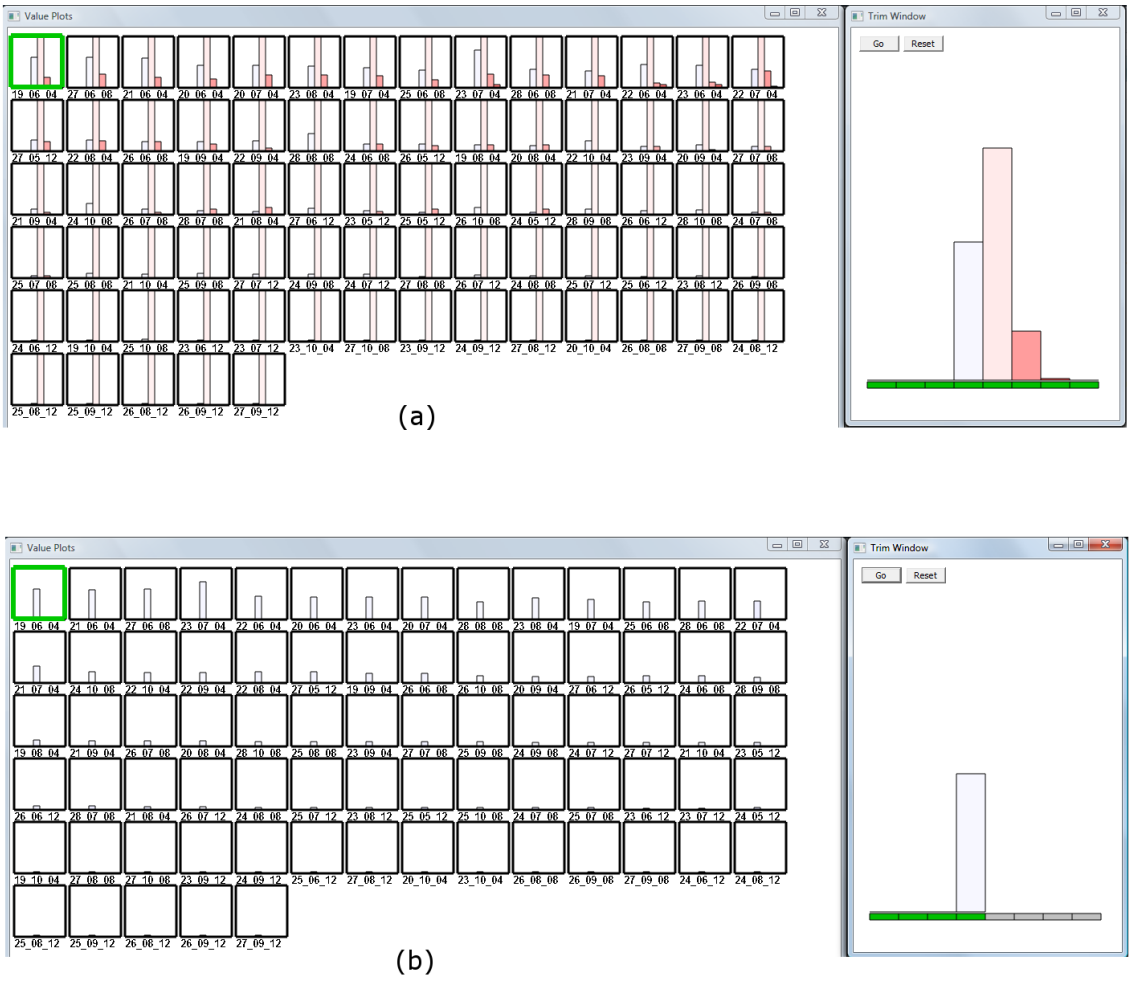



Figure 47: (a) Histograms before trimming is executed. (b) Histograms and trim tool after trimming is executed. Trimmed bins are removed from the histograms and are not considered in similarity calculations.

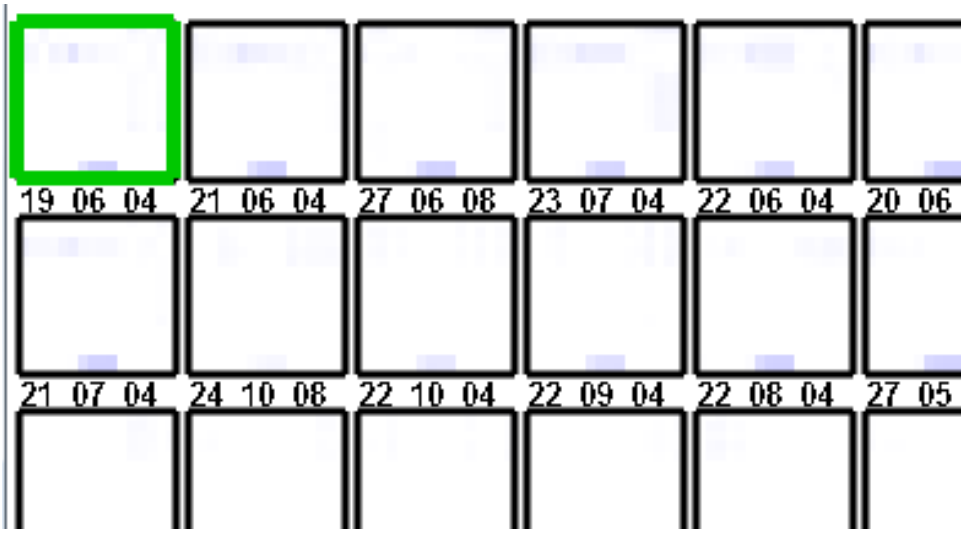



Figure 48: After bins are excluded members of the those bins are filtered from the subspaces when the original values are shown.

see a subset of bins with an average flexibility score (denoted by color) and want to compare the size and color of only the corresponding bins. When this tool is activated, the entire histogram for the target subspace is shown. When bins are clicked the green rectangle below each bin turns gray indicating that they have been trimmed from the entire data set. After the user trims the bins, all distances from the target subspace are recalculated and all subspaces are resorted. This can be accomplished because the columns under comparison do not need to be adjacent, a benefit of the bin-by-bin approach. After a bin is removed the data members contained in that bin are removed from subspaces when the histogram view is toggled off (Figure 48).

Once a subspace of interest has been chosen, the user can investigate individual residues. Individual residues are selected (Figure 44(d)) as the user clicks within an enlarged plot in the bottom left portion of the screen. The current residue is outlined in black and the residue visited just before the current one is outlined in gray. To aid in determining context, the current residue is marked in all of the subspace sections

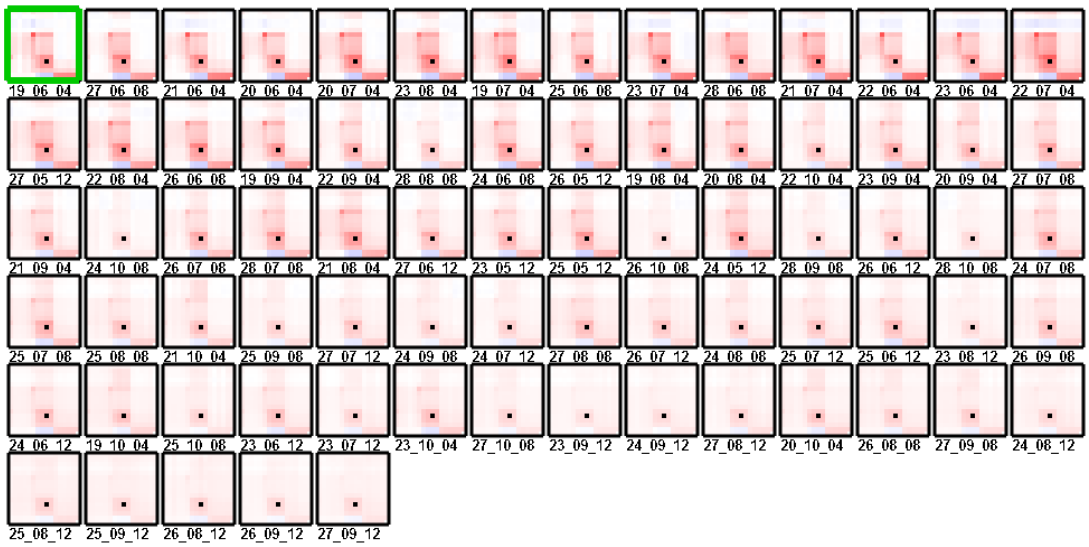



Figure 49: Black dots mark the location of the current residue in all subspaces.

by a black dot (Figure 49). The values for the current residue across the entire data set are sorted in the far right portion of the main display with the current plot labeled and highlighted (Figure 44(f)). Between the enlarged sub-region and the right-hand sorting display, the current residue and the previously examined residue are each sorted across the dataset in columns (Figure 44(e)). Beside each column, the current and previous residue for the plot being examined are marked with green tick marks. The sorted comparison columns were added to help users track their path of exploration and to compare multiple residues across the entire data set. Although the implementation only shows the current residue and the previously visited residue, this could be easily extended to include less recently examined residues enabling longer exploration paths to be kept.

