Modeling Evolutionary Constraints and Improving Multiple Sequence Alignments using Residue Couplings

K. S. M. Tozammel Hossain

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Applications

Naren Ramakrishnan, Chair
Alexey V. Onufriev
B. Aditya Prakash
Chris Bailey-Kellogg
Nathan A. Baker

Sep 13, 2016
Blacksburg, Virginia

Keywords: Residue Coupling, Mutiply Sequence Alignments, Probabilistic Graphical Models

Copyright 2016, K. S. M. Tozammel Hossain
Modeling Evolutionary Constraints and Improving Multiple Sequence Alignments using Residue Couplings

K. S. M. Tozammel Hossain

(ABSTRACT)

Residue coupling in protein families has received much attention as an important indicator toward predicting protein structures and revealing functional insight into proteins. Existing coupling methods identify largely pairwise couplings and express couplings over amino acid combinations, which do not yield a mechanistic explanation. Most of these methods primarily use a multiple protein sequence alignment—most likely a resultant alignment—which better exposes couplings and is obtained through manual tweaking of an alignment constructed by a classical alignment algorithm. Classical alignment algorithms primarily focus on capturing conservations and may not fully unveil couplings in the alignment. In this dissertation, we propose methods for capturing both pairwise and higher-order couplings in protein families. Our methods provide mechanistic explanations for couplings using physicochemical properties of amino acids and discernibility between orders. We also investigate a method for mining frequent episodes—called coupled patterns—in an alignment produced by a classical algorithm for proteins and for exploiting the coupled patterns for improving the alignment quality in terms of exposition of couplings. We demonstrate the effectiveness of our proposed methods on a large collection of sequence datasets for protein families.

This work received support from the NSF grant IIS-0905313.
Modeling Evolutionary Constraints and Improving Multiple Sequence Alignments using Residue Couplings

K. S. M. Tozammel Hossain

(GENERAL AUDIENCE ABSTRACT)

Proteins are biomolecules that comprise amino acid compounds. A chain of amino acid (a.k.a. protein sequence) forms the primary structure of a protein, and the shaping of this chain into various folds gives rise to a more complex 3D structure, a natural state of proteins. It is through structures protein performs various activities. To preserve these activities in proteins, evolution allows only those changes in protein sequences that do not disrupt the overall structures and functions of proteins. Coupling is a evolutionary phenomenon that helps proteins preserve their structures and functions. Two or more amino acid positions are coupled if changes of amino acids at a position is compensated by changes in the other position(s). In this thesis, we propose a set of probabilistic methods for modeling such couplings between two or more positions. Our methods identify the most probable couplings in a set of protein sequences and express them with probabilistic graphical models (a powerful and interpretable framework), which can be used for answering questions related to protein structures, functions, and protein synthesis. Using this notion of coupling, we also develop a method for improving the quality of multiple protein sequence alignment, a widely used tool for protein sequence analyses. We evaluate our methods with a large collection of sequence datasets for protein families, and the results substantiate the efficacy of our methods.
To my wonderful parents, sisters, and wife
Acknowledgments

My graduate experience at Virginia Tech has been wonderful due to kind support and encouragement provided towards me by many different people. It is my pleasure to thank them here.

First and foremost, I thank my advisor, Prof. Naren Ramakrishnan. Without his relentless support this dissertation would not be possible. From the very beginning of my PhD study, he set my course in the right direction. He has always given me ample freedom to pursue my research ideas. Whenever I was stuck with an obstacle, he was available to help me out. I found him very caring, generous, and patient with me—which made my graduate school experience a pleasant one. He has been a source of inspiration to me and will remain so. I will forever cherish that I had the opportunity to work with him.

I am grateful to the people who helped me conducting this research. I thank my dissertation committee members, Profs. Alexey Onufriev, Aditya Prakash, Chris Bailey-Kellogg, and Dr. Nathan Baker. Their insights and critical feedback during the proposal and the final defense helped me developing the work presented here. Dr. Chris and Dr. Nathan deserve
special mention as they have been supportive with valuable suggestions and necessary materials for this dissertation. I also thank Dr. Debprakash Patnaik, Dr. Srivatsan Laxman, and Dr. Prateek Jain, who inspired me, and helped me with early materials, to apply the concept of couplings for improving multiple sequence alignments.

Besides those who contributed to my dissertation, I acknowledge the people who influenced my life within and outside of graduate school. I am indebted to the members of Naren’s Lab: they always encouraged me to work hard, helped me with answering programming problems, and provided me new ideas for thought. Dr. Lenwood Heath, who inducted me into the PhD program at VT, deserves special mention as he has been very enthusiastic about my research. He always asked me with a big smile how I am progressing towards my degree and encouraged me to work hard. I have been fortunate to have excellent friends over the years who have inspired me for pursuing my graduate study. There are too many to list here, but few deserve special mention: Dr. M. Shahriar Hossain, Dr. Ahsanur Rahman, Naren Sundaravaradan, Dr. Abdullah Mueen, Dr. Masud Ahmed, Aminur Rahman, and Kibria Mozumder. They have been there in times of crisis. I owe them a great deal.

Finally, I thank my family for their unwavering support. My wife, Ribat, whose love, patience, and devotion have served as a crutch for me during trying times. My parents and sisters were confident in my abilities and encouraged me to pursue a PhD degree abroad. Their continuous love, support, and influence are the reasons this dissertation exists.
## Contents

1 Introduction .................................................. 1

1.1 Evolutionary Constraints on Proteins ....................... 4

1.2 Multiple Sequence Alignment ................................. 6

1.3 Probabilistic Graphical Models ............................. 7

1.4 Goals of the Dissertation ..................................... 8

1.5 Organization of the Dissertation ............................ 11

2 Couplings using Physicochemical Properties of Amino Acids 13

2.1 Introduction .................................................. 13

2.2 Literature Review ............................................. 18

2.3 Methods ..................................................... 21

2.3.1 UGMs from Inflated MSAs ............................... 21
List of Figures

1.1 An illustration of a multiple sequence alignment and couplings for a toy protein family: (a) A hypothetical protein family of 10 sequences, (b) A multiple sequence alignment of the family that is created using gaps, which represent insertion or deletion events of evolution, (c) Conservations and couplings in the family are captured in an undirected graphical model, where each node denotes a column in the alignment, an edge represents a coupling between two nodes, and each edge label shows the most enriched amino-acid combinations in the coupling.

2.1 Inferring graphical models from an MSA of a protein family: (a)-(c) illustrate input to our models and (d),(e) illustrate two different residue coupling networks.

2.2 Taylor’s classification: a Venn diagram depicting classes of amino acids based on physicochemical properties. Figure redrawn from [1].
2.3 Expansion of a multiple sequence alignment into an ‘inflated MSA’. Two classes—polarity and hydrophobicity—are used for illustration. Each column in the MSA is mapped to three columns in the expanded MSA.

2.4 Effect of alphabet length on mutual information. Here, A,P,H,S denote amino acid, polarity, hydrophobicity, and size column respectively. (a) Scatter plot of mutual information for every residue pair without normalization. (b) Scatter plot of mutual information for every residue pair with normalization. Notice the different scales of plots between (a) and (b).

2.5 Class labeling of coupled edges. The blue edges are already added to the network and dashed edges are not. The red edge is under consideration for addition in the current iteration of the algorithm. The “?” takes any of the four classes: polarity (P), hydrophobicity (H), size (S), or the default amino acid values (A).

2.6 A hypothetical residue coupling in terms of amino acid classes using a two-layered Bayesian network.

2.7 A rendering of NikR protein (PDB id 1Q5V) showing two domains: ACT domain (Nickel binding site) and RHH domain (DNA binding site). The distance between these two domains is 40Å. The molecular image is generated using VMD 1.9 [2].

2.8 A cartoon describing GPCR functionality. Figure redrawn from [3].
2.9 Histograms for class-coupling types on the NikR dataset using three methods: (a) GMRC-Inf, (b) GMRC-Inf*, and (c) HLCM. .............................................. 42

2.10 A hypothetical residue coupling in terms of amino acid classes using a factor graph model. ................................................................. 44

3.1 A toy example demonstrating a 3-order coupling. Eight subsequences for three columns $i, j,$ and $k$ in a multiple sequence alignment are shown. Residue columns $i, j,$ and $k$ exhibit a 3-order coupling with an enrichment of “AYR” and “PEN”. ................................................................. 46

3.2 A DAG representation for a graph learned by HCDG. The bottom layer represents observed variables $X$ (e.g., residues) and the upper layer denotes hidden factors $Y$. ................................................................. 53

3.3 Addition of a factor in a graph. a) While adding a 2-order factor the network score depends on the score of $X_i$ and $X_j$, which depends on neighboring nodes and residue groups containing them. b) Addition of a 3-order factor depends on its three nodes with their neighbors and residue groups they belong to. ................................................................. 56

4.1 Realignment of a hypothetical MSA using coupled pattern mining. Panel (b) is input to ARMiCoRe and (d) is the improved alignment. ............................. 66

4.2 Figure illustrating Example 2. ........................................................................................................................................................................ 70
4.3 (a) Clustering of amino acids proposed in [4]. (b) This figure describes window constraints. While looking for similar residue within a window the algorithm does not go beyond a conserved residue in a (semi)conserved column so that the (semi)conserved column is not distorted in the realignment process. . . . 75

4.4 Generating a coupled pattern set from all possible patterns. In the left figure a coupled pattern can be created from a dominant dominant pattern and three candidate non-dominant patterns that may overlap with each other. In right figure a possible construction of a coupled pattern consisting of one dominant pattern and two non-dominant pattern is shown. . . . . . . . . . . . . . . . . 79

4.5 Network $G$ used in the max-flow step. Each $\alpha_i$ is an indexed pattern and each $s_j$ is a sequence. The nodes $v_*$ and $v_\#$ denote the source and the sink respectively. Each edge from $\alpha_i$ to $s_j$ has a flow of 1 if $s_j$ contains $\alpha_i$. The minimum flow from $v_*$ to an $\alpha_i$ is $\tau$ since $\alpha_i$ has a support of at least $\tau$. . . . 81

4.6 Precision-recall plots for the dominant residue conservation threshold $\tau_d$ [0.2,0.4,0.6,0.8]. 91

4.7 Precision-recall plots for the coverage threshold $\tau$ [0.4,0.6,0.8,0.10,0.12] . . . 92

4.8 Precision-recall plots for the window size parameter $\epsilon$ [1 to 5]. . . . . . . . . 92
4.9 Pairwise sequence similarity analysis of an alignment ‘BB20006’ from RV20 dataset that contains an orphan sequence. We use SCA \[5\] for this analysis.

Fig. 4.9a has a peak for similarity score around 0.12 that indicates that the orphan is distant from the other sequences. Fig. 4.9b shows a reasonably narrow distribution without the orphan sequence with a mean pairwise similarity between sequences of about 27% and a range of 20% to 35%, which suggests that most sequences are about equally dissimilar from other.

4.10 An overview of user interaction with ARMiCoRe.

4.11 Interfaces for mining coupled patterns. (a) Loading of An input alignment. (b) Selection of coupled patterns with colored plot of corresponding residues.

4.12 Pairwise sequence similarity analysis using SCA \[5\]. Histograms for reference, ClustalW, and ARMiCoRe are drawn for the same number of bins. This figure shows that the ARMiCoRe alignment retains most of the sequence similarity structure of the reference alignment.
List of Tables

2.1 Important residues for allosteric activity in NikR collected from [6]. Residues are mapped from indices with respect to Apo Nikr (PDB id 1Q5V) to the indices of NikR MSA column. Important residues having conservation greater than 90% are not shown. ......................................................... 39

2.2 Comparisons of methods for various feature on NikR dataset. ........................................ 40

2.3 Important residues discovered by HLCM, GMRC-INF, GMRC-INF*, and GMRC in NikR. ........................................ 40

2.4 Classification of GPCR subclasses. ........................................ 41

3.1 Multiple sequence alignments used for model evaluation. ........................................ 60

3.2 Comparisons of methods for various feature on NikR dataset. ........................................ 62

3.3 Analysis of couplings with HCDG on GPCR protein family. ........................................ 63

3.4 Analysis of higher-order couplings with connected information on NikR protein family. ........................................ 64
4.1 Description of simulated datasets. Each of the dataset from A0 to F2 has 100 sequences and 100 residues.

4.2 Summary of datasets used in the study.

4.3 Comparison of ARMiCoRe with ClustalW and Cobalt on synthetic dataset.


4.5 Comparison of ARMiCoRe against ClustalW over all BaliBase datasets (using only core regions). The average scores are shown here. RV20* is curated from RV20 by removing orphan sequences.

4.6 Comparison of ARMiCoRe against ClustalW over the OXBench alignments.

4.7 Comparison of ARMiCoRe against ClustalW over the SABRE alignments.

4.8 Comparison of ARMiCoRe against ClustalW and COBALT over the CC subfamily of WW protein family.

4.9 Comparison of ARMiCoRe against ClustalW and COBALT over the PDZ family.

4.10 Comparison of ARMiCoRe against ClustalW and COBALT over the Nucleotide subfamily of GPCR protein family.
<table>
<thead>
<tr>
<th>4.11 Comparison of ARMiCoRe against ClustalW over the CC subfamily of WW</th>
</tr>
</thead>
<tbody>
<tr>
<td>family using the global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.12 Comparison of ARMiCoRe against ClustalW over the PDZ family using the</th>
</tr>
</thead>
<tbody>
<tr>
<td>global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.13 Comparison of ARMiCoRe against ClustalW over the Nucleotide subfamily</th>
</tr>
</thead>
<tbody>
<tr>
<td>of GPCR family using the global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.14 Comparison of ARMiCoRe against ClustalW over the CC subfamily of WW</th>
</tr>
</thead>
<tbody>
<tr>
<td>family using the statistical coupling analysis defined in [9]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Research in computational biology and bioinformatics has made strides over the last few decades with the advancement of sequencing techniques, which facilitate massive accumulation of biological sequences such as DNA and proteins [10, 11, 12]. Such deluge of biological sequence data unatched the door of studying various biological phenomena by sequence analysis. Now-a-days sequence analysis is conducted in many areas of computational biology and bioinformatics, including detection of homologous sequences, classification of sequences, prediction of biological functions, prediction of protein structures and fold motifs, analysis of phylogeny, analysis of evolutionary constraints, etc.

Proteins are the building blocks of a cell and play roles nearly in every cellular process. A protein is a polymeric macro-molecule that consists of amino acids, which are of twenty types. A protein sequence is a chain of amino acids linked by peptide bonds, and the chain
polypeptide) forms the primary structure of a protein [13]. For conciseness and readability each amino acid is represented with an English letter and each protein sequence is represented as a string of characters. Each position in a protein sequence is known as an amino acid residue or residue, in short. A protein sequence folds into a tertiary (three-dimensional) structure, which largely determines various characteristics and functions of proteins.