## 5.4 Subspace Clusters

When the plot in the detail window has the grid drawn on it, users can pick the size of the subspace and then click on the plot section to select it. Once the grid section

has been selected and plots having interesting subspaces placed on the clipboard, the user toggles the view to "Off" to go to the next view. Boundary coordinates of the selected sub-region are captured and the plots of interest are propagated to the next view. Before users are presented with the sliding window, an MDS [9], [14] display is shown. In the previous view, histograms are sorted according to bin similarity. The MDS view is a two-dimensional representation of the subsapces consistent with other views in the system and may reveal overall patterns not visible in the one-dimensional histogram sort. As mentioned earlier, alternative methods can be explored. The MDS display in the subspace workflow has the same layout as the MDS display in the original framework. However, the distances calculated, the coordinates determined, and the data within each plot shown only consider the plot section selected in the grid. Any general patterns for the entire subspace section that were not detected while viewing the histogram layout can be examined here.

## 5.5 Sliding Subspaces

### 5.5.1 Context Scans

Once a subspace has been located it should be placed in context with adjacent subspaces. However, simply comparing the single subspace with others is not enough because proteins are continuous in structure where parameter effects may be sudden or evolve gradually through several subspaces. Knowing how a subspace gradually merges with other adjacent subspaces will allow scientists to more finely characterize the effects of parameters sets. This includes both finding the magnitude of differences from other subspaces and better pinpointing where those differences occur. To accom-

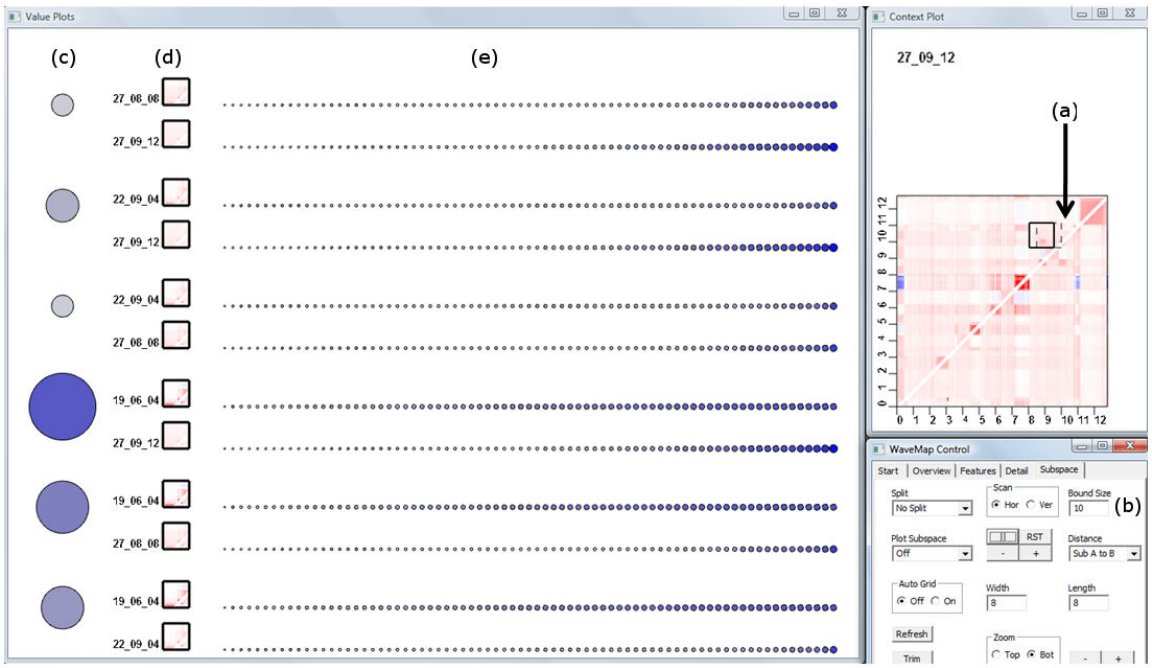



Figure 50: (a) A sliding window the size entered by the user at (b). (c) Bubbles represent pair-wise distances for clipboard items. (d) Raw data points for the subspace pairs. (e) Sorted bubbles relating distances from each subspace on the clipboard to subspaces across the entire data set.

plish this, domain scientists conveyed an interest in a mechanism where the bounds of a window could gradually scan across adjacent subspaces revealing distances among items as that window moved from one subspace into the selected subspace. What follows below is a description of the resulting design.

After viewing overall patterns in the cluster view the user can right-click the detail window to begin the sliding window (Figure 50(a)). The window's coordinates and size are the same as the subspace selected in the grid. After initiating the window, a dotted bounding box appears to the left of the subspace. The window's distance from the base is half of the bounding box size entered by the user (Figure 50(b)). Changing the bounding box size allows scientists to alter the space of analysis for which they desire to measure parameter effects.

When the scan begins, the sliding window moves right in increments of one residue at a time. The user can control window movement through buttons that cause the scan to stop, to increment by one, or to decrement by one. Because scientists may want to also examine subspaces that are vertically adjacent, users can change the direction of the scan through a toggle button. Being able to change the direction has a practical use for domain analysts. The different implications of direction choice for the allosteric response data is described below. To simplify the descriptions, they will only consider the scan of a single row or column for the length of the protein. However, the following descriptions can be generalized for larger sub-regions.

- **Horizontal scan** Scanning horizontally allows scientists to monitor the effects of parameters for a single residue as all residues are individually perturbed.

- **Vertical scan** Scanning vertically reveals changes in parameter effects for the perturbation of a single residue across all residues.

As the window moves horizontally or vertically, the differences in the effects of parameters are output to the user. Distances between subspaces are encoded as bubbles. As the scan progresses, the bubble area and the alpha value are altered as distances change. When distances are small, bubbles are small and gray. Large distances result in larger bubbles and deeper shades of blue. The screen space is divided into two main sections. On the far left (Figure 50(c)), bubbles for the pair of subspaces next to it (Figure 50(d)) are drawn in a larger scale. Each bubble on the left represents the distance between one of the pair-wise comparisons made from the clipboard contents. To the right of the raw values for each subspace, is a row of

bubbles increasing in size (Figure 50(e)). Each row of bubbles represents the distance from the subspace at the beginning of the row to every corresponding subspace across the entire data set.

Functionality exists to help users compare the overall scan. The system records the bubble sizes in each row at the beginning of the scan, at the original window, and at the end of the scan. These snapshots are automatically organized by the system according to subspace and are available to the user after one pass of the sliding window. As will be shown later, this organization can be useful for comparing changes occurring between time points among several subspaces.