Evolution in organisms is a well-known biological phenomena. Evolutionary process is orchestrated through changes in biological sequences, the driving force behind various biological processes. As protein plays a vital role in organisms it undergoes evolutionary pressures for maintaining the proper structure and function(s). Evolutionary pressure restricts the allowable mutations to protein sequences since an abrupt mutation at a site may disrupt the functionality of a protein. These types of selective pressures or constraints play roles within a protein (intramolecular) or between proteins (intermolecular). In this study, we focus on intramolecular constraints. The most well-known evolutionary constraints are conservations and couplings (a.k.a correlated mutations, covarying residues, coevolving residues, compensatory mutation). Conservation at a residue position is considered as a lack of mutation at that position. On the other hand, couplings occurs between two or more residue positions where a compensatory mutation eventuates at a residue position with mutation(s) at the other residue positions.

Identifying evolutionary constraints can be performed through analyzing a set of evolutionarily related protein sequences since evolutionary events leave imprints on biological sequences. A set of evolutionarily related sequences usually forms a protein family: sequences within
the family typically have similarities in terms of sequence identity, structure, and functions. Sequences within a family can vary in length as the sequences undergo different level of mutations. The first step to elucidate the evolutionary constraints is to align the sequences of a family with each other so that similar residues are aligned in a column and the sequences have the exact same length. This operation is carried out by padding gaps and the resultant alignment is called a multiple sequence alignment (Fig. 1.1(b) shows a toy alignment).

Multiple sequence alignments (MSA) are used as a primary data source for discovering couplings computationally. Existing MSA algorithms focus on preserving conservations in alignments although couplings are considered as an important aspect of sequences. For this reason experts sometimes perform manual tweaking for better exposition of couplings in alignments before using the alignments in other analysis. With the increase of protein sequences the manual tweaking of alignments is becoming harder. Various computational methods have been developed for identifying couplings in proteins. Most of these methods primarily discover couplings between two residue positions and represent couplings using amino acid combinations. Such a representation of couplings lacks a mechanistic explanations since evolutionary pressure limits allowable amino acids based on the physico-chemical properties of amino acids.

This dissertation deals with identification and representation of evolutionary constraints in protein sequences. We define a novel type of coupling using physico-chemical properties of amino acids. We model both pairwise and higher-order couplings with discernibility between orders using probabilistic graphical models, a modeling tool that provides a unifying
Figure 1.1: An illustration of a multiple sequence alignment and couplings for a toy protein family: (a) A hypothetical protein family of 10 sequences, (b) A multiple sequence alignment of the family that is created using gaps, which represent insertion or deletion events of evolution, (c) Conservations and couplings in the family are captured in an undirected graphical model, where each node denotes a column in the alignment, an edge represents a coupling between two nodes, and each edge label shows the most enriched amino-acid combinations in the coupling.

framework for knowledge representation and inference in a concise and elegant way. We also propose a novel method for eliminating the manual tweaking of alignments constructed by classical algorithms for protein: we mine coupled patterns in an alignment with distorted coupled columns and use the discovered patterns for improving the quality the alignment in term of various quality scores including coupling quality score.

1.1 Evolutionary Constraints on Proteins

Evolution plays vital roles in shaping the functionality of proteins. The functionality of proteins can be explained using sequence-structure-function paradigm: a protein sequence forms a tertiary structure, which determines the functions of the protein \[14, 15\]. In other words, proteins with different sequences but similar structure are likely to perform similar function(s). To maintain the functionality of proteins evolution rate in protein structures
is much slower compare to evolution rate in sequences: evolutionary constraints limit the allowable mutation at a particular site so that the stability and functions of the protein do not change substantially. Evolutionary constraints play roles within a protein (intra-molecular) or between proteins (inter-molecular). There are two types of constraints that are manifested in sequence records of a family: conservation and coupling. Conservation of a residue position can be seen as a lack of mutation at the residue position. Within a protein family, a particular residue position is *conserved* if a particular amino acid occurs at that residue position for most of the members in the family \[16\]. For example, position 2 of Fig 1.1(b) is conserved because position 2 at every sequence contains residue ‘W’. Coupling occurs between two or more residue positions, which may be far apart in the sequence but close in 3-D structure. Couplings are also known as correlated mutations, compensatory mutations, covarying residues, or coevolving residues. Two residues are *coupled* if certain amino acid combinations occur at these positions in the MSA more frequently than others \[9, 17\]. Fig. 1.1(b) illustrates an example of a residue coupling between position 3 and position 8. In Fig. 1.1(b), whenever there is a ‘K’ at position 3, then there is a ‘T’ at position 8 and whenever there is an ‘M’ at position 3, then there is a ‘V’ at position 8. The choice of a particular amino acid as a substitute for another amino acid within a coupling depends on the physicochemical properties of amino acids concerning that coupling. For example, if a residue is mutated with a larger size amino acid it may require compensating mutation(s) at other location(s) to maintain the protein’s proper structure and function(s). Based on number of participating residues, couplings can be further divided into two groups: pairwise
coupling and higher-order coupling. In a pairwise coupling two residues participate, whereas three or more residues constitute a higher-order coupling. From the perspective of distance between participating residues in couplings, we can divide coupling into two groups: direct coupling and indirect coupling. When couplings between residues imply physical contact between them, then this type of coupling is a direct coupling. On the other hand, a coupling between distant residues is known as an indirect coupling.

1.2 Multiple Sequence Alignment

Multiple sequence alignment is a fundamental step in biological sequence analysis as it elucidates the embedded evolutionary events that transpired at sequences over the time. MSA is a necessary step that allows researchers to answer deeper questions like identifying conserved regions, detecting functional motifs, profiling of genetic diseases, analysing phylogeny, profiling and prediction of ancestral sequences [18]. The first computational method for sequence alignment was developed in 1960s [18, 19]. Since then, a great number of algorithms have been proposed for solving the problem. Classical multiple sequence alignment algorithms aim to maximize conservations as much as possible in an alignment. These algorithms insert gaps into the sequences, if necessary, to match the length of each sequences. A gap in a sequence represents either an insertion or deletion of an amino acid. For example, Fig. 1.1(a) represents a toy protein family of 10 sequences with various lengths and Fig. 1.1(b) illustrates an alignment of the family. Recent studies have established that coupling is an important
evolutionary constraint that plays roles on sequences. Although couplings are seen as an important aspect of sequence evolution, classical MSA algorithms ignore couplings while constructing alignments. It is interesting to investigate whether the concept of coupling can be incorporated in MSA algorithms so that the quality of alignments become richer.

1.3 Probabilistic Graphical Models

Probabilistic graphical model (PGM) combines concepts from probability theory and graph theory. A probabilistic graphical model encodes conditional independences between random variables in a system using a graph, where nodes represent random variables and edges represent dependency (or independency) [20][21]. It is a powerful model for compactly representing joint probability distribution of a set of random variables in a complex system. PGMs are widely used in bioinformatics, neuroscience, natural language processing, and image processing [20]. There are essentially three types of probabilistic graphical models: directed graphical model, undirected graphical model, and hybrid graphical model. There is a class of graphical model within an undirected graphical model which is worth mentioning: factor graph model. Three types of problems are mainly associated with modeling interactions using PGMs: (a) represent the model, (b) perform inference using the model, and (c) learn the structure of the model.

We use directed, undirected, and factor graph models for representing couplings in protein sequences. These types of models are suitable for our problems as we aim to model in-
interactions between residues with and without hidden factors, distinguish various orders of couplings, and perform inferences and predictions with the models.

1.4 Goals of the Dissertation

There are two distinct goals of this dissertation: (a) modeling couplings using a novel representation and with higher-order and (b) improving the quality of multiple sequence alignment using coupled patterns. We propose three broad problems in these two spaces that we seek to explore.

Topic 1: Couplings using physicochemical properties of amino acids

Many algorithmic techniques have been proposed to discover couplings in protein families. These approaches discover couplings over amino acid combinations but do not yield mechanistic or other explanations for such couplings. We propose to study couplings in terms of amino acid classes such as polarity, hydrophobicity, and size, and present two algorithms for learning probabilistic graphical models of amino acid class-based residue couplings. Our probabilistic graphical models provide a sound basis for predictive, diagnostic, and abductive reasoning. Further, our methods can take optional structural priors into account for building graphical models. The resulting models are useful in assessing the likelihood of a new protein to be a member of a family and for designing new protein sequences by sampling from the graphical model. We apply our approaches to understanding couplings in two pro-
tein families: Nickel-responsive transcription factors (NikR) and G-protein coupled receptors (GPCRs). The results demonstrate that our graphical models based on sequences, physico-chemical properties, and protein structure are capable of detecting amino acid class-based couplings between important residues that play roles in activities of these two families.

**Topic 2: Representation and identification of higher-order couplings**

Current research in modeling evolutionary constraints predominantly focuses on discovering pairwise couplings between residues. Research suggests that only pairwise couplings may not be sufficient for better modeling of evolutionary relationships, rather additional higher-order interactions may help better learning of a protein’s structure and functions. Although recent endeavors show some success in modeling higher-order couplings in proteins, these studies focus on identifying group of coupled residues, but do not differentiate the contributions of couplings from each order to the total couplings in the group. There is a pressing need for modeling higher-order couplings between residues where couplings of different orders within a set will be distinguished.

We propose to study higher-order couplings in proteins: couplings of various orders within a set of coupled residues will be distinguished and their contributions to the total couplings of the set will be estimated. We represent and infer such couplings using hidden factors and express such factors with directed and factor graph models.
**Topic 3: Use of coupled patterns for improving multiple sequence alignments for proteins**

Aligning multiple biological sequences is a key step in elucidating evolutionary relationships, annotating newly sequenced segments, and understanding the relationship between biological sequences and functions. Classical MSA algorithms are designed to primarily capture conservations in sequences whereas couplings, or correlated mutations, are well known as an additional important aspect of sequence evolution. As a result, better exposition of couplings is sometimes one of the reasons for hand-tweaking of MSAs by practitioners.

We present a novel approach to a classical bioinformatics problem, viz. multiple sequence alignment (MSA) of gene and protein sequences. Our method introduces a distinctly pattern mining approach to improving MSAs: using frequent episode mining as a foundational basis, we define the notion of a coupled pattern and demonstrate how the discovery and tiling of coupled patterns using a max-flow approach can yield MSAs that are better than conservation-based alignments. Although we were motivated to improve MSAs for the sake of better exposing couplings, we demonstrate that our MSAs are also improvements in terms of traditional metrics of assessment. We demonstrate the effectiveness of our method on a large collection of datasets.

We are motivated to study these problems in order to advance the research of residue couplings, which can help pursue relevant scientific questions. Recent studies have successfully applied residue couplings to contact predictions in proteins [22][23][24], discovering pathways...
of residue interaction or allosteric communication [9], identifying protein-protein interaction sites and predict contacts across interfaces [25, 26], and designing synthetic proteins [27]. We have developed tools for inferring couplings in proteins and restoring couplings in traditional MSAs with an aim to help analysts perform some of these tasks.

1.5 Organization of the Dissertation

The remainder of the dissertation proposal is organized as follows. In Chapter 2 we address the problem of modeling couplings using physico-chemical properties of amino acids. Here we present how to define couplings in terms of physico-chemical properties of amino acids and propose two probabilistic graphical models—directed and undirected—for encoding the couplings. We use real-world data for learning and evaluating our model.

In Chapter 3 we define higher-order couplings in proteins and propose two models for learning such couplings. Our approaches are built on the notion of hidden factors and express higher-order couplings with directed graphical model and factor graph model.

In Chapter 4 we investigate the problem of improving multiple sequence alignment using coupled patterns. Given an alignment generated using a classical MSA algorithm we identify coupled patterns using a level-wise pattern finding algorithm. Our algorithm then uses the significant coupled patterns for generating a set of constraints, which are employed to realign the alignment for improvement.

Chapter 5 summarizes our experience with couplings and learning probabilistic graphical
models. We discuss the unique aspects involved in each of the problem presented in this dissertation. We also present some of the future directions that stem from this dissertation.
Chapter 2

Couplings using Physicochemical Properties of Amino Acids

2.1 Introduction

Proteins are grouped into families based on similarity of function and structure. It is generally assumed that evolutionary pressures in protein families to maintain structure and function manifest in the underlying sequences. Two well-known types of constraints are conservation and coupling, which are defined in Sec. 1.1. The most widely studied constraint is conservation of individual residues. Conservation of residues usually occurs at functionally and/or structurally important sites within a protein fold (shared by the protein family). For example in Figure 2.1(a), a multiple sequence alignment (MSA) of 10 sequences, the second
residue is 100% conserved with occurrence of amino acid "W".

A variety of recent studies have used MSAs to calculate correlations in mutations at several positions within an alignment and between alignments [9, 28, 29, 30]. These correlations have been hypothesized to result from structural/functional coupling between these positions within the protein [31]. For example, residues 3 and 8 are coupled in Fig. 2.1(d) because the presence of "K" (or "M") at the third residue co-occurs with "T" (or "V") at the eighth residue position. Going beyond sequence conservation, couplings provide additional information about potentially important structural/functional connections between residues within a protein family. Previous studies [9, 31, 28] show that residue couplings play key roles in transducing signals in cellular systems.

In this chapter, we study residue couplings that manifest at the level of amino acid classes
Figure 2.2: Taylor’s classification: a Venn diagram depicting classes of amino acids based on physicochemical properties. Figure redrawn from [1].

rather than just the occurrence of particular letters within an MSA. Our underlying hypothesis is that if structural and functional behaviors are the underlying cause of residue couplings within MSAs, then couplings are more naturally studied at the level of amino acid properties. We are motivated by the prior work of Thomas et al. [32, 28] which proposes probabilistic graphical models for capturing couplings in a protein family in terms of amino acids. Graphical models are useful for supporting better investigation, characterization, and design of proteins. The above works infer an undirected graphical model for couplings given an MSA where each node (variable) in the graph corresponds to a residue (column) in the MSA and an edge between two residues represents significant correlation between them. Figure 2.1(a),(b) illustrates the typical input (an MSA and a structural prior) and Figure 2.1(d) is an output (undirected graphical model) of the procedure of Thomas et al. In the output model (see Fig. 2.1(d)), three residue pairs—(3,8), (6,7), and (9,10)—are coupled.

Evolution is the key factor determining the functions and structures of proteins. It is assumed
that the type of amino acid at each residue position within a protein structure is (at least somewhat) constrained by its surrounding residues. Therefore, explaining the couplings in terms of amino acid classes is desirable. To achieve this, we consider amino acid classes based on physicochemical properties (see Fig. 2.2).

Graphical models can be made more expressive if we represent the couplings (edges in the graphs) in terms of underlying physicochemical properties. Figure 2.1(c) is a Venn diagram of three amino acid classes—polarity, hydrophobicity, and size. Figure 2.1(e) illustrates three couplings in terms of amino acid classes. For example, residue 3 and residue 8 are coupled in term of “polarity-polarity”, which means correlated changes of polarities occur at these two positions—a change from polar to nonpolar amino acids at residue 3, for instance, induces concomitant change from polar to nonpolar amino acid at residue 8. Similarly, residue 6 and residue 7 are also correlated since a change from hydrophobic to hydrophilic amino acids at residue 6 induces a change from big to small amino acids at residue 7. There is no edge between residue 5 and residue 7, however, because they are independent given residue 6. Hence, the coupling between residue 5 and residue 7 is explained via couplings (5,6) and (6,7). This is one of the key features of undirected graphical models as they help distinguish direct couplings from indirect couplings. Note that the coupling between residue 9 and residue 10 (originally present in Fig. 2.1(d)) does not occur in Figure 2.1(e) due to class conservation in residues 9 and 10. Also note that the coupling between residue 5 and residue 6 in Figure 2.1(e) is not apparent in Figure 2.1(d). Class-based representations of couplings hence recognize a different set of relationships than amino acid value-based
couplings. We show how the class-based representation leads to more explainable models and suggest alternative criteria for protein design.

The key contributions of this study are as follows:

1. We investigate whether residue couplings manifest at the level of amino acid classes and answer this question in the affirmative for the two protein families studied here.

2. We design new probabilistic graphical models for capturing residue coupling in terms of amino acid classes. Like the work of Thomas et al. [28] our models are precise and give explainable representations of couplings in a protein family. They can be used to assess the likelihood of a protein to be in a family and thus constitute the driver for protein design.

3. We demonstrate successful applications to the NikR and GPCR protein families, two key demonstrators for protein constraint modeling.

The rest of the chapter is organized as follows. We review related literature in Section 2.2. Methodologies for inferring graphical models are described in Section 2.3. Experimental results are provided in Section 2.4 followed by a discussion in Section 2.5. A version of this chapter is available in the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD) [33].
2.2 Literature Review

Early research on correlated amino acids was conducted by Lockless and Ranganathan \[9\]. Through statistical analysis they quantified correlated amino acid positions in a protein family from its MSA. Their work is based on two hypotheses, which are derived from empirical observation of sequence evolution. First, the distribution of amino acids at a position should approach their mean abundance in all proteins if there is a lack of evolutionary constraint at that position; deviance from mean values would, therefore, indicate evolutionary pressure to prefer particular amino acid(s). Second, if two positions are functionally coupled, then there should be mutually constrained evolution at the two positions even if they are distantly positioned in the protein structure. The authors developed two statistical parameters for conservation and coupling based on the above hypothesis, and use these parameters to discover conserved and correlated amino acid positions. In their SCA method, a residue position in an MSA of the family is set to its most frequent amino acid, and the distribution of amino acids at another position (with deviant sequence at the first position removed) is observed. If the observed distribution of amino acids at the other position is significantly different from the distribution in the original MSA, then these two positions are considered to be coupled. Application of their method on the PDZ protein family successfully determined correlated amino acids that form a protein-protein binding site.

Valdar surveyed different methods for scoring residue conservation \[1\]. Quantitative assessment of conservation is important because it sets a baseline for determining coupling. In
particular, many algorithms for detecting correlated residues run into trouble when there is an ‘in between’ level of conservation at a residue position. In this survey, the author investigates about 20 conservation measures and evaluates their strengths and weaknesses.

Fodor and Aldrich reviewed four broad categories of measures for detecting correlation in amino acids [34]. These categories are: 1) Observed Minus Expected Squared Covariance Algorithm (OMES), 2) Mutual Information Covariance Algorithm (MI), 3) Statistical Coupling Analysis Covariance Algorithm (SCA; mentioned above), and 4) McLachlan Based Substitution Correlation (McBASC). They applied these four measures on synthetic as well as real datasets and reported a general lack of agreement among the measures. One of the reasons for the discrepancy is sensitivity to conservation among the methods, in particular, when they try to correlate residues of intermediate-level conservation. The sensitivity to conservation shows a clear trend with algorithms favoring the order McBASC > OMES > SCA > MI.

Although current research is successful in discovering conserved and correlated amino acids, they fail to give a formal probabilistic model. Thomas et al. [28] is a notable exception. This paper differentiates between direct and indirect correlations which previous methods did not. Moreover, the models discovered by this work can be extended into differential graphical models which can be applied to protein families with different functional classes and can be used to discover subfamily-specific constraints (conservation and coupling) as opposed to family-wide constraints.

The above research on coupling and conservation do not aim to model evolutionary processes
**Figure 2.3:** Expansion of a multiple sequence alignment into an ‘inflated MSA’. Two classes—polarity and hydrophobicity—are used for illustration. Each column in the MSA is mapped to three columns in the expanded MSA.

directly. Yeang and Haussler, in contrast, suggest a new model of correlation in and across protein families employing evolution [29]. They refer to their model as a *coevolutionary model* and their key claims are: coevolving protein domains are functionally coupled, coevolving positions are spatially coupled, and coevolving positions are at functionally important sites. The authors give a probabilistic formulation for the model employing a phylogenetic tree for detecting correlated residues.

A more recent work, by Little and Chen [30], studies correlated residues using mutual information to uncover evolutionary constraints. The authors show that mutual information not only captures coevolutionary information but also non-coevolutionary information such as conservation. One of the strong non-coevolutionary biases is stochastic bias. By first calculating mutual information between two residues which have evolved randomly (referred to as random mutual information), the authors then study relationships with other mutual information quantities to detect the presence of non-coevolutionary biases.
2.3 Methods

A multiple sequence alignment $S$ allows us to summarize each residue position in terms of the probabilities of encountering each of the 20 amino acids (or a gap) in that position. Let $V = \{v_1, \ldots, v_n\}$ be a set of random variables, one for each residue position. The MSA then gives a distribution of amino acids for each random variable. We present two different classes of probabilistic graphical models to detect couplings. These inferred graphical models capture conditional dependence and independence among residues, as revealed by the MSA. The first approach uses an undirected graphical model (UGM), also known as a Markov random field. The second method employs a specific hierarchical latent class model (HLCM) which is a two-layered Bayesian network.

2.3.1 UGMs from Inflated MSAs

This approach can be viewed as an extension of the work of Thomas et al. [28]. It induces an undirected graphical model, $G = (V, E)$, where each node, $v \in V$, corresponds to a random variable and each edge, $(u, v) \in E$, represents a direct relationship between random variables $u$ and $v$. In our problem setting, a node of $G$ corresponds to a residue position (a column of the given MSA) and each edge represents a coupling between two residues. In this method, we redefine the approach of Thomas et al. [28] to discover MSA residue position couplings in terms of amino acid classes rather than residue values.
Inflated MSA

We augment the MSA $S$ of a protein family by introducing extra ‘columns’ for each residue. Let $l$ be the number of amino acid classes and $A_i$ be the alphabet for the $i$th class where $1 \leq i \leq l$. Legal vocabularies for the classes can be constructed with the help of Taylor’s diagram (see Fig. 2.2). For example, possible classes are polarity, hydrophobicity, size, charge, and aromaticity. Moreover, we may consider the amino acid sequence of a column as a “amino acid name” class. These classes take different values; e.g., the polarity class takes two values: polar and non-polar. Each column of $S$ is mapped to $l$ subcolumns to obtain an inflated MSA $S_e$ where the extra columns (referred to as subcolumns) encode the corresponding class values. We use $v_{ik}$ to denote the $k$th subcolumn of residue $v_i$. Figure 2.3 illustrates the above procedure for obtaining an inflated alignment $S_e$. (A gap character in $S$ is mapped to a gap character in $S_e$.)

Detecting Coupled Residues

Couplings between residues can be quantified by many statistical and information-theoretic metrics [34]. In our model, we use conditional mutual information because it allows us to separate direct from indirect correlations. Recall that the mutual information (MI), $I(v_i, v_j)$, between residues $v_i$ and $v_j$ is given by:
\[
I(v_i, v_j) = \sum_{a \in A} \sum_{b \in A} P(v_i = a, v_j = b) \cdot \log \frac{P(v_i = a, v_j = b)}{P(v_i = a)P(v_j = b)}
\] (2.1)

where the probabilities are all assessed from \( S \). If \( I(v_i, v_j) \) is non-zero, then they are dependent, and each residue position \((v_i \text{ or } v_j)\) encodes information that can be used to predict the other. In the original graphical models of residue coupling (GMRC) model \[28\], Thomas et al. use conditional mutual information:

\[
I(v_i, v_j | v_k) = \sum_{c \in A^*} \sum_{a \in A} \sum_{b \in A} P(v_i = a, v_j = b | v_k = c) 
\cdot \log \frac{P(v_i = a, v_j = b | v_k = c)}{P(v_i = a | v_k = c)P(v_j = b | v_k = c)}
\] (2.2)

to construct edges, where the conditionals are estimated by subsetting residue \( k \) to its most frequently occurring amino acid types \((A^* \subset A)\). The most frequently occurring amino acid types are those that appear in at least 15% of the original sequences in the subset. As discussed \[9\], such a bound is required in order to ensure sufficient fidelity to the original MSA and allow for evolutionary exploration.

For modeling residue position couplings in terms of amino acid classes, we use Eq. 2.2. As each residue in \( S_e \) has \( l \) columns, we consider all \( O(l^2) \) pairs of columns for estimating mutual information between two residues. For calculating conditional mutual information in
an inflated MSA, we condition a residue to its most appropriate class. The most appropriate
class is the one that reduces the overall network score the most. The modified equation for
conditional mutual information is as follows:

\[ I_e(v_i, v_j|v_{kr}) = \sum_{p=1}^{l} \sum_{q=1}^{l} I_e(v_{ip}, v_{jq}|v_{kr}) \]  

(2.3)

where

\[ I_e(v_{ip}, v_{jq}|v_{kr}) = \sum_{c \in A^*} \sum_{a \in A_p} \sum_{b \in A_q} P(v_{ip} = a, v_{jq} = b|v_{kr} = c) \]

\[ \cdot \log \frac{P(v_{ip} = a, v_{jq} = b|v_{kr} = c)}{P(v_{ip} = a|v_{kr} = c)P(v_{jq} = b|v_{kr} = c)} \]  

(2.4)

Here \( A_i \) denote the alphabet of the \( i \)th amino acid class where \( 1 \leq i \leq l \). The conditional
variable \( v_k \) is set to the \( r \)th class. If \( I_e(v_i, v_j|v_{kr}) = 0 \), then it implies that residue \( v_i \) and \( v_j \)
are independent conditioned on the \( r \)th class of \( v_k \). Observe that we can subset the residue
\( v_k \) to any class out of \( l \) classes. We take the minimum of \( I_e(v_i, v_j|v_{kr}) \) for \( 1 \leq r \leq l \) to obtain
the final mutual information between \( v_i \) and \( v_j \).

**Normalized Mutual Information**

In an inflated MSA, the subcolumns corresponding to a residue take values from different
alphabets of different sizes. Let \( v_{ip} \) and \( v_{jq} \) be two subcolumns that take values from alpha-
bets $A_p$ and $A_q$ respectively. To understand the effect of the sizes of alphabets in mutual information score, we calculate pairwise mutual information of subcolumns for every residue pair and produce a scatter plot (see Fig. 2.4(a)).

In Fig. 2.4(a), we see that $MI(A, A)$ is dominating over $MI(P, P)$, $MI(H, H)$, and $MI(S, S)$. This is expected, because amino acids are of 21 types whereas polarity, hydrophobicity, and size have 3 types. We adopt the following equation to normalize mutual information scores proposed by Yao [35]:

$$I_{\text{norm}}(v_{ip}, v_{jq} | v_{kr}) = \frac{I(v_{ip}, v_{jq} | v_{kr})}{\min(H(v_{ip} | v_{kr}), H(v_{jq} | v_{kr})}$$

where $H(v_{ip} | v_{kr})$ and $H(v_{jq} | v_{kr})$ denote the conditional entropy.
Algorithm 1 GMRC-INF($\mathcal{S}, P$)

**Input:** $\mathcal{S}$ (multiple sequence alignment), $P$ (possible edges)

**Output:** $G$ (a graph that captures couplings in $\mathcal{S}$)

1. $V = \{v_1, v_2, \ldots, v_n\}$
2. $E \leftarrow \emptyset$
3. $s \leftarrow S_{\text{UGM}}(G = (V, E))$
4. for all $e = (v_i, v_j) \in P$ do
5.   $C_e \leftarrow s - S_{\text{UGM}}(G = (V, \{e\}))$
6. while stopping criterion is not satisfied do
7.   $e \leftarrow \text{arg max}_{e \in P - E} C_e$
8. if $e$ is significant then
9.   $E \leftarrow E \cup \{e\}$
10. label $e$ based on the score
11. $s \leftarrow s - C_e$
12. for all $e' \in P - E$ s.t $e$ and $e'$ share a vertex do
13.   $C_{e'} \leftarrow s - S_{\text{UGM}}(G = (V, E \cup \{e'\}))$
14. return $G = (V, E)$

**Learning UGMs**

Given an expanded MSA $\mathcal{S}_e$, we infer a graphical model by finding *decouplers* which are sets of variables that makes other variables independent. If two residues $v_i$ and $v_j$ are independent given $v_k$, then $v_k$ is a decoupler for $v_i$ and $v_j$. In this case, we add edges $(v_i, v_k)$ and $(v_j, v_k)$ to the graph. Thus the relationship between $v_i$ and $v_j$ is explained transitively by edges $(v_i, v_k)$ and $(v_j, v_k)$. Moreover, we can consider a prior that can be calculated from a contact graph of a representative member of the family. A prior gives a set of edges between residues which are close in three-dimensional structure. When a residue contact network is given as a prior, we consider each edge of the residue contact network as a potential candidate for couplings. Without a prior, we consider all pairwise residues for coupling. Algorithm 1 gives the formal details for inferring a graphical model.
Our algorithm builds the graph in a greedy manner. At each step, the algorithm chooses the edge from a set of possible couplings which scores best with respect to the current graph. The score of the graph is given by:

$$S_{UGM}(G = (V, E)) = \sum_{v_i \in V} \sum_{v_j \notin N(v_i)} I_e(v_i, v_j | N(v_i))$$  \hspace{1cm} (2.6)$$

where $N(v_i)$ is the set neighbors of $v_i$.

The calculation of conditional mutual information and labeling of edges with different properties is illustrated in Fig. 2.5. In Fig. 2.5 we consider edge $(v_i, v_k)$ for addition to the graph where $v_i$ already has two neighbors $v_l$ and $v_m$. The edge $(v_i, v_l)$ has the label S-H which means the coupling models $v_i$ with respect to size and $v_l$ with respect to hydrophobicity. Similarly, the edge $(v_i, v_m)$ has the label P-P which means the coupling between $v_i$ and $v_m$ can be described with respect to their polarities. To evaluate the edge $(v_i, v_k)$, we condition on $v_m$ and $v_l$ first and then condition $v_k$ on any of the properties. We then sum up all $I_e(v_i, v_j)$, where $v_j \notin \{v_l, v_m, v_k\}$. The subsetting class of $v_k$ for which we obtain a maximum for $\sum I_e(v_i, v_j)$ is the label that we finally assign to $v_k$ (the question mark in Fig. 2.5) if the edge $(v_i, v_k)$ is added. Similarly, we do the same calculation for $v_k$ while subsetting only $v_i$, as the residue $v_k$ does not have any neighbors in the current network.