### 5.5.2 Interactions

The user has several interactions to reveal more detailed information when the scan is paused. When any bubble (including the enlarged bubbles on the left or the smaller bubbles to the right) is selected by a mouse-over, the normalized distance is shown at the top of the screen within a rectangle that matches the corresponding bubble color (Figure 51(a)). On either side of the rectangle are the plot names for which that distance was calculated. Additionally, green rectangles enclose one circle on each row (Figure 51(b)). The green rectangle indicates where the distance represented by the enlarged bubble on the left ranks when the entire data set is sorted. Finally, users have access to a free-hand tool if the current size or location of the selected subspace is inadequate. To use this tool, the upper-left corner of the new bounding box is selected by clicking in the detail window and then releasing the mouse at the desired bottom-right corner (Figure 52).

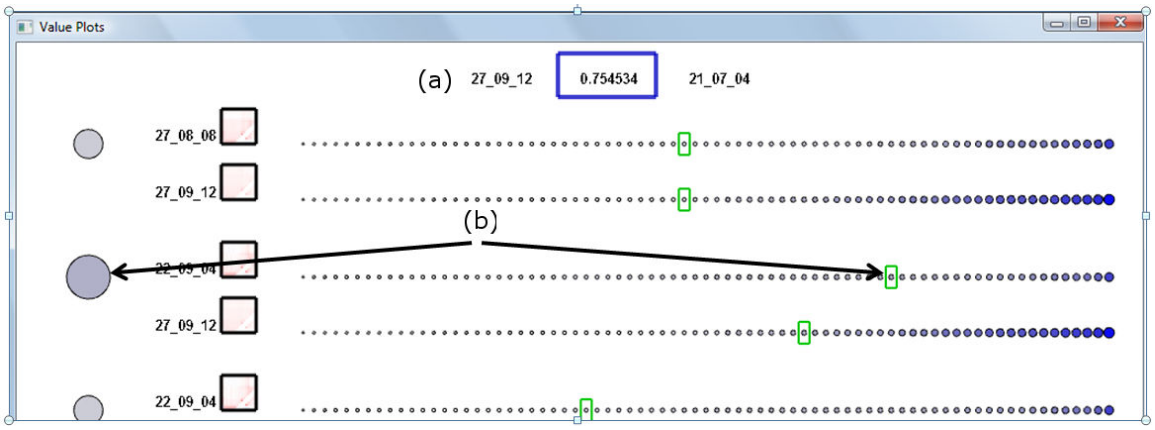



Figure 51: (a) Moving the mouse over a bubble causes the distance and plot names to be displayed within a rectangle in the same color as the buble. (b) Green rectangles show where the subspace at the left ranks when the all distances to that subspace are sorted.

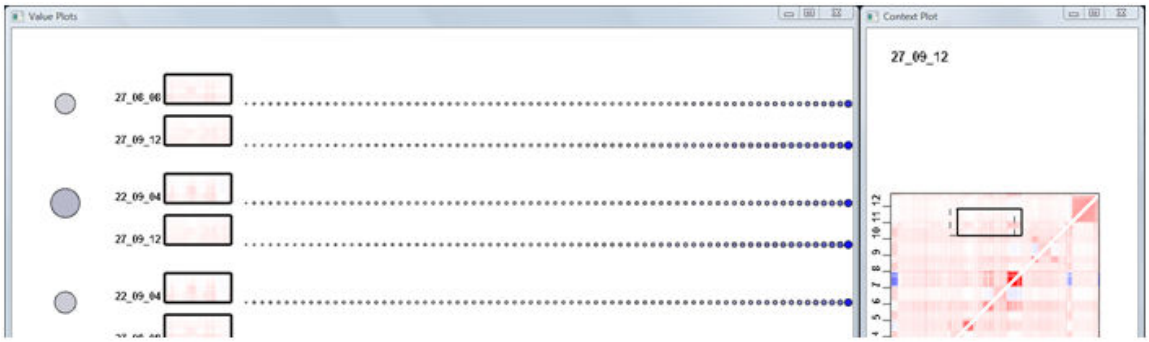



Figure 52: Subspaces can be selected by window-selecting the desired area.

### 5.5.3 Outlier Detection

Before this view emerged there was one attempt that was not suitable for detailed examination. This view, shown in Figure 53, split the display space along the screen's diagonal. The lower right portion contains enlarged distance bubbles. In the top portion is one-half of the symmetrical distance matrix. Bubble size and color represents each pair-wise similarity between two subspace regions as in the previous view. Even though the views contained the same basic components, the split-screen view resulted in the crowding of the distances across the data set. Additionally, the fixed rows and

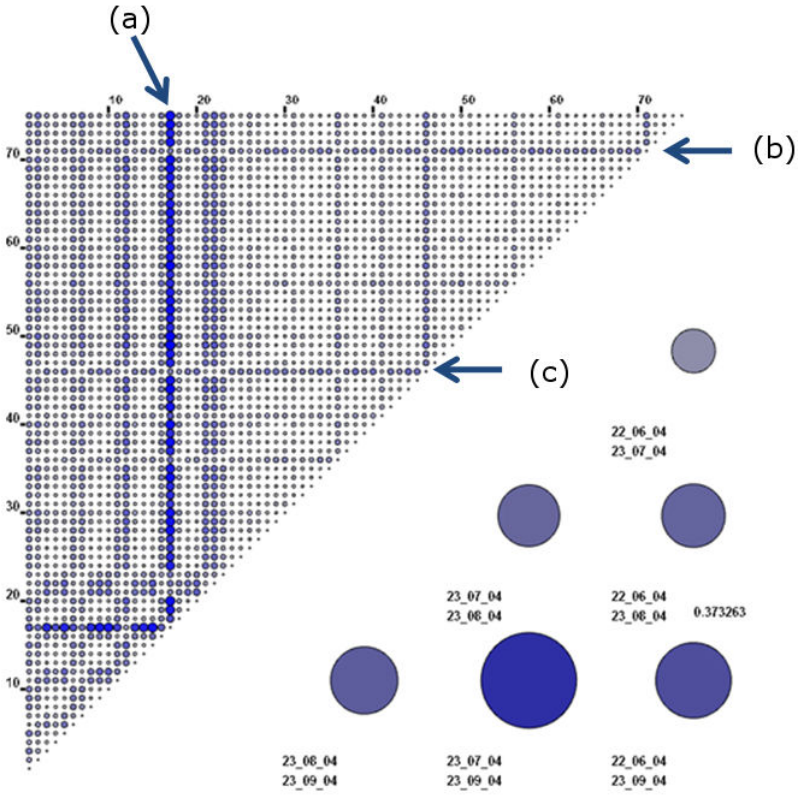



Figure 53: Each row and column in the top-left portion of the screen represent different plots. Each bubble relates the dissimilarity of subspaces. In the bottom-right, bubbles for plots of interest on the clipboard are drawn larger. (a) An outlier is identified as bubbles in this column are large across the data set. (b) and (c) point to rows having bubbles of various sizes, the majority of which are large bubbles. This indicates an outlier that is less extreme than that in (a).

columns were arranged in the order that they were input from storage which made patterns difficult to detect. However, this view is useful for easily identifying subspace outliers as scans are performed. The previous view only displays a limited number of subspaces but this expanded version can help users monitor the entire data set for outliers. A user can easily monitor this portion of the screen to see if a row (or column) develops only large bubbles, an indication that an outlier has emerged. Furthermore, the user can stop the scan to see exactly where the behavior started that resulted in outlier status.