Algorithm 1 can incorporate various stopping criteria: 1) stop when a newly added edge does not contribute much to the score reduction of the graph, 2) stop when a designated number of edges have been added, and 3) stop when the likelihood of the model is within acceptable
Figure 2.5: Class labeling of coupled edges. The blue edges are already added to the network and dashed edges are not. The red edge is under consideration for addition in the current iteration of the algorithm. The “?” takes any of the four classes: polarity (P), hydrophobicity (H), size (S), or the default amino acid values (A).

bounds. We use the first criterion in our model. Algorithm I is a heuristic approach. With naive implementation of this algorithm the running time per iteration is $O(dn^2)$ where $n$ is the number of residues in a family and $d$ is the maximum degree of nodes in the prior. With an uninformative prior, $d$ is $O(n)$; thus the running time per iteration is $O(n^3)$. By caching and preprocessing conditional mutual information, the running time per iteration can be reduced to $O(dn)$ and $O(n^2)$ with and without prior, respectively.

2.3.2 Hierarchical Latent Class Models

A latent class model (LCM) is a hidden-variable model which consists of a hidden (class) variable and a set of observed variables [36]. The semantics of an LCM are that the observed variables are independent given a value of the class variable. Let $u$ and $v$ be two observed
Figure 2.6: A hypothetical residue coupling in terms of amino acid classes using a two-layered Bayesian network.

Variables. The latent class model of $u$ and $v$ introduces a latent variable $z$, so that

$$P(u, v) = \sum_k P(z = k)P(u|z = k)P(v|z = k)$$

(2.7)

When the number of observed variables increases, the LCM model performs poorly due to the strong assumption of local independence. To improve the model, Zhang et al. proposed a richer, tree-structured, latent variable model [37]. Our hierarchical model is a restricted case of the model proposed by Zhang et al. We propose a two-layered binary hierarchical latent class model where the lower layer consists all the observed variables and the upper layer consists of hidden class variables. In our problem setting, observed variables correspond to residues and the hidden class variables take values from all possible permutations of pairwise amino acid classes. Figure 2.6 illustrates a hypothetical hierarchical latent class model.

Let $Z$ be the set of all hidden variables and $V$ be the set of observed variables. The joint
probability distribution of the model is as follows:

$$P(Z) \prod_{i=1}^{n} P(v_i | \text{Pa}(v_i))$$  \hspace{1cm} (2.8)

where \(\text{Pa}(v_i)\) denotes the set of parents of \(v_i\).

**Learning a HLCM**

We learn this model in a greedy fashion as before. We define the following scoring function:

$$S_{\text{HLCM}}(G = (\{V, Z\}, E)) = \sum_{v_i \in V} \sum_{v_j \not\in \text{Pa}(v_i)} I_e(v_i, v_j | \text{Pa}(v_i))$$ \hspace{1cm} (2.9)

where \(\text{Pa}(v_i)\) is the set neighbors of \(v_i\). When we condition on the parent nodes, we use a 35% support threshold for the sequences. This support threshold is required in order to ensure sufficient fidelity to the original MSA and allow for evolutionary exploration. From extensive experiments with this parameter (data not shown), we found that while there is some variation in the edges with changes of this parameter from 15% to 60%, many of the best edges are retained when support threshold is 35%. Moreover, the model has less number of couplings when support threshold is 35% which is an indication in the reduction of the overfitting effect. Besides, we use a parameter \textit{minsupport} which is set to 2; \textit{minsupport} is used to avoid class conservation between sequences. The value of \textit{minsupport} for two residue positions is the number of class-values combinations for which the number of sequences in
each subset is greater than the support threshold. When minsupport is 1 for two residue positions, we consider that a class conservation has occurred in these residue positions. The algorithm chooses a pair of residues for which introducing a hidden variable reduces the current network score the most. We then add the hidden variable if it is statistically significant. Algorithm 2 gives the formal details for learning HLCMs. We can employ various stopping criteria: 1) stop when a newly added hidden node does not contribute much to the score reduction of the graph, 2) stop when a designated number of hidden nodes have been added, and 3) stop when the likelihood of the model is within acceptable bounds. Similar to Algorithm 1, Algorithm 2 is a heuristic approach. We use the first criterion in our model. With prior the running time per iteration is $O(dn)$, where $n$ is the number of residues in a family and $d$ is the maximum degree of nodes in the prior. With an uninformative prior, $d$ is $O(n)$; thus the running time per iteration is $O(n^2)$.

2.3.3 Statistical Significance

While learning the edges, hidden nodes or factors of the above graphical models, we assess the significance of each coupling imputed. In both algorithms, we perform a statistical significance test on potential pairs of residues before adding an edge or hidden variable to the graph. To compute the significance of the edge, we use $p$-values to assess the probability that the null hypothesis is true. In this case, the null hypothesis is that two residues are truly independent rather than coupled. We use the $\chi^2$-squared test on potential edges. If $p$-value is less than a certain threshold $p_\theta$, we add the edge to the graph. In our experiment,
Algorithm 2 HLCM($\mathcal{S}, P$)

**Input:** $\mathcal{S}$ (multiple sequence alignment), $P$ (possible pairs of residues)

**Output:** $G$ (a graph that captures couplings in $\mathcal{S}$)

1. $V = \{v_1, v_2, \ldots, v_n\}$
2. $Z \leftarrow \emptyset$ \Comment*[r]{set of hidden nodes}
3. $E \leftarrow \emptyset$
4. $T \leftarrow \emptyset$ \Comment*[r]{tabu list of residue pairs}
5. $s \leftarrow \text{S}_{\text{HLCM}}(G = (V, E))$
6. for all $e = (v_i, v_j) \in P$ do
7. \hspace{1em} $E' \leftarrow \{(h_e, v_i), (h_e, v_j)\}$ \Comment*[r]{$h_e$ is a hidden class between $v_i$ and $v_j$}
8. \hspace{1em} $C_e \leftarrow s - \text{S}_{\text{HLCM}}(G = (\{V, \{h_{ij}\}\}, E'))$
9. while stopping criterion is not satisfied do
10. \hspace{2em} $e \leftarrow \text{arg max}_{e \in P - T} C_e$
11. \hspace{2em} if $e$ is significant for coupling then
12. \hspace{3em} $E \leftarrow E \cup \{(h_e, v_i), (h_e, v_j)\}$
13. \hspace{3em} $Z \leftarrow Z \cup \{h_e\}$
14. \hspace{3em} $T \leftarrow T \cup \{e\}$
15. \hspace{2em} label two edges of $h_e$ based on the score
16. \hspace{2em} $s \leftarrow s - C_e$
17. \hspace{2em} for all $e' = (v_k, v_l) \in P - T$ s.t $e$ and $e'$ share a vertex do
18. \hspace{3em} $E'' \leftarrow \{(h_{e'}, v_k), (h_{e'}, v_l)\}$
19. \hspace{3em} $C_{e'} \leftarrow s - \text{S}_{\text{HLCM}}(G = (\{V, Z\}, E \cup E''))$
20. return $G = (V, E)$
we use $p_\theta = 0.005$.

2.3.4 Classification

The graphical models learned by algorithm are useful for annotating protein sequences of unknown class membership with functional classes. To demonstrate the classification methodology, we consider HLCM as an example. We adopt Eq. 2.10 to estimate the parameters of a residue in the HLCM model. The reason for using this estimator is that the MSA may not sufficiently represent every possible amino acid value for each residue position. Therefore, we must consider the possibility that an amino acid value may not occur in the MSA but still be a member of the family. In Eq. 2.10, $|S|$ is number of sequences in the MSA and $\alpha$ is a parameter that weights the importance of missing data. We employ a value of .1 for $\alpha$ but tests (data not shown) indicate that results are similar for values in [0.1, 0.3].

\[
P(v = a) = \frac{freq(v = a) + \alpha |S|}{|S|(1 + \alpha)}
\]  

(2.10)

Given two different graphical models, $G_{C_1}$ and $G_{C_2}$, say for two different classes, we can classify a new sequence $s$ into either functional class $C_1$ or $C_2$ by computing the log likelihood ratio $LLR$: 
If $LLR$ is greater than 0 then, then we classify $s$ to the class $C_1$; otherwise, we classify it to the class $C_2$. 

2.4 Experiments

In this section, we describe the datasets that we use to evaluate our model and show results that reflect the capabilities of our models. We seek to answer the following questions using our evaluation:

1. How do our graphical models fare compared to other methods? Do our learned models capture important covariation in the protein family? (Section 2.4.2)

2. Do the learned graphical models have discriminatory power to classify new protein sequences? (Section 2.4.3)

3. What forms of amino acid class combinations are prevalent in the couplings underlying a family? (Section 2.4.4)
2.4.1 Datasets

Nickel receptor protein family

The Nickel receptor protein family (NikR) consists of repressor proteins that bind nickel and recognize a specific DNA sequence when nickel is present, thereby repressing gene transcription. In the *E. coli* bacterium, nickel ions are necessary for the catalytic activity of metalloprotein enzymes under anaerobic conditions; NikABCDE permease acquires Ni\(^{2+}\) ions for the bacterium [6]. NikR is one of the two nickel-responsive repressors which control the excessive accumulation of Ni\(^{2+}\) ions by repressing the expression of NikABCDE. When Ni\(^{2+}\) binds to NikR, it undergoes conformational changes for binding to DNA at the NikABCDE operator region and represses NikABCDE [6].

NikR is a homotetramer consisting of two distinct domains [38]. The N-terminal domain of each chain has 50 amino acids and constitutes a ribbon-helix-helix (RHH) domains that contact the DNA. The C-terminal of each chain consisting of 83 amino acids form a tetramer composed of four ACT domains that together contain the high-affinity Ni\(^{2+}\) binding sites [6]. Figure 2.7 shows a representative NikR structure determined by X-ray crystallography [6].

We organized an MSA of the NikR family that has 82 sequences which are used to study allostERIC communication in NikR [6]. Each sequence has 204 residues. For a structural prior, we use Apo-NikR (pdb id 1Q5V) as a representative member of the NikR family and calculate prior edges from its contact map. Residue pairs within 7Å of each other are considered to be in contact which gives us 734 edges as a prior. We use this prior for the
analysis to ensure that all identified relationships have direct mechanistic explanations.

**G-protein coupled receptors**

G-protein coupled receptors (GPCRs; see Fig. 2.8) represent a class of large and diverse protein family and provide an explicit demonstration of allosteric communication. The primary function of this proteins is to transduce extracellular stimuli into intracellular signals [39]. GPCRs are a primary target for drug discovery.

We obtained an MSA of 940 GPCR sequences used in the statistical coupling analysis by Ranganathan and colleagues [31]. Each sequence has 348 residues. GPCRs can be organized into five major classes, labeled A through E. The MSA that we obtained is from class A; using the GPCRDB [40], we annotate each sequence with functional class information according
to the type of ligand the sequence binds to. The three largest functional classes—Amine, Peptide, and Rhodopsin—have more than 100 sequences. There are 12 other functional classes having less than 45 sequences. There are 66 orphan sequences which do not belong to any family. For prior couplings, we constructed a contact graph network from the 3D structure of a prominent GPCR member, viz. bovine rhodopsin (pdb id 1GZM). We identify 3109 edges as coupling priors using a pairwise distance threshold of 7Å.

<table>
<thead>
<tr>
<th>Residue</th>
<th>Sequence Conservation</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.83</td>
<td>Specific DNA binding</td>
</tr>
<tr>
<td>5</td>
<td>0.62</td>
<td>Specific DNA binding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>0.81</td>
<td>Specific DNA binding</td>
</tr>
<tr>
<td>9</td>
<td>0.58</td>
<td>Unknown</td>
</tr>
<tr>
<td>22</td>
<td>0.45</td>
<td>Unknown</td>
</tr>
<tr>
<td>27</td>
<td>0.64</td>
<td>Nonspecific DNA contact</td>
</tr>
<tr>
<td>30</td>
<td>0.81</td>
<td>Low-affinity Metal Site</td>
</tr>
<tr>
<td>33</td>
<td>0.87</td>
<td>Nonspecific DNA contact</td>
</tr>
<tr>
<td>34</td>
<td>0.71</td>
<td>Low-affinity Metal Site</td>
</tr>
<tr>
<td>37</td>
<td>0.85</td>
<td>Unknown</td>
</tr>
<tr>
<td>42</td>
<td>0.41</td>
<td>Unknown</td>
</tr>
<tr>
<td>58</td>
<td>0.60</td>
<td>Ni2+ site H-bond network</td>
</tr>
<tr>
<td>60</td>
<td>0.86</td>
<td>Close proximity to Ni2+ site</td>
</tr>
<tr>
<td>62</td>
<td>0.83</td>
<td>Close proximity to Ni2+ site</td>
</tr>
<tr>
<td>64</td>
<td>0.38</td>
<td>Nonspecific DNA contact</td>
</tr>
<tr>
<td>65</td>
<td>0.52</td>
<td>Nonspecific DNA contact</td>
</tr>
<tr>
<td>69</td>
<td>0.51</td>
<td>Unknown</td>
</tr>
<tr>
<td>75</td>
<td>0.74</td>
<td>Ni2+ site H-bond network</td>
</tr>
<tr>
<td>109</td>
<td>0.49</td>
<td>Unknown</td>
</tr>
<tr>
<td>114</td>
<td>0.47</td>
<td>Unknown</td>
</tr>
<tr>
<td>116</td>
<td>0.39</td>
<td>Low-affinity Metal Site</td>
</tr>
<tr>
<td>118</td>
<td>0.45</td>
<td>Low-affinity Metal Site</td>
</tr>
</tbody>
</table>
Table 2.1: Important residues for allosteric activity in NikR collected from [6]. Residues are mapped from indices with respect to Apo Nikr (PDB id 1Q5V) to the indices of NikR MSA column. Important residues having conservation greater than 90% are not shown.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>0.62</td>
<td>Nonspecific DNA contact</td>
</tr>
<tr>
<td>121</td>
<td>0.82</td>
<td>Low-affinity Metal Site</td>
</tr>
</tbody>
</table>

2.4.2 Evaluation of Couplings

We evaluate four methods on the NikR and GPCR datasets: the traditional GMRC method proposed by Thomas et al. [28, 32]; GMRC-Inf from this study; GMRC-Inf* (a variant of GMRC-Inf) where the inflated alignment uses only class-based information; and HLCM. We consider three physicochemical properties—polarity, hydrophobicity, and size—of amino acids as classes. Although GMRC discovers couplings in terms of amino acids, we compare our methods with GMRC with respect to the number of discovered important residues (we desire to investigate whether our models can recapitulate important residues identified by previous methods). In Table 2.1, we list 24 important residues for NikR activity from [6] which are not conserved. (We exclude seven important residues for NikR which have a conservation of more than 90%.) Table 3.2 gives comparisons between methods for these two datasets.

Likewise, we identify 47 important residues for the GPCR family from [31]. The support threshold for GMRC and GMRC-Inf is set to 15%; the support threshold and minsupport
Table 2.2: Comparisons of methods for various feature on NikR dataset.

<table>
<thead>
<tr>
<th>Features</th>
<th>GMRC</th>
<th>GMRC-Inf</th>
<th>GMRC-Inf*</th>
<th>HLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Threshold (%)</td>
<td>15</td>
<td>15</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>Num of couplings</td>
<td>80</td>
<td>65</td>
<td>26</td>
<td>51</td>
</tr>
<tr>
<td>Num of important residues (out of 24)</td>
<td>15</td>
<td>11</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>Unique residues in the network</td>
<td>81</td>
<td>61</td>
<td>38</td>
<td>74</td>
</tr>
<tr>
<td>Num of components</td>
<td>11</td>
<td>6</td>
<td>13</td>
<td>23</td>
</tr>
</tbody>
</table>

for HLCM is set to 35% and 2 respectively. (To be more confident about the quality of the model, the support for HLCM is set to a higher value.)