### 5.5.4 Subspace Similarity Types

To help quantify the differences in parameter effects, two distance measures are available in the bubble view (Figure 54). The default distances for pair-wise comparisons are simply the covariances used in the cluster view. However, scientists may need to track if the distances between two corresponding subspaces is different from the distance calculated from the entire plots. In other words, is the similarity of the subspaces different from the similarity calculated when the entirety of each plot is considered? This difference is conveyed through changing bubble size and color *B(a,b)* according to the absolute value of the difference between the distance *E(a,b)* (entire matrix A to entire matrix B) and a second distance *S(a,b)* (sub-region in matrix A

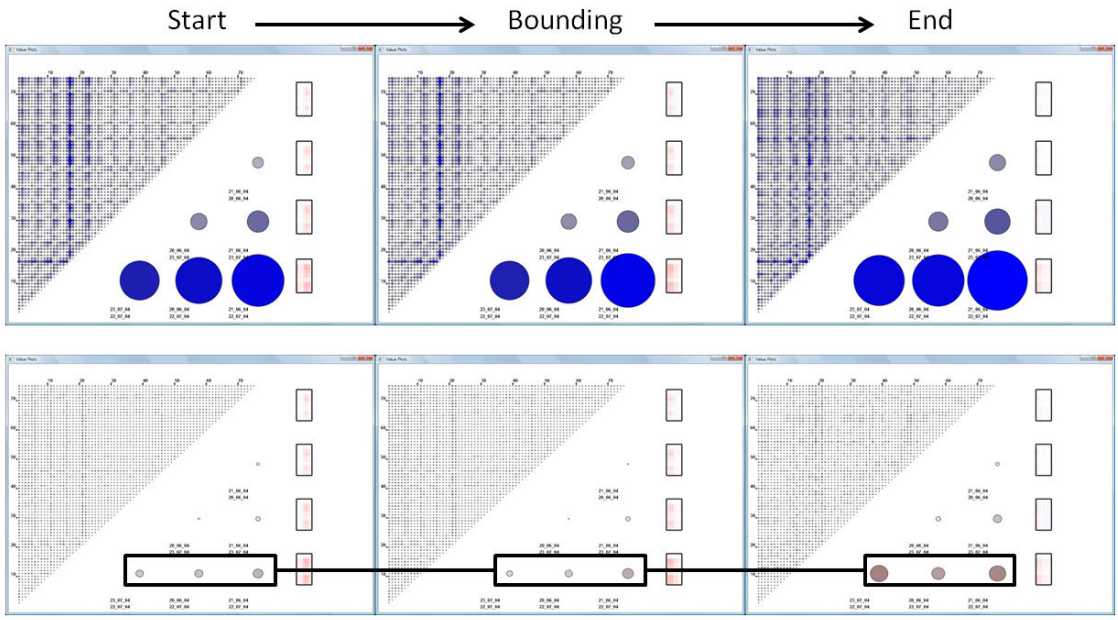



Figure 54: Beginning, middle, and end samples of an animated scan. Top: Bubbles relate the similarity of corresponding subspaces. There is a gradual, but noticeable increase in bubble size indicating a decrease in similarity. Bottom: Bubbles relate the difference in the similarities when comparing subspaces and when comparing entire plots. A larger, deeper red bubble indicates that the subspaces exhibit a different degree of similarity than the plots as whole entities.

to the corresponding sub-region in matrix B) as shown in Equation (3).

$$B(a,b) = |E(a,b) - S(a,b)| \tag{3}$$

A small difference indicates that comparing subspaces for two plots follow the same general behavior as if the entire plots were compared. A large difference indicates that subspace behavior deviates from the comparative behavior of the two plots as whole entities. A large difference flags scientists to an area that may lead to insight not possible when entire plots are considered.

## 5.6 Evaluation

### 5.6.1 Example Use

An example now illustrates how some of the above functions are able to aid scientists in exploring subspace data sets. Mottonen et al [58] found that allosteric response is both conserved and variable across the CheY protein. To investigate this further using the added subspace functions, each plot is divided into eight grid sections along both the width and length. In this case, the grid is browsed systematically and sections will be investigated individually for their consistency. Figure 55 displays one subspace that was visited and exhibits a high degree of consistency. In the top of Figure 55 the original data values are shown. The histogram for each subspace is shown on the bottom. From both the top and the bottom views, there is little variability throughout the data set. Most subspaces are flexible (red). Sorting makes it clear that there are only a couple sections that have any rigid sections (blue). This is also evident in the histogram view. After the browsing is finished, it would be

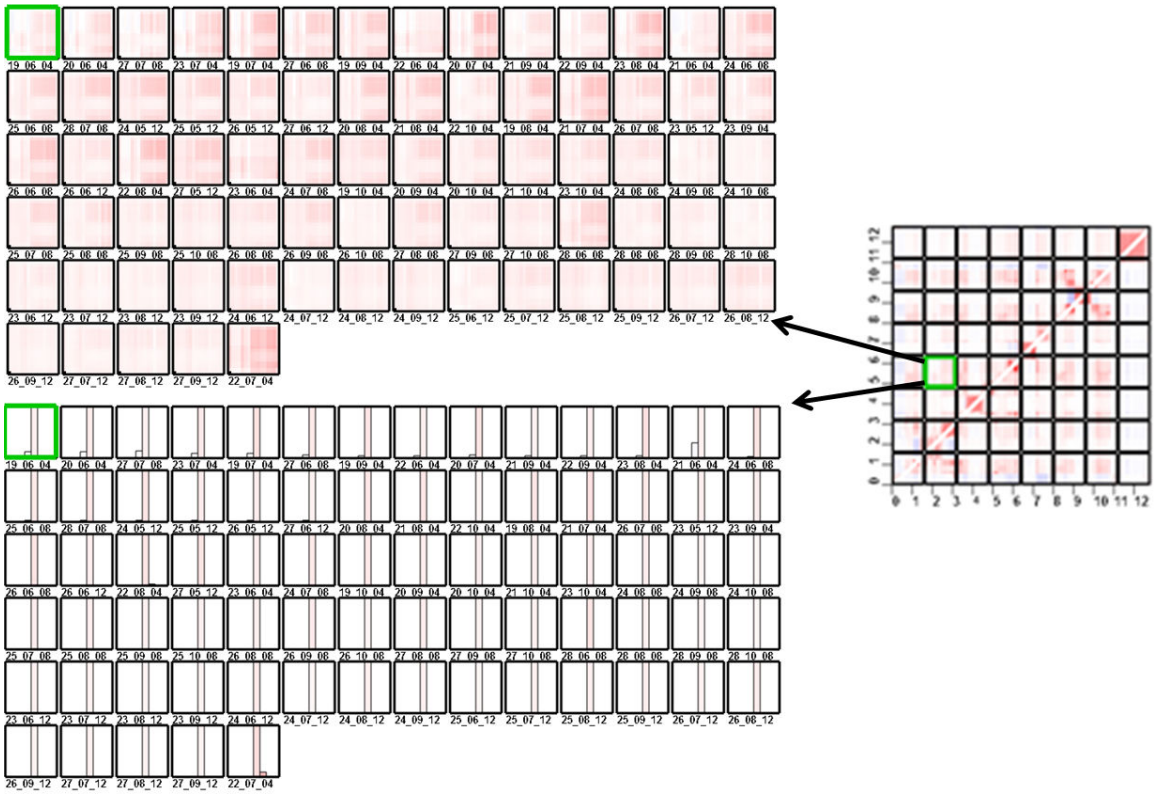


Figure 55: The original values and histograms for the subspace marked in green on the right. Across the data set, this subspace exhibits great consistency. The only significant deviations are a small number of blue residues in the first row. (Green borders on the top-left and bottom-left indicate the current plot selected and not the particular sub-region selected on the right.)

interesting to re-visit the few variable sections (those with blue) and investigate what causes the difference in flexibility.

Another subspace visited is shown in Figure 56. Again, the original data values and the histograms are displayed. This set of subspaces can be roughly segregated into two groups. One is consistently flexible (red) and the other has a mix of rigid (blue) and flexible (red) residues. Viewing the histograms confirms that observation. Although there is consistency in the red group, the mixed group seems a little more variable in both the degree of flexibility (and rigidity) and the differences in the total number of residues exhibiting flexible or rigid behavior. This is also confirmed in

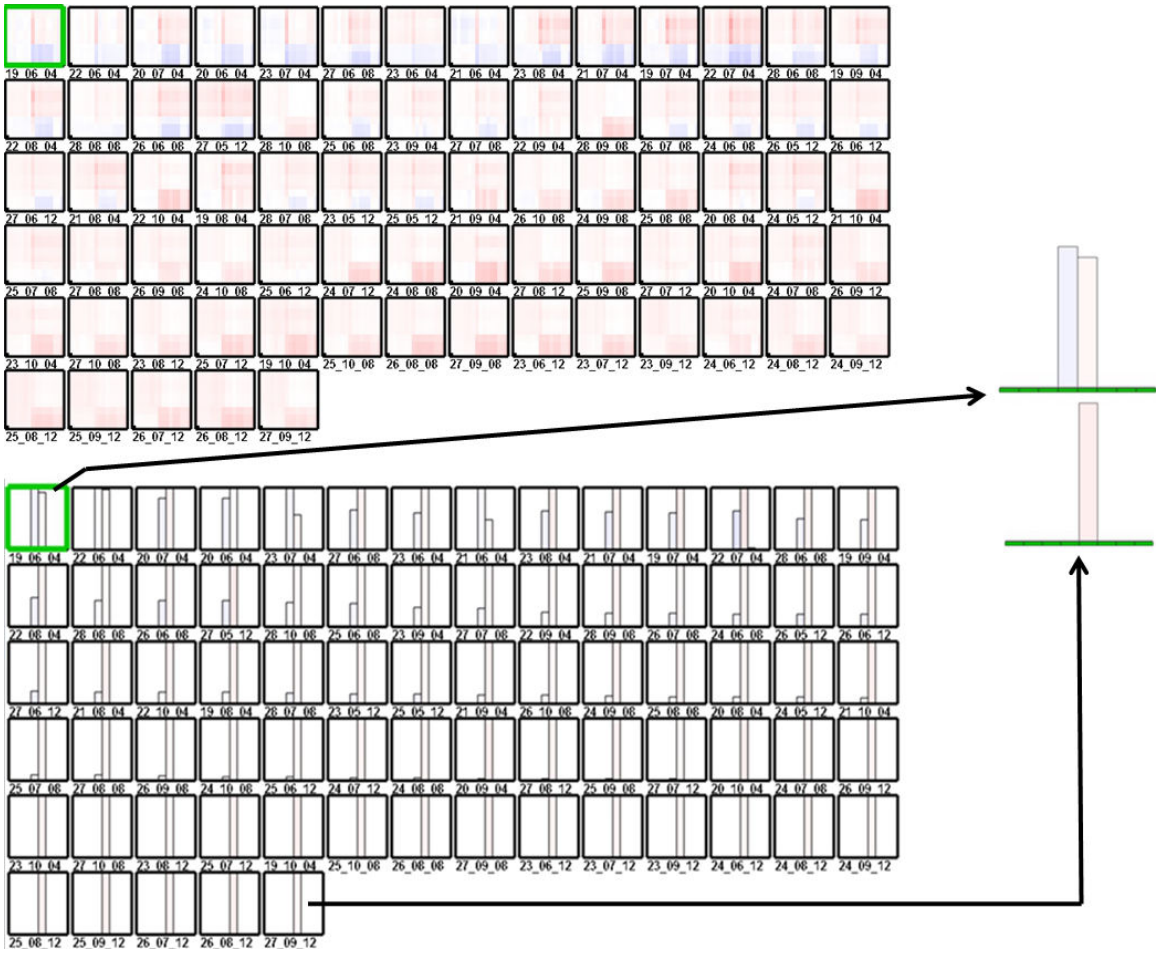


Figure 56: A more variable plot subsection. Histograms for the extreme plots are shown on the right.

the histograms. The top left plot has a rigid bin having more members than the flexible bin. Additionally, the rigid bin has an average value that is more rigid than the others. However, the rigid bin generally becomes smaller and more faded as the sequence of histograms progresses towards the end of the list. At the end of the list, there are no rigid residues.

To adequately examine the effect of parameters in this subspace, the surrounding sections should be included in the analysis. Two subspaces from the mixed (variable) group and two plots from the consistently flexible group are placed on the clipboard.

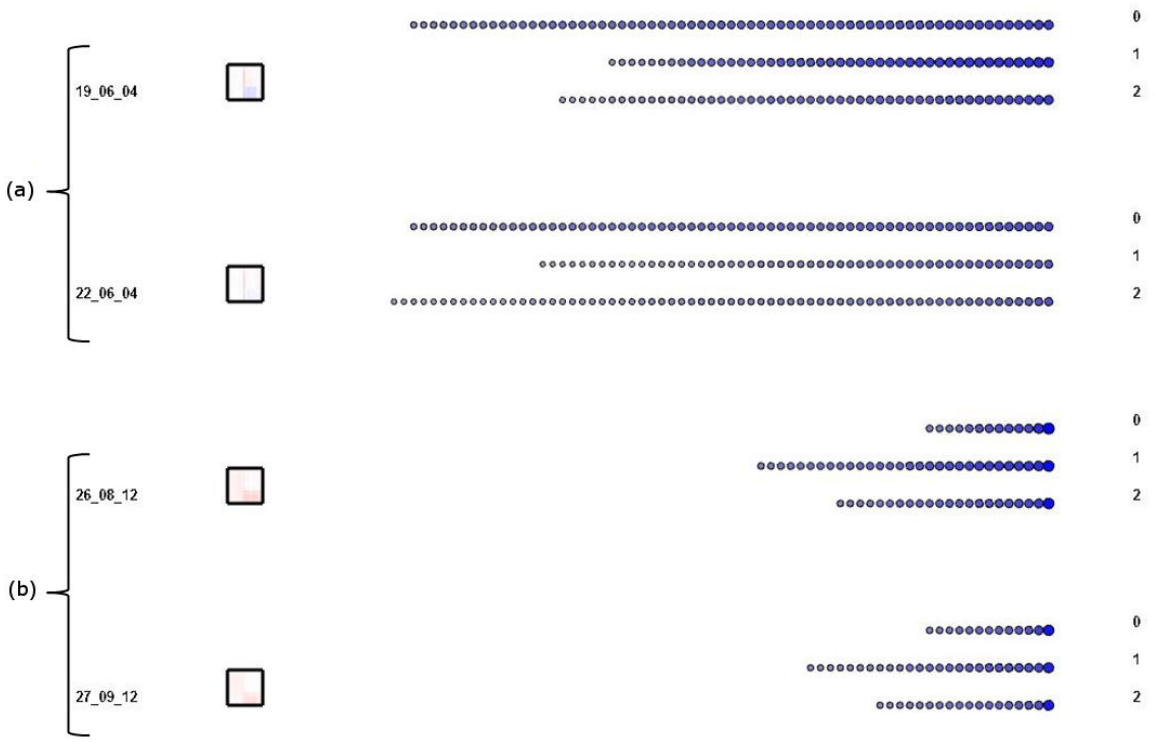



Figure 57: Resulting distances (bubbles) after a scan is initiated organized by plot. Three places in the scan were recorded: the beginning of the scan (0), the original window (1), and the end of the scan (2). As the scans progress, differences across the data set become evident and allow adjacent subspaces to be included.

By investigating the extremities of the sorted list, the entire range of parameter differences should be evident. To get a wider view of the adjacent subspaces to the left and right of the selected region, 50 is chosen as the window size (each subspace is only 16 residues wide). The scan is started and the changes are mostly gradual. To better visualize the gradual changes, bubble sizes in each row are recorded by the system at the beginning of the scan, at the original window, and at the end of the scan. The three points in the scan are shown in Figure 57. The system organizes each recording by plot after one complete scan. In the figure, line 0 is the beginning, line 1 is the original window, and line 2 is the end of the scan. The more rigid (mixed) sub-regions are in Figure 57(a) and the more consistent sections are in 57(b).

The extent of the parameter differences can now be examined so that they can be placed in context of adjacent sections. In this case, only bubbles having a size greater than half of the largest bubble in that row are drawn. This helps filter subspaces that are similar to the target and more clearly show the subspaces that are less similar. In Figure 57(a), the number of subspaces drawn decreases as the original window is reached and then increases again as the bounding box passes through to the other side. *For scientists, this case means that, in relation to the selected plot, the difference in parameter effects is more narrow in the original window than in adjacent sections.* For the more consistent subspaces on the bottom, the opposite occurs. *For scientists, this means that, in relation to the selected plot, the difference in the effects of parameters is evident across more subspaces in the original window than in adjacent sub-regions.* Also clear in this view are general characteristics of the two groups. The subspaces in 57(a) have more bubbles in each row than those in 57(b). This indicates that the bottom subspaces have more overall similarity to the rest of the data set than the top subspaces.