Bradley et al. [6] identify four residues (Res 9, Res 37, Res 62, and Res 118) as highly connected “hubs”. In our models, Res 9 and Res 118 are present, but Res 37 and Res 62 are not present since these residues are highly conserved. Important residues discovered by four methods are shown in Table 2.3. We see that GMRC-Inf and GMRC-Inf* are progressively more strict than GMRC in the number of important residues discovered but GMRC-Inf* has a greater ratio of important residues discovered to the total residues in the network. HLCM provides as good performance as the GMRC method in terms of the important residues but compacts them into a smaller set of couplings.

Table 2.3: Important residues discovered by HLCM, GMRC-Inf, GMRC-Inf*, and GMRC in NikR.

<table>
<thead>
<tr>
<th>Method</th>
<th>Important Residues</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLCM</td>
<td>3, 7, 9, 27, 30, 34, 42, 60, 97, 109, 114, 116, 118, 119, 121</td>
</tr>
<tr>
<td>GMRC-Inf</td>
<td>27, 30, 33, 34, 37, 58, 60, 97, 116, 118, 121</td>
</tr>
<tr>
<td>GMRC-Inf*</td>
<td>3, 5, 27, 33, 37, 42, 60, 116, 121</td>
</tr>
<tr>
<td>GMRC</td>
<td>3, 7, 9, 27, 30, 33, 34, 37, 58, 60, 97, 116, 118, 119, 121</td>
</tr>
</tbody>
</table>
Table 2.4: Classification of GPCR subclasses.

<table>
<thead>
<tr>
<th>Functional Class</th>
<th>Total Sequence</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GMRC</td>
</tr>
<tr>
<td>Amine</td>
<td>196</td>
<td>99.5</td>
</tr>
<tr>
<td>Peptide</td>
<td>333</td>
<td>100</td>
</tr>
<tr>
<td>Rhodopsin</td>
<td>143</td>
<td>98.6</td>
</tr>
</tbody>
</table>

### 2.4.3 Classification Performance

Although our goal is to represent amino acid class-based residues couplings in a formal probabilistic model, we demonstrate that our models can also classify protein sequences. We use the GPCR dataset to assess the classification power of our models. The GPCR datasets has 16 subclasses with, as stated earlier, the three major subclasses being Amine, Peptide, and Rhodopsin. We performed a five-fold cross-validation test for these three major classes. A comparison between our HLCM model and the vanilla GMRC is given in Table 2.4. We see an improved performance for the Amine subclass and a slightly decreased performance for the Rhodopsin subclass.

Recall that there are 66 orphan sequences in GPCR family which are not assigned to any functional class. We apply our model to classify these orphan sequences to any of the three major classes: Amine, Peptide, and Rhodopsin. Toward this end, we build models for the three classes using HLCM method by considering all of the sequences. Of the 66 sequences, 3 are classified to Amine and the rest are classified to the Peptide class. This result is the same as the GMRC result reported in [28].
2.4.4 Finding Coupling Types

We determine the frequency of each class-coupling type for the various models on the NikR dataset. Histograms are shown in Figure 2.9. We see that there are a significant number of class-based residue coupling relationships discovered, although in the case of GMRC-Inf, there are many value-based couplings as well (as expected). Many of the couplings discovered by GMRC-Inf* and HLCM have polarity as one of the properties, but there are interesting differences as well: HLCM identifies a significant number of P-S couplings whereas GMRC-Inf* finds P-P, P-H, and S-S couplings.

![Histograms for class-coupling types on the NikR dataset using three methods: (a) GMRC-Inf, (b) GMRC-Inf*, and (c) HLCM.](image)

2.5 Discussion

Our results on the NikR dataset demonstrate that employing amino acid types is useful for learning couplings and the underlying properties of those couplings. This approach provides us with a way to build an expressive model for residue couplings. We have shown that our extended graphical model is more powerful than the previous graphical model approach of Thomas et al. [28].
A challenging issue with learning coupling is whether our proposed method work for multiple sequence alignments with low-sequence similarities. While learning subsetting context our proposed algorithms accepts only those amino acids or class values that satisfy a subsetting threshold. This approach would prevent adding a spurious coupling into the learned network.

Our use of conditional mutual information as a correlation measure is subject to different biases [30]. Removing possible biases is a direction for future work. A more unifying probabilistic approach for residue couplings would be a factor graph representation since it can capture couplings among more than two residues. A factor graph is a bipartite graph that represents how a joint probability distribution of several variables factors into a product of local probability distributions [41]. Let $G = (\{F, V\}, E)$ be a factor graph, where $F = \{f_1, f_2, \ldots, f_m\}$ is a set of factor nodes and $V = \{v_1, \ldots, v_n\}$ is a set of observed variables. A scope of a factor $f_i$ is set a set of observed variables. Each factor $f_i$ with scope $C$ is a mapping from $\text{Val}(C)$ to $\mathbb{R}^+$. The joint probability distribution of $V$ is as follows:

$$
P(v_1, v_2, \ldots, v_n) = \frac{1}{Z} \prod_{j=1}^{m} f_j(C_j) \tag{2.12}
$$

where $C_j$ is the scope of the factor $f_j$ and the normalizing constant $Z$ is the partition function. Figure 3.3 illustrates a hypothetical residue coupling network for four residues with two factors. Observe how such a model can capture couplings involving more than two residues.

While there are polynomial time algorithm for learning factor graphs from polynomial sam-
Figure 2.10: A hypothetical residue coupling in terms of amino acid classes using a factor graph model.

Such methods require a canonical parameterization which constrains the applicability of factor graphs to learn couplings from an MSA. Canonical parameterizations are defined relative to an arbitrary but fixed set of assignments to the random variable, and it is hard to define such a ‘default sequence’. Hence, newer algorithms need to be developed.
Chapter 3

Higher-Order Residue Couplings in Proteins

3.1 Introduction

Coupling in proteins has garnered attention due to its applications in gaining functional insights into proteins and predicting structures, a central problem in molecular biology. Existing research on coupling primarily focus on identifying couplings between two residues. As more than two residues come close to each other in a 3-D structure of protein, it is interesting to investigate whether higher-order residue couplings exist. In this study, we explore the notion higher-order couplings and present two methods for identifying and expressing such couplings.
Figure 3.1: A toy example demonstrating a 3-order coupling. Eight subsequences for three columns \(i, j,\) and \(k\) in a multiple sequence alignment are shown. Residue columns \(i, j,\) and \(k\) exhibit a 3-order coupling with an enrichment of “AYR” and “PEN”.

Within a higher-order coupling, more than two residues are involved (see Sec. 1.1). Fig. 3.1 illustrates a toy example for a 3-order coupling. In this figure, there are eight subsequences for three columns in an alignment. Pairwise columns show low level of enrichment in terms of the presence of amino acid combinations. If we measure the score with total correlation (see Sec. 3.3), then each of the three pairs displays a score of 0.4. If we consider three columns together, then we observe a high level of enrichment in the columns—the amino-acid combinations “AYR” and “PEN” are dominant and its total correlation score is 0.7.

Motivated by our prior work [28, 33], we develop two probabilistic graphical models (directed and factor graph models) for capturing higher-order residue couplings. Graphical models are useful tool for supporting better investigation, characterization, and design of proteins. Fig. 3.2–3.3 illustrate how a graphical model represents higher-order couplings. Circular nodes in these models denote residues and rectangular nodes represent higher-order couplings between a set of residues. For example, the residues in triplet \((1,2,4)\) in Fig. 3.2 are coupled.

The key contributions of this study are as follows:
1. We investigate whether higher-order residue couplings exist in proteins and answer this question in the affirmative for protein families studied in this paper.

2. We design probabilistic graphical models for capturing higher-order residue couplings. Our models are precise and can be exploited to predicting contacts and assessing the likelihood of protein to be in a family and thus constitute the driver for protein design.

3. This study not only detects higher-order couplings but also presents a way to distinguish the couplings of various orders within a set of residues.

3.2 Related Work

There are lot of research interest in studying different types of couplings in protein. In this section we discuss some of the pertinent studies.

Methods from different domains such as information theory, probabilistic graphical models, statistical analysis have been employed for studying residue interactions (for review see [42, 43]). These methods can be be divided into two groups base on the number of residues involved in a coupling—pairwise residue interaction and multi-residue or higher-order interaction. Again, these methods study two types of interactions based on the proximity of the interacting residues: in-contact interactions or direct couplings [44] and long-distance interactions or indirect couplings [9]. If the distance between two residues are small (<7Å) in the spatial conformation of a protein, then residues are considered within the contact of
each other. Couplings in these methods are learned in various contexts and applications: contact or structure prediction [44 45 46], protein-protein interaction [47], biological insights of structure and functions [9 31], binding specificity [31], and classification of new proteins [28].

Although there has been extensive research, there have been few studies that focus on higher-order couplings in proteins. These studies identify groups of two or more residues that exhibit interactions. Ye et al. propose a method for modeling for higher-order interactions between residues using hypergraphs [48]. In the proposed model, a hyperedge represents a group of correlated residues and edge weights represent the degree of hyperconservation or coupling potential. The model captures in-contact interactions, but can be extended to modeling long-distance interactions between noncontacting residues. Although the model estimates coupling potential for a hyperedge, the contributions of coupling potential of various orders to the total coupling potential of a hyperedge are not distinguished. Clark et al. propose a method for discarding influences of higher-order interactions between residues on a pairwise coupling [49]. Their model with a generalized mutual information identifies higher-order interactions, which are discarded in the subsequent step for improving the quality of direct (in-contact) couplings. A related approach for dealing with multi-residue interactions is to fragment proteins into various portions and measure coevolution between protein fragments [50]. This approach does not truly model higher-order interactions rather it focuses on studying pairwise interactions between fragments of a protein.
3.3 Background

We briefly discuss few concepts from information theory that are used for this study.

3.3.1 Entropy and Mutual Information

Let $X_i$ be a random variable with its finite domain $A$ and $x_i$ be an instance of $X_i$. Unless ambiguity arises, we use the term random variable and variable interchangeably. Entropy of $X_i$ with probability mass function $P(X = x_i)$ is defined as $H(X_i) = -\sum_{x_i \in A} P(X_i = x_i) \log_2 P(X_i = x_i)$. Entropy measures how much we do not know about a variable. It is also known as a measure for disorder or chaos.

The dependency between random variables can be measured in terms of how much of this uncertainty is reduced given the value of another variable. Mutual information is a measure that assesses the dependency (both linear and non-linear) between two random variables [51]. Given two random variables $X_i$ and $X_j$, the mutual information, $I(X_i, X_j)$, is defined as follows:

$$I(X_i, X_j) = H(X_i) - H(X_i | X_j)$$

$$= H(X_j) - H(X_j | X_i)$$

$$= H(X_i) + H(X_j) - H(X_i, X_j)$$

Here $H(X_i | X_j)$ and $H(X_i, X_j)$ are the conditional and the joint entropies respectively. Note
that \( I(X_i, X_j) \) is a symmetric measure.

### 3.3.2 Total Correlation

Total correlation is one of many generalizations of mutual information. This measure, also known as multi-information or multivariate mutual information, is expounded by S. Watanabe \[52\]. Given a set of random variables \( X = \{X_1, X_2, \ldots, X_n\} \) the total correlation \( (C) \) is defined as follows:

\[
C(X) = \sum_{i=1}^{n} H(X_i) - H(X_1, X_2, \ldots, X_n) \tag{3.1}
\]

\[
C_{\text{max}}(X) = \sum_{i=1}^{n} H(X_i) - \max_{X_i} H(X_i)
\]

\[
C_{\text{norm}}(X) = \frac{C(X)}{C_{\text{max}}(X)}
\]

Here \( C_{\text{max}} \) is the maximum value for multi-information and \( C_{\text{norm}} \) is the normalized multi-information. Note that multi-information also captures both linear and nonlinear correlation. Total correlation becomes mutual information with \( n = 2 \).

### 3.3.3 Connected Information

Total correlation captures the total dependency for a set of variables, but does not exhibit contributions from different orders of variables within the set. Schneidman et al. \[53\, 54\].
propose the notion of connected information for decomposing the contributions from different orders of variables to the total correlation. The first term in the right side of Eq. 3.1 is essentially a first-order model that assumes no dependencies between variables, whereas the second term is an n-th order model that considers all possible correlation between variables.

The first-order model, $P^{(1)}(X)$, is the maximum entropy distribution with first-order marginals as constraints, which is essentially the independence model, $P^{(1)}(X) = P_{\text{ind}}(X) = \prod_i P(X_i)$. Independence model is ineffective in describing the interactions between variables. The second-order model, $P^{(2)}(X)$, is the maximum entropy distribution that is consistent with both the first-order marginals and the second-order marginals. Similar to the second-order model, the third-order and other successive order of models can be defined. The n-th order model, $P^{(n)}(X)$, is the maximum entropy distribution consistent with all possible orders of marginals. Given the probability distributions of $X$ for order up to $k$, the connected information for order $k$, $I^{(k)}$, is defined as follows:

$$I^{(k)}(X) = H(P^{(k-1)}(X)) - H(P^{(k)}(X))$$

(3.2)

Estimation of $P^{(k)}(X)$ with $2 \leq k \leq n - 1$, unlike the first-order and n-th order model, is performed in an optimization setting in which we maximize entropy with constraints corresponding to the concerned order (i.e., marginals). As an example, for the second-order,
we search for a \( P(X) \) that maximizes the following objective function.

\[
    L(P(X), \lambda) = -\sum_x P(x) \log_2 P(x) - \sum_i \sum_k \lambda^k_i (P(x_k) - P_{\text{emp}}(x_k)) \\
    - \sum_{i<j} \sum_k \sum_l \lambda^{kl}_{ij} (P(x_k, x_l) - P(x_k, x_l)) - \lambda_0 \left( \sum_x P(x) - 1 \right)
\] (3.3)

The solution to this optimization problem is

\[
    P^{(2)}(X) = \frac{1}{Z} \left( \sum_i \sum_k \lambda^k_i f_i(x_k) + \sum_{i<j} \sum_k \sum_l \lambda^{kl}_{ij} f_{ij}(x_k, x_l) \right),
\] (3.4)

where \( \lambda^k_i \) is the Lagrangian multiplier for \( X_i \) with \( k \)-th value and \( \lambda^{kl}_{ij} \) is the Lagrangian multiplier for a variable pair \((X_i, X_j)\) with \( k \)th and \( j \)th values respectively.