In the example here, the changes are slow during the scan indicating that the changes are gradual. To more accurately pinpoint any sudden change, the grid size and section can be altered or the free-hand tool employed. Because the user has identified the behavior at the beginning, middle, and end of the scan, the window size can also be narrowed to either the first part of the scan or the latter part of the scan. Which part of the scan is investigated further depends on the behavior (increasing or decreasing similarity) that is desired. This can be repeated until the residue-level differences are found. Another option for exploration is to move the mouse over plots

having larger distances, record the plot names, and then explore them for residue-level variations.

### 5.6.2 Expert Evaluation

The subspace exploration functionality was shown to the domain scientists. The scientists expressed interest in the histogram view by mentioning that an approach tried earlier by a graduate student in their group involved heavy use of histograms and binning. Scientists had many questions regarding the determination of bin size, both in terms of height and width. Any future use would need to investigate exactly what normalization procedures (per subspace, over the entire data set, etc.) would create the best environment for comparison.

The domain scientists found the bubble view shown in Figure 50 and Figure 51 confusing at first. When all of the subspace additions were demonstrated, explaining this view took the most time. Scientists had trouble conceptualizing the distances among pair-wise comparisons. In other words, it seemed as though they needed to mentally convert the layout to the more symmetrical distance matrix before interpreting the results. However, they stated that the view was compact and showed more information in a way they had not considered before. They showed interest in investigating the data with the new information available.

A few changes were made to simplify the bubble view. Previously, each large bubble on the left was followed by two subspaces and two lines of bubbles representing the distance from each subspace to all other corresponding subspaces. Now, each row only consists of the subspace and its label followed by its distance to the other cor-

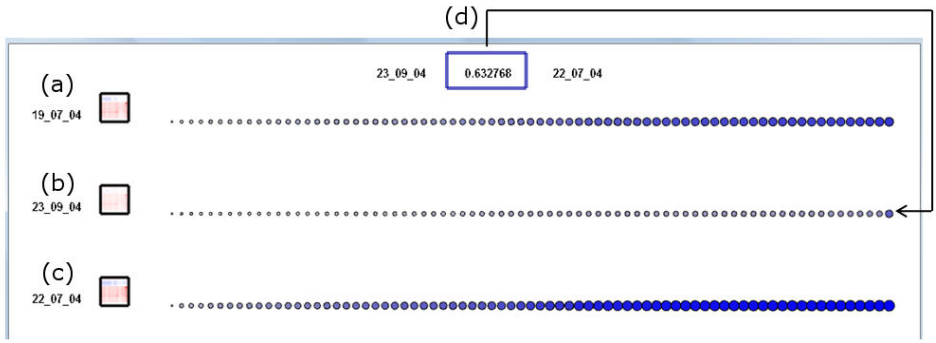



Figure 58: Subspaces that are (a) similar to some but dissimilar to others, (b) similar to all other subspaces except for one, and (c) dissimilar to almost all others. (d) Detail information shown when the scan is stopped and the mouse moves over the last bubble in row (b). The last bubble in row (b) is the distance to the outlying subspace in row (c).

responding subspaces across the data set. The green rectangles were also eliminated when a user stops the animation. The double entries were eliminated and the large bubbles on the left were also removed. The updated view is shown in Figure 58.

The updated version streamlined the previous version with little information loss. Because domain scientists expressed an interest in some of the information in the older view, they now have access to both. The new version can be viewed for a less cluttered overview of a few interesting subspaces and then users can move to the advanced view for more detail.

After scientists began to understand the layout, there were two noteworthy observations. First, the allosetric response data was used in the demonstration and scientists expressed a desire for a very specific subspace definition. Scientists were more interested in setting the subspaces to single columns. Although it wasn't surprising that scientists were interested in defining the subspaces as columns, it was somewhat surprising that little interest was shown in subspaces of different shapes

and sizes. In future attempts, this will be taken into account so that response vectors can be used as the base unit of comparison. During development a mechanism for clustering columns had been implemented. This implementation clustered columns within a single plot. However, the interest in column-based analysis expressed by scientists included clustering the entire data set according to weighted columns of interest. Second, domain scientists stated that the bubble view introduced them to information that they had not even considered before. More exploration would be needed to see how this new information fit into their analysis. There was agreement that the bubble view could provide useful information but quickly become unmanageable as subspaces are added.

There was interest from domain scientists in at least two directions for further work. First, more guidance should be given when searching for interesting subspaces, regardless of shape and size. If domain analysts have no beginning suspicions about where to start exploration or the wavelet functionalities prove fruitless, finding a subspace of interest is a purely exploratory process. Much time can be wasted finding the subspace of interest. Second, there was agreement from everyone that alternate clustering configurations which better conveyed distances between column response vectors would be beneficial. A new configuration should include an evolving picture of the data set and allow scientists to move from column to column more fluidly without having to click on the grid, an action that becomes increasingly difficult in the current implementation as columns become more narrow.

# CHAPTER 6: CONCLUSION AND FUTURE WORK

This work addresses the challenges encountered when analyzing outputs from a protein flexibility model. Providing solutions to the analysis of complex computational data sets will speed advances in predicting and treating disease. Data commonly encountered by users of this model has been categorized and described so that overlap with other scientists can be identified and used in the construction of more standardized processing tools. Frequent communication with domain scientists, the study of multiple data sets, and several attempts at prototyping have resulted in a set of high-level tasks so that better analysis tools can be developed in the future. A prototype incorporating feature extraction, visualizations, and interactions emerged from the formalization of data types and high-level tasks. A user study, user scenarios, and feedback illustrate the prototype's effectiveness in locating and exploring areas of interest in a large, complex data set.

There are many directions for future work in addition to addressing the shortcomings already discussed. For example, experimentation with feature extraction should continue. Custom wavelets and even other feature extraction techniques could be integrated in the system. Giving the user a wide range of options for the type of feature extraction and parameter options for the chosen method would allow scientists to fine tune which characteristics are extracted. Expanding this idea would include

finding ways to effectively compare the effects of different extraction techniques over an entire data set.

Another direction for future work is searching for better ways to associate input parameters with model results. Most of the effort here concentrated on discriminating between model outputs represented by plots. Options were included for displaying plot names and border highlighting which reflect only a few inputs of interest to the domain scientist. In the future, model outputs and extracted features should be more closely bound to all model parameters, the list of which could be pruned by the user. Access to line plots and scalar values of all input variables, to name just a couple, can be integrated into the system so that change in outputs can be traced back to the single or group of responsible parameters. Extensions to this line of research include providing tools for what-if analysis when changing model inputs.

Finally, the pipeline and prototype in this work should be applied to other domains. The possibilities for applying this approach to other problems encountered when analyzing biological data was discussed. Experimentation should be conducted to find any differences in the data processing or high-level needs of scientists. Additionally, applications of the tool to domains beyond biological data sets can be explored.

# REFERENCES


[1] Anandhavalli, M., Mishra, C., and Ghose, M. Analysis of microarray image spots intensity: A comparative study. *International Journal of Computer Theory and Engineering 1*, 5 (December 2009), 1793–8201.

[2] Assent, I., Krieger, R., Muller, E., and Seidl, T. Visa: Visual subspace clustering analysis. *SIGKDD Explorations 9*, 2 (2007), 5–12.

[3] Barlowe, S., Liu, Y., Yang, J., Livesay, D. R., Jacobs, D., Mottonen, J., and Verma, D. Wavemap: Interactively discovering features from protein flexibility matrices using wavelet-based visual analytics. *Computer Graphics Forum 30* (2011), 1001–1010.

[4] Barlowe, S., Zhang, T., Liu, Y., Yang, J., and Jacobs, D. Multivariate visual explanation for high dimensional datasets. *IEEE Symposium on Visual Analytics Science and Technology* (2008), 147–154.

[5] Barlowe, S. A., and Tashakkori, R. A lifting-based knowledge discovery in microarray data. *International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences* (2005), 291–298.

[6] Bernas, T., Asem, E. K., Robinson, J. P., and Rajwa, B. Quadratic form: A robust metric for quantitative comparison of flow cytometric histograms. *Cytometry A. 73*, 8 (2008), 715–26.

[7] Biochem.co - biochem & science notes. `http://biochem.co/2008/08/proteins-primary-secondary-structures/`.

[8] Bonneau, G.-P. An introduction to wavelets for scientific visualization. *Scientific Visualization Conference (dagstuhl '97).* (1997), 16.

[9] Borg, I., and Groenen, P. *Multidimensional Scaling: Theory and Applications.* Springer, 2009.

[10] Bremm, S., Schreck, T., Boba, P., Held, S., and Hamacher, K. Computing and visually analyzing mutual information in molecular co-evolution. *BMC Bioinformatics 11*, 1 (2010).

[11] Cai, W., and Adjouadi, M. Minimization of boundary artifacts on scalable image compression using symmetric-extended wavelet transform. *International Conference on Information Technology: Coding and Computing 1* (2004), 598.

[12] Camoglu, O., Kahveci, T., and Singh, A. K. Psi: indexing protein structures for fast similarity search. *Bioinformatics 19* (2003), i81–i83.

[13] Chandola, V., Baneriee, A., and Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv 41*, 3 (2009), 1–58.

[14] Cox, T., and Cox, M. *Multidimensional Scaling.*, second ed. Chapman and Hall, 2000.

[15] DeLano Scientific. *The PyMOL Molecular Graphics System*, 2002.

[16] Do, C. B., Mahabhashyam, M. S., Brudno, B., and Batzoglou, S. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Research. 15* (2005), 330–340.

[17] Dunteman, G. H. *Principal Components Analysis.* Sage Publications, 1989.

[18] Elshenawy, L. M., Yin, S., s. Naik, A., and Ding., S. X. Efficient recursive principal component analysis algorithms for process monitoring. *Industrial & Engineering Chemistry Research. 49* (2010), 252–259.

[19] Farin, G., Hamann, B., and Hagen, H. *Hierarchical and Geometric Methods in Scientific Visualizations.* Springer, 2003.

[20] Ferdosi, B. J., Buddelmeijer, H., Trager, S., Wilkinson, M. H., and Roerdink, J. B. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. *Proc. IEEE Conference on Visual Analytics Science and Technology* (October 2010), p. 35–42.

[21] Ferdosi, B. J., and Roerdink, J. B. Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis. *Eurographics/IEEE Symposium on Visualization 30*, 3 (2011), 1121–1130.

[22] Forlines, C., and Balakrishnan, R. Improving visual search with image segmentation. *CHI* (2009), 1093–1102.

[23] Gehlenborg, N., O'Donoghue, S., Baliga, N., Goesmann, A., Hibbs, M., Kitano, H., Kohlbacher, O., Neuweger, H., Schneider, R., Tenenbaum, D., and Gaviin., A. Visualization of omics data for systems biology. *Nature Methods. 7* (2010), S56–S68.

[24] Gonzalez, R., and Woods, R. *Digital Image Processing.* Prentice Hall, 2001.

[25] Guo, D. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization 2* (2003), 232–246.

[26] Guo, Z., Ward, M. O., and Rundensteiner, E. A. Model space visualization for multivariate linear trend discovery. *VAST* (2009), 75–82.

[27] Gutierrez, A., and Somolinos, A. Influence of wavelet boundary conditions on the classification of biological signals. *Proceedings of the IEEE 26th Annual Northeast Bioengineering Confernece* (2000), 25–26.

[28] Guyon, F., Tuffery, P., and Camproux, C. A hidden markov model applied to the protein protein 3d structure analysis. *Source Computatonal Statistics Data Analysis Archive 52*, 6 (2008), 3198–3207.

[29] Hao, M. C., Dayal, U., Keim, D. A., and Schreck, T. Importance-driven visualization layouts for large time series data. *IEEE Symposium on Information Visualization* (2005), 27.

[30] Hart, S. G., and Staveland, L. E. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds. North Holland Press, 1988, pp. 239–250.

[31] Hess, B. Convergence of sampling in protein simulations. *Physical Review E 65*, 3 (March 2002), 031910.

[32] Hu, J., Shen, X., Shao, Y., Bystroff, C., and Zaki, M. J. Mining protein contact maps. *Workshop on Data Mining in Bioinformatics* (2002), 3–10.

[33] Huitema, H., and Liere, R. V. Interactive visualization of protein dynamics. *Visualization 2000* (2000), 465–468.

[34] Jacobs, D., and Dallakyan, S. Elucidating protein thermodynamics from the three-dimensional structure of the native state using network rigidity. *Biophys J 88* (2005), 903–915.

[35] Jadhao, D. V., and Holambe, R. S. Feature extraction and dimensionality reduction using radon and fourier transforms with application to face recognition. *International Conference on Computational Intelligence and Multimedia Applications. 2* (2007), 256–260.

[36] Jagielska, A., Wroblewska, L., and Skolnick, J. Protein model refinement using an optimized physics-based all-atom force field. *Proceedings of the National Academy of Sciences of the United States of America 105*, 24 (2008), 8268–8273.

[37] Jensen, A., and Cour-Harbo, A. L. *Ripples in Mathematics.* Springer, 2001.

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

[39] Jr., F. F., Lopes, C. E., de Melo, R. C., Santoro, M. M., Carceroni, R. L., Jr., W. M., de A. Araujo, A., and Silveira, C. H. An image-matching approach to protein similarity analysis. *Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing* (2004), 17–24.

[40] Keim, D. A., and Oelke, D. Literature fingerprinting: A new method for visual literary analysis. *IEEE Symposium on Visual Analytics Science and Technology* (2007), 115–122.

[41] Keim, D. A., Oelke, D., Truman, R., and Neuhaus, K. Finding correlations in functionally equivalent proteins by integrating automated and visual data exploration. *Sixth IEEE Symposium on Bioinformatics and BioEngineering (BIBE '06)* (2006), 183–192.

[42] Keskin, O., Jernigan, R., and Bahar, I. Proteins with similar architecture exhibit similar large-scale dynamic behavoir. *Biophys J 78* (2000), 2093–2106.

[43] Kondekar, V., Kolkure, V., Sodal, G., and Mudegaonkar, J. Image retrieval techniques based on image features: a state of art approach for cbir. *International Conference & Workshop on Emerging Trends in Technology. 2010* (2010), 998–999.

[44] Li, S., Liao, C., and Kwok, J. Wavelet-based feature extraction for microarray data classification. *International Joint Conference on Neural Networks* (October 2006), 5028–5033.

[45] Li, T., Li, Q., Zhu, S., and Ogihara, M. A survey on wavelet applications in data mining. *SIGKDD Exploration 4* (2002), 49–68.

[46] Liu, D., Gao, Q., Wang, H. H., and Zhang, J. A web-based interactive data visualization system for outlier subspace analysis. *SEDE* (2010), 275–280.

[47] Liu, Y. Wavelet feature extraction for high-dimensional microarray data. *Neurocomputing 72* (2009), 985–990.

[48] Livesay, D., Dallakyan, S., Wood, G., and Jacobs, D. A flexible approach for understanding protein stability. *FEBS Lett 576* (2004), 468–476.

[49] Livesay, D., and Jacobs, D. Conserved quantitative stability/flexibility relationships (qsfr) in an orthologous rnase h pair. *PROTEINS: Structure, Function, and Bioinformatics 62* (2006), 130–143.

[50] Livesay, D. R. The distance constraint model. Lecture slides for Biophysical Modeling, University of North Carolina at Charlotte, Fall 2008.

[51] Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., William Cleveland, S. G., and Ebert, D. A visual analytics approach to understanding spatiotemporal hotspots. *IEEE Transactions on Visualization and Computer Graphics 16*, 2 (2010), 205–220.