### 3.4 Methods

Let \( S \) be a multiple sequence alignment (MSA) with \(|S|\) sequences of length \( n \). Each column \( i \) in \( S \) corresponds to a random variable \( X_i \). We denote the finite domain of \( X_i \) by \( A \), and let \( x_i \in A \) be a value of \( X_i \). For a protein alignment, \( A \) is a set of 20 amino acids with a gap. The MSA \( S \) then gives a distribution of amino acids for each \( X_i \). We propose two probabilistic graphical models for higher-order residue coupling: HCDG and HCFG. Both of these methods exploit information theoretic measures, more specifically conditional total correlation. The first method uses a directed graphical model, also known as Bayesian
Figure 3.2: A DAG representation for a graph learned by HCDG. The bottom layer represents observed variables $X$ (e.g., residues) and the upper layer denotes hidden factors $Y$.

network, for representing higher-order couplings. The second method employs a factor graph model, which is an undirected graphical model. These methods can be viewed as extensions of pairwise residue couplings with graphical models presented in [28, 33].

### 3.4.1 Higher-Order Couplings with Directed Graphical Models

The first method, HCDG, is based on the notion of Correlation Explanation (CorEx) proposed by Ver Steeg et al. [55]. This method can be viewed as an unsupervised method that take all the variables $X_i$ into account and explains the common correlation or dependency with a hidden layer of factors. The key idea of CorEx is based on the conditional total correlation, which is defined as

$$C(X|Y) = \sum_{i=1}^{n} H(X_i|Y) - H(X|Y)$$  \hspace{1cm} (3.5)
Given $C(X)$ and $C(X|Y)$, we can measure the extent to which $Y$ reduces or explains the dependency in $X$ as follows:

$$C(X;Y) = C(X) - C(X|Y)$$

$$= \sum_{i=1}^{n} I(X_i, Y) - I(X, Y)$$

(3.6)

Here $C(X;Y)$ is not symmetric unlike mutual information. The quantity $C(X;Y)$ is maximized with $C(X|Y) = 0$, and can be seen as ‘common information’ \[56\]. In this setting $Y$ fully explains the correlation in $X$. Moreover, $Y$ can be viewed as a Markov blanket for $X$; thus, $Y$ is represented as the parent of $X$ in a DAG representation of a Bayesian network \[57, 20\]. Fig. 3.2 illustrates a typical plate diagram of this method.

By optimizing Eq. (3.6), we can search for a latent factor $Y$ (e.g., discrete variable with $k$ possible values) that explains the correlation in $X$ (see \[55\] for details). For protein alignments, we learn binary factors. This notion of a hidden factor can be extended to $m$ different hidden factors as follows:

$$\max_{G_j, P(Y_j|X_{G_j})} \sum_{j=1}^{m} C(X_{G_j}; Y_j) \text{ such that } |Y_j| = k, G_j \cap G_{j' \neq j} = \emptyset$$

(3.7)

$$\max_{\alpha, P(Y_j|X)} \sum_{j=1}^{m} \sum_{i=1}^{n} \alpha_{i,j} I(Y_j : X_i) - \sum_{j=1}^{m} I(Y_j : X), \text{ where } \alpha_{i,j} = \mathbb{I}(X_i \in G_j) \in \{0,1\}$$

Eq. (3.7) searches for hidden factors $Y_j$ and its group $G_j$ with a constraint: there is no overlaps between two groups. This constraint does not affect the tractability of the optimization;
Algorithm 3 HCDG(S)

Input: \( S \) (multiple sequence alignment of size \( m \times n \))
Output: \( G \) (a graph that captures couplings in \( S \))

1. \( X = \{X_1, X_2, \ldots, X_n\} \)
2. Initialize \( l \) (number of latent variables, \( Y_j \))
3. Initialize \( k \) (number of possible values for \( Y_j \))
4. Randomly initialize \( P(y|x^{(l)}) \)
5. while Stopping criterion is not satisfied do
6. \hspace{1em} Estimate \( P(y_j) \) and \( P(y_j|x_j) \)
7. \hspace{1em} Calculate \( I(X_i : Y_j) \) from marginals
8. \hspace{1em} Update \( \alpha \)
9. \hspace{1em} Calculate \( P(y|x^{(l)}) \)

therefore, this constraint can be removed (see \[55\] for details). The factors learned by this method captures at least a variable. As coupling is considered a rare event, we prune factors based on their sizes and total correlation scores. Alg. 3 demonstrates the pseudocode for learning a DAG. For each iteration the running time of the algorithm is \( O(mn) \), where \( n \) is number of residues and \( m \) is number of hidden variables. The algorithm is not guaranteed to find the global optimum.

This method has some limitations for capturing couplings. As a variable can participate in at most one factors, there is no flow of dependency between two factors through a common variable. This may limit explaining some biological phenomena such as allosteric communication through coupling. Some of the factors can be of extremely sizes (e.g., size of 1). We aim to remove these limitations in HCDG.
Figure 3.3: Addition of a factor in a graph. a) While adding a 2-order factor the network score depends on the score of $X_i$ and $X_j$, which depends on neighboring nodes and residue groups containing them. b) Addition of a 3-order factor depends on its three nodes with their neighbors and residue groups they belong to.

3.4.2 Higher-Order Couplings with Factor Graphs

Our second method, HCFG, is also based on the notion of conditional total correlation (see Eq. 3.5). This method represents higher-order couplings with a factor graph model, $G = (V, F)$, where each node $v \in V$ corresponds to a random variable $X_v$ (i.e., a column in $S$) and each factor $f \in F$ corresponds to a hidden factor with a set of nodes in $V$. Unless an ambiguity arises we denote each node $v$ with its corresponding random variable $X_v$.

Given an MSA $S$, we infer a factor graph model with HCFG by identifying factor nodes that makes other variables independent. Our algorithm builds the graph in a greedy manner. At each step, the algorithm selects a factor from a set of possible factors which scores the best with respect to the current graph. In this graph, each factor of order $k$ represents a $k$-order coupling between the nodes. A pseudocode for learning a factor graph is shown in Alg. 4.

To measure available dependency between residues, we create candidate groups of residues
from which the method chooses factors of different orders. We can choose groups of equal size (e.g., triplets and quadruples) or we can employ structural priors for selecting groups of residues that are in mutual contact. We consider two residues to be in mutual contact if the distance between two residues is less than 7 Å in the 3-D structure of a protein. This formulation of a candidate groups provides mechanistic explanations for couplings. The score of the graph is given by:

$$S(G = (V, F)) = \sum_{X_v \in V} C(R_{X_v} | N(X_v))$$ (3.8)

where $R_{X_v}$ is residue group with $X_v$ as one its members and $N(X_v)$ is the set of neighboring nodes for $X_v$ in $G$. Due to the limited number of sequences we consider residue groups of size 3 and learn only 2-order and 3-order couplings.

The calculation of conditional total correlation and addition of factors is illustrated in Fig. 3.3. In Fig. 3.3(a), the algorithm considers a factor $(X_i, X_j)$ within a residue group $R_{X_i, X_j, X_k}$ for addition to the graph, where $X_i$ already has a neighbor $X_o$. To assess the importance of the factor $(X_i, X_j)$, we first calculate the score $S(X_i)$ associated with $X_i$ condition on $X_o$ and $X_j$. We then calculate the score $S(X_j)$ associated with $X_j$ condition on $X_i$. Based on $S(X_i)$ and $S(X_j)$, we estimate the reduction of the network score $S$. While conditioning on a node $X_v$, we subset $X_v$ to its most frequent values. We use a subsetting threshold of 10% to maintain the fidelity to the original MSA $S$. Fig. 3.3(b) shows the scenario of adding a triplet $(X_i, X_j, X_k)$ within a residue group $R_{X_i, X_j, X_k}$ for addition to the
graph. Similar to a 2-order factor, we calculate $S(X_i)$, $S(X_j)$, and $S(X_k)$, and estimate the reduction score of the network. We normalize the score reduction with a 3-order factor to compare it with a 2-order factor. If score reduction with a 3-order factor is greater than a score reduction with a 3-order, we add the 3-order factor to $G$. Otherwise, if the reduction score with a 3-order lies within a threshold $\theta$ of the score with a 2-order factor, we choose the 3-order factor with a probability $\alpha$. For rest of the cases, we choose a 2-order factor.

We continue adding best edges as long as stopping criteria is not satisfied. The algorithm can use various stopping criteria. If the difference between network scores in two consecutive iterations is less than a threshold, then the algorithm stops. Another stopping criteria is that if a used-defined number of couplings are added into the network, then algorithm stops. The algorithm can also use both of the stopping criteria together. Algorithm 4 is a heuristic approach. With a prior the running time of each iteration is $O(dn^2)$, where $n$ is the number of residues in a family and $d$ is the maximum number of triplets to which a node belongs. Without a prior a node can belong to $O(n^2)$ number of triplets; thus the running time per iteration is $O(n^4)$.

This methods is robust with multiple sequence alignments with low sequence similarity. While learning subsetting context our proposed algorithms accepts only those amino acids that satisfy a subsetting threshold. This approach would prevent adding a spurious coupling into the learned network.
Algorithm 4 HCFG(\(S, C\))

**Input:** \(S\) (multiple sequence alignment), \(C\) (candidate factors)

**Output:** \(G\) (a graph that captures couplings in \(S\))

1. \(V = \{v_1, v_2, \ldots, v_n\}\)
2. \(F \leftarrow \emptyset\)
3. \(s \leftarrow S(G = (V, F))\)
4. for all \(f \in C\) do
   5. \(s_f \leftarrow s - S(G = (V, \{f\}))\)
5. while stopping criterion is not satisfied do
6.   \(f \leftarrow \arg \max_{f \in C-F} C_f\)
7.   if \(f\) is important then
8.     \(F \leftarrow F \cup \{f\}\)
9.   end if
10.  \(s \leftarrow s - s_f\)
11. for all \(f' \in C - F\) s.t. \(f\) and \(f'\) share a vertex do
12.   \(s_{f'} \leftarrow s - S(G = (V, F \cup \{f'\}))\)

### 3.5 Experimental Results

In this section, we assess our models on protein families, which are demonstrators for evolutionary constraints modeling. Our models can be leveraged to answer questions about couplings, e.g., How do the higher-order models fare compare to other methods in terms of capturing pairwise couplings? Does the learned model capture important couplings in the protein family? Do the higher-order couplings provide any interesting biological insights into the protein family?

#### 3.5.1 Datasets

We learn our models with Nickel repressor protein (NikR) and G-protein coupled receptor (GPCR). A summary of the datasets is listed in Table 3.1.
Table 3.1: Multiple sequence alignments used for model evaluation.

<table>
<thead>
<tr>
<th>Family</th>
<th>Subfamily</th>
<th>Num. of Seq.</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>NikR</td>
<td>Amine</td>
<td>196</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Peptide</td>
<td>333</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Rhodopsin</td>
<td>143</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Nucleotide</td>
<td>41</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Olfactory</td>
<td>41</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Prostanoid</td>
<td>33</td>
<td>348</td>
</tr>
<tr>
<td>GPCR</td>
<td>Hormone</td>
<td>21</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>Orphan</td>
<td>66</td>
<td>348</td>
</tr>
</tbody>
</table>

Nickel receptor protein family

We apply our model on Nickel-responsive transcription factor (NikR), which represses expression of NikABCDE operon in the presence of excessive concentration of $Ni^{2+}$ in a cell (see Sec. 2.4.1 for details). We organized an MSA of the NikR family that has 82 sequences which are used to study allosteric communication in NikR [6]. Each sequence has 204 residues. For HCFG with a structural prior, we use Apo-NikR (pdb id 1Q5V) as a representative member of the NikR family and calculate prior triplets from its contact map. Residue pairs within 7Å of each other are considered to be in contact which gives us 548 triplets as a prior. We use this prior for the analysis to ensure that all identified relationships have direct mechanistic explanations.

G-protein coupled receptors

We also evaluate our models with G-protein coupled receptor (GPCR) data. Sec. 2.4.1 describes properties of GPCR family. We obtained an MSA of 940 GPCR sequences used
in the statistical coupling analysis by Ranganathan and colleagues [31]. Each sequence has 348 residues. GPCRs can be organized into five major classes, labeled A through E. The MSA that we obtained is from class A; using the GPCRDB [40], we annotate each sequence with functional class information according to the type of ligand the sequence binds to. The three largest functional classes—Amine, Peptide, and Rhodopsin—have more than 100 sequences. There are 12 other functional classes having less than 45 sequences. There are 66 orphan sequences which do not belong to any family. For HCFG with structural prior, we constructed a contact graph network from the 3D structure of a prominent GPCR member, viz. bovine rhodopsin (pdb id 1GZM). We identify 1841 triplets as priors using a pairwise distance threshold of 7Å.

### 3.5.2 Evaluation of Couplings

Similar to evaluation for class-based coupling presented in Sec. 2.4.2, we compare our methods against four other methods on the NikR: the traditional GMRC method proposed by Thomas et al. [28, 32]; GMRC-INF, GMRC-INF*, and HLCM presented in [33]. For the proposed three methods in [33], we consider three physicochemical properties—polarity, hydrophobicity, and size—of amino acids as classes. Although GMRC-INF, GMRC-INF*, and HLCM discovers couplings in terms of amino acid classes, we compare our methods with them with respect to the number of discovered important residues (we desire to investigate whether our models can recapitulate important residues identified by previous methods). In Table 2.1, we list 24 important residues for NikR activity from [6] which are not conserved.
Table 3.2: Comparisons of methods for various feature on NikR dataset.

<table>
<thead>
<tr>
<th>Features</th>
<th>GMRC</th>
<th>GMRC-Inf</th>
<th>GMRC-Inf*</th>
<th>HLCM</th>
<th>HCFG</th>
<th>HCDG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Threshold (%)</td>
<td>15</td>
<td>15</td>
<td>35</td>
<td>35</td>
<td>10</td>
<td>NA</td>
</tr>
<tr>
<td>Num of couplings/factors</td>
<td>80</td>
<td>65</td>
<td>26</td>
<td>51</td>
<td>50</td>
<td>23</td>
</tr>
<tr>
<td>Num of important residues (out of 24)</td>
<td>15</td>
<td>11</td>
<td>9</td>
<td>15</td>
<td>16</td>
<td><strong>18</strong></td>
</tr>
<tr>
<td>Unique residues in the network</td>
<td>81</td>
<td>61</td>
<td>38</td>
<td>74</td>
<td>104</td>
<td>85</td>
</tr>
</tbody>
</table>

(We exclude seven important residues for NikR which have a conservation of more than 90%.)

Table 3.2 gives comparisons between methods for NikR. Table 3.2 shows that HCDG and HCFG perform better compared to other methods in terms of capturing important residues. Note that the current settings of HCDG limits overlaps between factors. Removal of this restriction could provide a sparse graph. We observe that HCFG tends to capture more residues in the graph; thus, gives a more spreader graph compared to other methods. The support threshold for GMRC and GMRC-Inf is set to 15%; the support threshold and minsupport for HLCM is set to 35% and 2 respectively; and the support threshold for HCDG and HCFG is set to 10%. For HCDG, we use 2 and 8 as minimum and maximum factor size respectively.

Likewise, we identify 47 important residues for the GPCR family from [31]. Table 3.3 shows performance of HCDG with different expected number of factors. We observe that if we set the expected number of hidden factors to 70 or more, then the coupled factors capture around half of the important residues of GPCR family.
Table 3.3: Analysis of couplings with HCDG on GPCR protein family.

<table>
<thead>
<tr>
<th>Expected num hidden factors</th>
<th>Num of filtered factors as couplings</th>
<th>Num of unique residue</th>
<th>Num important res</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>24</td>
<td>123</td>
<td>20</td>
</tr>
<tr>
<td>75</td>
<td>25</td>
<td>116</td>
<td>19</td>
</tr>
<tr>
<td>80</td>
<td>28</td>
<td>130</td>
<td>17</td>
</tr>
<tr>
<td>85</td>
<td>26</td>
<td>128</td>
<td>25</td>
</tr>
<tr>
<td>90</td>
<td>36</td>
<td>164</td>
<td>19</td>
</tr>
<tr>
<td>95</td>
<td>29</td>
<td>127</td>
<td>24</td>
</tr>
</tbody>
</table>

Unlike HCFG, we do not provide any structural priors to HCDG. We analyze the spatial distance between residues within learned factors with HCDG. We observe that some factors containing residues that are close to each other in the 3-D structure. For the result with HCDG shown in Table 3.2, we notice that some of the factors capture residues that lie within 7Å to each other. For example, within the factor (86, 164, 166), the residues in pairs (86, 166) and (164, 166) are in contact; for the factor (49, 70, 73, 76), the residues in pairs (70, 73) and (73, 76) reside close to each other.

### 3.5.3 Decomposition of Higher-Order Couplings

Motivated by the approach in [54], we use connected information (see Sec. 3.3.3) for achieving an insight into the learned 3-order couplings with our models. Connected information allow us to decompose the total correlation for a set of residues into its constituents, which are connected information of different orders. Table 3.4 exhibits decompositions of three triplet factors learned with HCFG for NikR family. We notice that the connected information of order-2 contribute more than the connected information of order-3 to the total correlation.
Table 3.4: Analysis of higher-order couplings with connected information on NikR protein family.

<table>
<thead>
<tr>
<th>Triplet coupling</th>
<th>Total correlation</th>
<th>$I^{(2)}$</th>
<th>$I^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>71, 73, 74</td>
<td>0.30</td>
<td>0.18</td>
<td>0.12</td>
</tr>
<tr>
<td>49, 50, 51</td>
<td>0.25</td>
<td>0.15</td>
<td>0.10</td>
</tr>
<tr>
<td>143, 144, 145</td>
<td>0.15</td>
<td>0.074</td>
<td>0.076</td>
</tr>
</tbody>
</table>

for first two triplets, whereas the connection information of order-3 contribute more than the order-2 to the total correlation for the third triplet. We conclude that triplet couplings learned in our methods are meaningful as it contains both 2-order and 3-order interactions between three residues.

3.6 Discussion

Our models based on probabilistic graphical models can capture higher-order couplings and have advantages over traditional approaches to representing coupling: these models capture independence, expose essential constraints of a family, allow integration of priors (structural and functional), and enables prediction and inference. Results on both NikR and GPCR families demonstrate that higher-order couplings may provide functional insight into a protein family which may not be achieved with pairwise couplings. The presented approaches advance the ways to build a more powerful and expressive model for residue couplings. In future, we aim to improve our models that would yield sparse representation of higher-order couplings.
Chapter 4

Improved Multiple Sequence Alignments using Coupled Patterns

4.1 Introduction

Multiple sequence alignment (MSA) of biological sequences is a classical approach to understand evolutionary constraints. It has been said that “one or two homologous sequences whisper, ..., a full [MSA] shouts out loud” [58]. There is a plethora of MSA algorithms exists, with origins ranging from discrete algorithms [59] to probabilistic models, such as HMMs [60].
4.1.1 Isn’t MSA a Solved Problem?

Although sequence alignment has become a widely deployed tool in bioinformatics, practically every MSA algorithm (e.g., ClustalW [59], Muscle [61], T-Coffee [62], and more) is designed to model and expose conservation, which although being a key evolutionary constraint, does not capture the richness of how sequences evolve and diverge. In a typical alignment (e.g., Fig. 4.1 (b)), conservations (e.g., column 4) are manifest and couplings (e.g., between column 2 and 8) are obscured. In fact, it is often accepted practice for biologists to ‘hand tweak’ such an alignment to incorporate structural information about sequences and thus obtain a better alignment.

Such tweaking is still somewhat of a black art and requires significant domain expertise. We were motivated to design an automated approach to better expose couplings in an MSA; but in doing so, our approach also improves MSAs according to traditional measures of assessment.
4.1.2 Contributions

- We present Alignment Refinement by Mining Coupled Residues (ARMiCoRe), a pattern mining approach to the problem of multiple sequence alignment. Using frequent episode mining as a foundational basis, we define the notion of a coupled pattern that elucidates covarying residues. Such coupled patterns are inferred using a levelwise approach and subsequently ‘tiled’ using a max-flow algorithm. The tiling is then used to direct the adjustment of a conservation based alignment to capture covarying residues.

- ARMiCoRe can be viewed as a novel application of pattern set discovery [63] where the goal is not just to mine interesting patterns (which is the purview of pattern discovery) but to select among them to optimize a set-based measure. ARMiCoRe can be used to tweak alignments from any existing algorithm, to better expose couplings or correlated mutations.

- As multiple sequence alignment is an established topic in bioinformatics, we subject ARMiCoRe to a thorough experimental evaluation involving 108 protein families. We identify selective superiorities of ARMiCoRe and demonstrate situations where it outperforms state-of-the-art MSA algorithms.

A preliminary version of this chapter is published in the proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine [64] while a complete version is available in the IEEE/ACM Transactions on Computational Biology and Bioinformatics [65].
4.2 Related Work

Multiple sequence alignment has been studied extensively for the past several decades (see [66, 67] for reviews). A rich set of features exist to classify MSA algorithms. These approaches fall broadly into two categories of alignment algorithms: global alignment vs. local alignment algorithms. Global alignment algorithms (e.g., ClustalW [59], MUSCLE [61], T-coffee [62], MAFFT [68], and ProbCons [69]) match sequences over their full lengths, whereas local alignment algorithms (e.g., DIALIGN [70], DIALIGN-T [71], and POA [72]) aim to align only the most similar regions between sequences. Local alignment is appropriate for sequence families where well-conserved regions are surrounded by variable regions. A second way to classify algorithms is in terms of the objective function (e.g., sum of pairs score, entropy, circular sum) used to identify the highest scoring alignment [67]. Finally, MSA algorithms can be classified based on their underlying optimization scheme: exact algorithms, progressive algorithms, and iterative algorithms. An exact algorithm attempts to simultaneously align all of the sequences and find an optimal alignment using an objective function [73]. The underlying problem has been proved to be NP-complete [74] and, hence, impractical for large numbers of sequences.

4.2.1 Progressive and Iterative Algorithms:

Heuristic approaches to MSA are either progressive or iterative algorithms. Progressive alignment algorithms (e.g., ClustalW [59] and T-Coffee [62] and LAGAN [75]), typically more
appealing, involve building a guide tree based on sequence similarity and progressively aligning sequences following the order of the guide tree. Variants on progressive alignment typically use guide tree reestimation, modifying objective functions, and/or post-processing [67]. In guide tree reestimation, algorithms compute new distance matrices based on the initial MSA produced by progressive alignment, and the revised distance matrix is used to create a new guide tree. MAFFT [68], MUSCLE [61], PRIME [76], PRRP [77], MULTAN [78], and PROMALS [79] use this approach. Methods that modify the objective function are referred to as consistency-based methods, e.g., T-Coffee [62], DIALIGN [70], ProbCons [69], PCMA [80], and PROMALS [79]. The third variant involves post-processing, also known as iterative algorithms. In this approach, an alignment is first produced rapidly and then refined through a series of iterations until no more improvements can be made [67]. Examples are MUSCLE [61] and DIALIGN [70].

4.2.2 Probabilistic Algorithms

Probabilistic algorithms approach MSA by modeling different aspects: evolutionary models of indels, profile models, and hybrid models that combine probabilistic models with progressive alignment techniques. ProbCons [69] is a well known example that uses maximum expected accuracy scoring to infer a model and is especially useful for divergent sequences. A second example [60] uses a pair of HMMs as the scoring strategy.
4.2.3 Constraint-based Algorithms

These approaches (a.k.a. segment-based alignment algorithms) improve alignment quality by searching and incorporating information about homologs, conserved motifs/domains, and expert-supplied feedback about local similarity. Examples are COBALT [81], DIALIGN [70], DbClustal [82], and PROMALS [79].

As rich as the above landscape of MSA algorithms is, none of the above algorithms use covariation as a property to align sequences. Coupling is often viewed as a feature that ‘comes out’ of an alignment as opposed to a criterion or driver for computing the alignment. A very recent work, published in 2010 [83], is the lone exception which uses mutual information to detect coupled residues, and uses constraint programming to realign sequences. As we will show, ARMiCoRe captures not just coupled residues but the richer class of coupled patterns that tile the entire set of sequences; this greater expressiveness leads to improved MSAs, both in terms of exposing couplings, and in terms of traditional metrics of assessment (see Section 4.5).
4.3 Formulation

We are given a collection \( S = \{s_1, \ldots, s_n\} \) of \( n \) aligned sequences (or strings), each of length \( m \), over a finite alphabet. As shown in Fig. 4.1 (b), the sequences in \( S \) are assumed to have been aligned by a standard MSA method that typically favors conservation (and thus might contain gaps). Each sequence \( s_i, i = 1, \ldots, n \), can hence be expressed as \( s_i = \langle E^i_1, \ldots, E^i_m \rangle \), \( E^i_j \in \mathcal{E} \cup \{\varphi\}, j = 1, \ldots, m \), where \( \mathcal{E} \) denotes a finite alphabet and \( \varphi \) is the gap symbol. In the case of DNA sequences, \( \mathcal{E} = \{A, C, T, G\} \), whereas for protein sequences, \( \mathcal{E} \) comprises the 20 amino acid residues. We can even for instance denote amino acids by their physico-chemical properties so that the set of 20 amino acids can be reduced to a smaller set of properties.

**Definition 1** An indexed pattern \( \alpha \) (of size \( \ell \)) is defined by a pair of \( \ell \)-length sequences, \((\langle A^\alpha_1, \ldots, A^\alpha_\ell \rangle, \langle \delta^\alpha_1, \ldots, \delta^\alpha_\ell \rangle)\), where each \( A^\alpha_j \in \mathcal{E} \), \( \delta^\alpha_j \in \mathbb{Z}^+ \), \( j = 1, \ldots, \ell \), and \( \delta^\alpha_{j+1} > \delta^\alpha_j \), \( j = 1, \ldots, (\ell - 1) \). We refer to \( \langle \delta^\alpha_1, \ldots, \delta^\alpha_\ell \rangle \) as the sequence of positions over which \( \alpha \) is defined.

The semantics of an indexed pattern \( \alpha \) is essentially that in a sequence \( s \) where \( \alpha \) is said to occur, we expect that \( A^\alpha_j \) will appear at position \( \delta^\alpha_j \) (or very close to it) for every \( 1 \leq j \leq \ell \).

**Definition 2** A sequence \( s = \langle E_1, \ldots, E_m \rangle \) is said to contain an \( \epsilon \)-approximate occurrence of indexed pattern \( \alpha \) if there exists a map \( h : \{1, \ldots, \ell\} \to \{1, \ldots, m\} \), strictly increasing, such that \( \forall j, 1 \leq j \leq \ell, E_{h(j)} = A^\alpha_j \) and \( |h(j) - \delta^\alpha_j| \leq \epsilon \).

**Example 1** \( \alpha = (\langle A, E, M, C \rangle, \langle 5, 9, 15, 20 \rangle) \) is an indexed pattern of size \( \ell = 4 \). An example
sequence $s$ that contains an $\epsilon$-approximate occurrence of $\alpha$ is shown below (for $\epsilon = 1$). Note that occurrences of symbols $A$, $E$, $M$ and $C$ can be found within 1 position of the locations 5, 9, 15 and 20 respectively.

$$s = \langle \text{KFF KR A C E PT DA I P M E PH EM C PE} \rangle$$

**Definition 3** The $\epsilon$-support of an indexed pattern $\alpha$ over the collection $S$ of sequences, denoted $f_\epsilon(\alpha)$, is the number of sequences in $S$ that contain at least one $\epsilon$-approximate occurrence of $\alpha$; the corresponding set of $\epsilon$-supporting sequences is denoted by $U_\epsilon(\alpha) \subseteq S$, $f_\epsilon(\alpha) = |U_\epsilon(\alpha)|$.

**Definition 4** A coupled pattern, $\psi$, of size $k$ is defined as a $k$-tuple, $(\alpha_1, \ldots, \alpha_k)$, where each $\alpha_i$, $i = 1, \ldots, k$ (referred to as a constituent of $\psi$) is an indexed pattern over a common sequence of positions $\langle \delta_1, \ldots, \delta_\ell \rangle$. The $\epsilon$-support of $\psi$ over a collection $S$ of sequences, denoted $F_\epsilon(\psi)$, is defined as the total number of $\epsilon$-supporting sequences of its constituents found in $S$, i.e., $F_\epsilon(\psi) = |\cup_{\alpha_i \in \psi} U_\epsilon(\alpha_i)|$.

**Example 2** Consider the collection of sequences, $S = \{s_1, \ldots, s_8\}$, defined in Figure 4.2. $\psi = (\alpha_1, \alpha_2)$ is an example coupled pattern of size 2, where $\alpha_1 = (\langle H, L, F, K \rangle, \langle 5, 9, 15, 20 \rangle)$ and $\alpha_2 = \langle A, E, M, C \rangle, \langle 5, 9, 15, 20 \rangle$ are indexed patterns over the same sequence of positions $\langle 5, 9, 15, 20 \rangle$. The $\epsilon$-support of $\psi$ over $S$, for $\epsilon = 1$, is $F_1(\psi) = 8$.

Our main intuition here is that when there is enough evidence for a coupled pattern $\psi$...
in a given data set $S$, the associated sequence of positions $(\delta_1, \ldots, \delta_\ell)$ are *coupled* across multiple sequences of $S$, in the sense that, mutations in one position are accompanied by corresponding mutations in the others. In *Example 4.2*, mutations of $H$ to $A$ in position 5, would be accompanied by three other mutations, namely, $L$ to $E$ in position 9, $F$ to $M$ in position 15 and $K$ to $C$ in position 20. To facilitate the detection and measurement of the evidence for a coupled pattern, we define the notion of $\tau$-coverage with respect to the pattern’s $\epsilon$-supporting sequences.

**Definition 5** Let $S$ be a given collection of sequences over $\mathcal{E} \cup \{\varphi\}$. Consider a coupled pattern $\psi = (\alpha_1, \ldots, \alpha_k)$ and its corresponding sets, $\mathcal{U}_\epsilon(\alpha_i), i = 1, \ldots, k$, of $\epsilon$-supporting sequences. The $\tau$-coverage of $\psi$ in $S$ with respect to its $\epsilon$-supporting sequences, denoted $\Gamma_\epsilon(\psi, \tau)$, is defined as follows:

$$\Gamma_\epsilon(\psi, \tau) = \max_{\mathcal{D}_1, \ldots, \mathcal{D}_k} \sum_{i=1}^k |\mathcal{D}_i|$$

where $\mathcal{D}_i \subset S$, $i = 1, \ldots, k$, such that the following hold: $\mathcal{D}_i \subset \mathcal{U}_\epsilon(\alpha_i)$, $\mathcal{D}_i \cap \mathcal{D}_j$ is empty for $i \neq j$, and $|\mathcal{D}_i| \geq \tau$.