[52] Masso, M. Improving prediction accuracy via subspace modeling in a statistical geometry based computational protein mutagenesis. *International Journal of Knowledge Discovery in Bioinformatics 1*, 4 (2010), 54–68.

[53] Mathworks. `http://www.mathworks.com`.

[54] Merino, C. S., Sips, M., Keim, D. A., Panse, C., and Spense, R. Task-at-hand interface for change detection in stock market data. *AVI* (2006), 420–427.

[55] Miller, N., Wong, P. C., Brewster, M., and Foote, H. Topic islandsa wavelet-based text visualization system. *Proceedings of the 9th IEEE Conference on Visualization* (1998), 189–196.

[56] Miwa, M., Saetre, R., Miyao, Y., and Tsujii, J. A rich feature vector for protein-protein interaction extraction from multiple corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (2009), 121–130.

[57] Morrison, J., Breitling, R., Higham, D., and Gilbert, D. A lock-and-key model for protein-protein interactions. *Bioinformatics, 22 16* (2006), 2012–2019.

[58] Mottonen, J., Jacobs, D., and Livesay, D. Allosteric response is both conserved and variable across three chey orthologs. *Biophysical Journal 99*, 7 (2010), 2245–54.

[59] Mottonen, J., Xu, M., Jacobs, D., and Livesay, D. Unifying mechanical and thermodynamic descriptions across the thioredoxin protein family. *PROTEINS: Structure, Function, and Bioinformatics 75*, 3 (2009), 610–627.

[60] Muller, M. J. Participatory design: The third space in hci, handbook of hci. In *Human-Computer Interaction: Development Process*, J. Jacko and A. Sears, Eds. Erlbaum, 2008.

[61] Nanni, L. Comparison among feature extraction methods for hiv-1 protease cleavage site predicition. *Pattern Recognition* (2006), 711–713.

[62] National center for biotechnology information. `http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html`.

[63] National human genome research institute. `http://www.genome.gov/Glossary/`.

[64] Needleman, S. B., and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology. 48* (1970), 443–453.

[65] Neshich, G., Rocchia, W., Mancini, A., Yamagishi, M., Kuser, P., Fileto, R., Baudet, C., Pinto, I., Montagner, A., Palandrani, J., Krauchenco, J., Torres, R., Souza, S., Togawa, R., and Higa, R. Javaprotein dossier: A novel web-based data visualization tool for comprehensive analysis of protein structure. *Nucleic Acids Research 32* (2004).

[66] Nigham, A., Hsu, D., and Latombe, J.-C. Characterizing protein conformation space. *Singapore and MIT Alliance (SMA) Symposium* (2007).

[67] Nowell, L., Hetzler, E., and Tanasse, T. Change blindness in information visualization: A case study. *Proceedings of the IEEE Symposium on Information Visualization .* (2001), 15–22.

[68] Oelke, D., Bak, P., Keim, D. A., Last, M., and Danon, G. Visual evaluation of text features for document summarization and analysis. *IEEE Symposium on Visual Analytics Science and Technology* (2008), 75–82.

[69] Patil, A., Kolhe, S., and Patil, P. 2d face recognition techniques: A survey. *International Journal of Machine Intelligence. 2* (2010), 74–83.

[70] Pei, J. Multiple protein sequence alignment. *Current Opinion in Structural Biology. 18* (2008), 382–386.

[71] Polharkar, S., and Reddy, C. K. Identifying information-rich subspace trends in high-dimensional data. *SIAM* (2009), 557–568.

[72] Potzsch, S., Scheuermann, G., Wolfinger, M., Flamm, C., and Stadler, P. Visualization of lattice-based protein folding simulations. *Proceedings of the Information Visualization* (2006), 89–94.

[73] Qazi, S., Panetta, K., and Agaian, S. Detection and comparison of color edges via median based pca. *IEEE International Conference on Systems, Man and Cybernetics, 2008.* (2008), 702–706.

[74] Rasmussen, C. E., de la Cruz, B. J., Ghahramani, Z., and Wild, D. L. Modeling and visualizing uncertainty in gene expression clusters using dirichlet process mixtures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 6*, 4 (October-December 2009), 615–628.

[75] Rubner, Y., Thomasi, C., and Guibas, L. J. The earth movers distance as a metric for image retrieval. *International Journal of Computer Vision. 40* (2000), 99–121.

[76] Saldanha, A. Java treeview-extensible visualization. *Bioinformatics. 20*, 17 (2004), 3246–3248.

[77] Schreck, T., Schneidewind, J., and Keim, D. A. An image-based approach to visual feature space analysis. *16th Int. Conference in Central Europe on Computer Graphics, Visualization and Computer Vision.* (2008), 223–230.

[78] Seo, J., and Shneiderman., B. Interactively exploring hierarchical clustering results. *Computer. 35* (2002), 80–86.

[79] Shen, Z., and Ma, K. Path visualization for adjacency matrices. *Eurographics/IEEE-VGTC Symposium on Visualization.* (2007), 83–90.

[80] Smith, T., and Waterman, M. Identification of common molecular subsequences. *Journal of Molecular Biology. 147* (1981), 195–197.

[81] Stein, L. Creating a bioinformatics nation. *Nature. 417* (2002), 119–120.

[82] Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal 3*, 2 (1996), 186–200.

[83] Sweldens, W., and Schrder, P. Building your own wavelets at home. In *Wavelets in the Geosciences*, R. Klees and R. Haagmans, Eds., vol. 90 of *Lecture Notes in Earth Sciences*. Springer Berlin / Heidelberg, 2000, pp. 72–107.

[84] Talbot, J., Lee, B., Kapoor, A., and Tan, D. S. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. *CHI* (2009), 1283–1292.

[85] Tarassov, K., and Michnick, S. W. ivici: Interrelational visualization and correlation interface. *Genome Biology 6* (2005), 1–6.

[86] Tegge, A. N., Wang, Z., Eickholt, J., and Cheng, J. Nncon: improved protein contact map prediction using 2d-recursive neural networks. *Nucleic Acids Research* (2009), 1–4.

[87] Tokuyasu, T. A., and Albertson, D. Wavelet transforms for the analysis of microarray experiments. *Proceedings of IEEE Computer Society Bioinformatics Conference* (2003), 429–430.

[88] Varshney, S. S., Rajpal, N., and Purwar, R. Comparative study of image segmentation techniques and object matching using segmentation. *International Conference on Methods and Models in Computer Science.* (2009), 1–6.

[89] Wattenberg, J. A note on space-filling visualizations and space-filling curves. *Proc. IEEE Symposium on Information Visualization.* (2005), 181–186.

[90] Wong, P. C., and Bergeron, R. D. Multiresolution multidimensional wavelet brushing. *Proceedings of the 7th IEEE Conference on Visualization* (1996), 141–148.

[91] Yang, J., Fan, J., Hubball, D., Gao, Y., Luo, H., Ribarsky, W., and Ward, M. Semantic image browser: Bridging information visualization with automated intelligent image analysis. *IEEE Symposium on Visual Analytics Science and Technology* (2006), 191–198.

[92] Yang, J., Hubball, D., Ward, M., Rundensteiner, E., and Ribarsky, W. Value and relation display: Interactive visual exploration of large datasets with hundreds of dimensions. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 494–507.

[93] Yeung, K., and Ruzzo, W. Principal component analysis for clustering gene expression data. *Bioinformatics.* (2001), 763–774.

[94] Zhang, L., Kuljis, J., and Liu, X. Information visualization for dna microarray data analysis: A critical review. *IEEE Transactions on Systems, Man, and CyberneticsPart C: Applications and Reviews 38*, 1 (January 2008), 42–54.

[95] Ziegler, H., Nietzschmann, T., and Keim, D. A. Visual exploration and discovery of atypical behavior in financial time series data using two-dimensional colormaps. *11th International Conference Information Visualization* (2007), 308–315.

[96] Zou, H., Hastie, T., and Tibshirani., R. Sparse principal component analysis. *Journal of Computational and Graphical Statistics. 15* (2006), 265–286.