Essentially, we want to compute mutually exclusive sets of $\epsilon$-supporting sequences for each of the $k$ constituents of $\psi$, such that each mutually exclusive set contains at least $\tau$ sequences, while the total number of distinct sequences in these sets is maximized.

**Example 3** For the same example as before, with $\epsilon = 1$, we get the following sets of $\epsilon$-
supporting sequences for $\alpha_1$ and $\alpha_2$: $U_{\epsilon}(\alpha_1) = \{s_1, s_2, s_3, s_4, s_5\}$ ($f_1(\alpha_1) = 5$) and $U_{\epsilon}(\alpha_2) = \{s_5, s_6, s_7, s_8\}$ ($f_1(\alpha_2) = 4$). Setting $D_1 = \{s_1, s_2, s_3, s_4\}$ and $D_2 = \{s_5, s_6, s_7, s_8\}$ we get the 4-coverage of $\psi$ with respect to its 1-supporting sequences to be $\Gamma_1(\psi, 4) = 8$.

There are two main challenges in the detection and use of coupled patterns for improving multiple sequence alignment. First, given a data set $S$ of (approximately aligned) sequences, we need to find coupled patterns which have high $\tau$-coverage over $S$. Second, we need to use the high-coverage coupled patterns discovered to improve the MSA relative to the original alignment in $S$.

**Problem 1 (Mining Coupled Patterns)** Consider a data set $S$ of $m$-length sequences over $E \cup \{\varphi\}$ and a fixed sequence of position indices, $\langle \delta_1, \ldots, \delta_\ell \rangle$. Given user-defined parameters, $\epsilon$, $K$ and $\tau$ (all non-negative integers) find a coupled pattern of size $k \leq K$ over $\langle \delta_1, \ldots, \delta_\ell \rangle$ which maximizes $\tau$-coverage with respect to its $\epsilon$-supporting sequences in $S$.

The MSA realignment problem can then be stated as follows.

**Problem 2 (MSA Realignment)** Given a data set $S$ of $m$-length sequences over $E \cup \{\varphi\}$ and a set of coupled patterns $\Psi = \{\psi\}$ in $S$ each of which has $\tau$-coverage of $\Gamma_\epsilon(\psi, \tau) = \gamma$ over $\epsilon$-supporting sequences, find a realignment $S'$ of the sequences in $S$ where all patterns in $\Psi$ have a $\tau$-coverage of $\Gamma_\epsilon'(\psi, \tau) \geq \gamma$ for $\epsilon' < \epsilon$.

In the above formulation, note that we require coupled patterns discovered in the original (approximate) alignment to still be manifest in the new alignment, but in a more obvious
Figure 4.3: (a) Clustering of amino acids proposed in [4]. (b) This figure describes window constraints. While looking for similar residue within a window the algorithm does not go beyond a conserved residue in a (semi)conserved column so that the (semi)conserved column is not distorted in the realignment process.

manner. Ideally $\epsilon' = 0$ (which is the situation for the example pattern in Fig. 4.1 (d)) but in practice we aim to obtain $\epsilon' < \epsilon$.

4.4 Algorithms

In this section, we present ARMiCoRe, a new method for aligning multiple sequences based on coupling relationships that may exist between residues found in two or more sequence positions. The method consists of two main steps. We start by discovering high-support coupled patterns over various choices of position sequences (described in Sec. 4.4.1). Finally, in Sec. 4.4.3 we derive an alternative alignment $S'$ for $S$ based on both the original ungapped sequences and the just-discovered coupled patterns.
Algorithm 5 \textsc{Cp-Miner}(S, \Psi^\ell, \tau_d, \tau, \epsilon, K)

\textbf{Input:} A set of aligned sequences $S = \{s_1, s_2, \ldots, s_n\}$, a set of frequent coupled patterns $\Psi^\ell$ of size $\ell$, dominant residue conservation threshold $\tau_d$, block coverage threshold $\tau$, column-window parameter $\epsilon$, maximum size of a coupled pattern, $K$.

\textbf{Output:} A set of frequent coupled patterns $\Psi^{\ell+1}$ of size $\ell + 1$.

1. $\Psi^{\ell+1} \leftarrow \phi$
2. $\mathcal{C}^{\ell+1} \leftarrow \textsc{Candidate-Gen}(\Psi^\ell)$
3. $\Psi^{\ell+1}_1 \leftarrow \{\psi : \psi_{\text{dom}} = \{\alpha\}, \forall \alpha \in \mathcal{C}^{\ell+1}\}$
4. for $\psi \in \Psi^{\ell+1}_1$ do
5. \hspace{0.5cm} $\alpha \leftarrow \psi_{\text{dom}}$ \hspace{0.5cm} $\triangleright$ dominant indexed pattern.
6. \hspace{0.5cm} $S^+ \leftarrow \{s_i : s_i \text{ has an } \epsilon\text{-approx. occurrence of } \alpha\}$
7. \hspace{0.5cm} if $|S^+| \geq n\tau_d$ then
8. \hspace{1.0cm} $S^- \leftarrow S - S^+$
9. \hspace{0.5cm} $\mathcal{I} \leftarrow \forall\epsilon\text{-approximate indexed patterns from } S^-$
10. \hspace{0.5cm} $\mathcal{I}' \leftarrow \{\alpha : f_\epsilon(\alpha) \geq \tau, \forall \alpha \in \mathcal{I}\}$
11. \hspace{0.5cm} if $\mathcal{I}' \neq \phi$ and $|\mathcal{I}'| \leq K$ then
12. \hspace{1.0cm} $\psi \leftarrow \psi \cup \mathcal{I}'$
13. \hspace{0.5cm} if $\psi$ is significant then
14. \hspace{1.0cm} $\Psi^{\ell+1} \leftarrow \Psi^{\ell+1} \cup \psi$
15. return $\Psi^{\ell+1}$

Algorithm 6 \textsc{Candidate-Gen}(S, \Psi^\ell)

\textbf{Input:} A set of frequent coupled patterns $\Psi^\ell$ of size $\ell$.

\textbf{Output:} A set of indexed patterns $\mathcal{C}^{\ell+1}$ of size $\ell + 1$.

1. $\mathcal{C}^{\ell+1} \leftarrow \phi$
2. $\mathcal{A}^\ell \leftarrow \{\alpha : \alpha = \psi_{\text{dom}}, \forall \psi \in \Psi^\ell\}$ \hspace{0.5cm} $\triangleright$ $\psi_{\text{dom}}$ denotes an indexed pattern of the most frequent residue.
3. for all $\alpha_i, \alpha_j \in \mathcal{A}^\ell$ do
4. \hspace{0.5cm} if there is a prefix match of length $\ell - 1$ between $\delta^{\alpha_i}$ and $\delta^{\alpha_j}$ then
5. \hspace{1.0cm} $\alpha_k \leftarrow \text{Merge}(\alpha_i, \alpha_j)$
6. \hspace{0.5cm} for all $\alpha_t \in \mathcal{A}^\ell$ and $\alpha_k$ containing $\alpha_t$ do
7. \hspace{1.0cm} $\alpha_{k_{ab}}^{\alpha_t} \leftarrow \alpha_t$ \hspace{0.5cm} $\triangleright$ listing subpatterns
8. $\mathcal{C}^{\ell+1} \leftarrow \mathcal{C}^{\ell+1} \cup \alpha_k$
9. return $\mathcal{C}^{\ell+1}$

4.4.1 Discovering Coupled Patterns

The first step of ARMiCoRe is to choose the sequence positions over which to mine coupled patterns. Then standard level-wise methods (Apriori) are used to discover coupled patterns.
(restricted to the chosen sequence positions) with sufficient support (cf. Sec. 4.4.1). While level-wise searching for coupled patterns ARMiCoRe looks for patterns that have at most \( K \) constituents ignoring \( \tau \)-coverage (cf. Sec. 4.4.1). Then ARMiCoRe applies a statistical significance test to filter out uninteresting coupled patterns (cf. Sec. 4.4.1). This gives us the pattern set, \( \Psi^\ell = \{ \psi_1, \ldots, \psi_{|\Psi|} \} \), of \( \ell \)-size indexed patterns, each with support at least \( \tau \), each has at most \( K \) constituents, and each defined over a common sequence of positions, \( \langle \delta_1, \ldots, \delta_\ell \rangle \). Each subset of indexed patterns in \( \psi \) can thus be a potential candidate for a \( \tau \)-coverage coupled pattern. Finally, ARMiCoRe applies a max-flow approach to get the \( \tau \)-coverage of each \( \psi \) (cf. Sec. 4.4.1).

A lower-bound \( \tau \) on the sizes \( |D_i| \) of the blocks corresponding to each constituent of a coupled pattern (see Definition 5) automatically enforces an upper-bound \( \left\lfloor \frac{n}{\tau} \right\rfloor \) on the size, \( k \), the coupled pattern. At first, it might appear as if the user only needs to prescribe \( \tau \) to detect interesting patterns (since an upper-bound on \( k \) is implied). However, we have observed that in the couplings that are already known in biological data sets, the number of constituents are typically far fewer than \( \left\lfloor \frac{n}{\tau} \right\rfloor \). Hence, in our framework, the user must specify both an upper-bound \( K \) for \( k \) as well as a lower-bound \( \tau \) on the block-sizes \( |D_i| \) of coupled patterns.

We now describe the steps in ARMiCoRe for finding a subset of indexed patterns that implies a coupled pattern, of size at most \( K \), and which maximizes the \( \tau \)-coverage over its \( \epsilon \)-supporting sequences. The main hardness in the problem arises from having to maximize coverage with a \( \tau \) constraint while restricting the number of constituent patterns to no more
than $K$. Hence, we decouple the two problems and show that the individual problems can be solved efficiently. Specifically, we show that by ignoring the $\tau$ constraint, the problem of maximizing coverage is a sub-modular function-maximization problem with cardinality constraint. We propose Algorithm 5 for generating all possible coupled patterns of size at most $K$. On the other hand, after selecting coupled patterns of size at most $K$, maximizing coverage with the $\tau$ constraint reduces to a max-flow problem.

**Level-wise Coupled Pattern Mining**

Our basic idea here is to organize the search for coupled patterns around the (semi) conserved columns of the current alignment. Level 1 patterns are comprised of individual columns, level 2 patterns are comprised of pairs of level 1 patterns, and so on.

For choosing a (semi) conserved column, we employ a *dominant residue conservation threshold* $\tau_d$ (see Line 7 of Algorithm 3). We use class-based conservation so that amino acid residues that have similar physico-chemical properties are considered conserved. Class-based conservation can be estimated using the Taylor diagram or by k-means clustering of substitution matrices such as Blosum62. We have explored both approaches and found the latter to work better (with a setting of 7 non-overlapping clusters)(see Fig. 4.3a).

Amino acids in and around the semi-conserved columns (to within a window length of $\epsilon$) are organized into positive and negative sets of sequences describing the dominant combination and other, non-dominant, ones (see Fig. 4.4 (left)). While increasing the size of both the
dominant and nondominant patterns for a column by searching for similar residues within a window for that column, the algorithm restricts itself to not go beyond a (semi)conserved column if it encounters any such column within the window. For example, in Fig. 4.3b the column 5 is semiconserved and the residue ‘H’ is the dominant residue in this column as it is the most frequent residue. The residue ‘H’ at position 7 of sequence 2 is a candidate for extending the dominant pattern at column position 5. As the column position 6 is almost fully conserved for residue ‘A’, the inclusion of ‘H’ at position 7 of sequence 2 for the dominant pattern at column position 5 may destroy the conservation of column 6 in the realignment process. So the algorithm does not include ‘H’ at position 7 of sequence 2 as a dominant residue for column 5. On the other hand, the algorithm will include the residue ‘H’ at position 6 of the sequence 6 as a dominant residue for column 5 since this inclusion does not destroy the conservation of column 6. As we construct level-2 and greater patterns, we take care to ensure that $\epsilon$ does not yield window lengths that cross another semi-conserved
column.

**High \(\epsilon\)-support using at most \(K\) Constituents**

We now present the approach taken by ARMiCoRe to solve the problem of maximizing coverage by enforcing only the upper-bound \(K\) (user-defined) on the number of constituents of \(\psi\) while ignoring the \(\tau\) constraint. We will test for \(\tau\)-coverage later as a post-processing step (see Sec. 4.4.1). Note that at \(\tau = 0\), \(\tau\)-coverage is same as \(\epsilon\)-support, and this can be shown to be both monotonic and sub-modular with respect to its constituents. That is, if \(A\) and \(B\) are two subsets of \(\psi\), such that \(A \subset B\), then it can be shown that: \(\Gamma_\epsilon(A \cup \alpha, 0) \geq \Gamma_\epsilon(A, 0)\), and, \(\Gamma_\epsilon(A \cup \alpha, 0) - \Gamma_\epsilon(A, 0) \geq \Gamma_\epsilon(B \cup \alpha, 0) - \Gamma_\epsilon(B, 0)\). Consequently, we can use a greedy algorithm which guarantees a \((1 - \frac{1}{e})\)-approximate solution [85]. In other words, we would find a subset of \(\psi\) whose \(\epsilon\)-support (or 0-coverage) is within a factor of \((1 - \frac{1}{e})\) of the optimal subset.

**Significance Testing of Coupled Patterns**

For level-2 patterns and greater, we perform a 2-fold significance test, the first focusing on the dominant pattern and the second focusing on the non-dominant patterns. For the dominant pattern, we compute the probability, and thus the \(p\)-value, of encountering the dominant pattern given the column marginals. For the non-dominant patterns, we conduct a standard enrichment analysis using the hypergeometric distribution to determine if the symbols in the non-dominant pattern are over-represented.
Checking \( \tau \)-coverage using Max-Flow

Once we have generated \( \psi \) with high \( \epsilon \)-support we proceed to check if a non-zero \( \tau \)-coverage is feasible (Recall that the coverage will either be zero or the full \( \epsilon \)-support corresponding to the chosen subset of \( \psi \)). This problem reduces to a standard max-flow problem for which efficient (poly-time) algorithms exist. We now present the reduction of this problem to max-flow (see Fig. 4.5).

Let \( \mathcal{G} = (V, E) \) be a network with \( v_s, v_t \in V \) denoting the source and sink of \( \mathcal{G} \) respectively. In addition to \( v_s \) and \( v_t \), there is a unique node in \( V \) corresponding to each indexed pattern \( \alpha_i \in \psi \) and also to each sequence \( s_j \in S \), i.e., \( V = \{v_s, v_t\} \cup \psi \cup S \). Three kinds of edges are in set \( E \):

1. \( e_{si} \in E \), representing an edge from the source node \( v_s \) to the pattern node, \( \alpha_i \in V \).
We will have $e_{\ast i} \in E$, $\forall \alpha_i \in \psi$

2. $e_{j\#} \in E$, representing an edge from the sequence node $s_j \in V$ to the sink node $v_{\#}$. We will have $e_{j\#} \in E$, $\forall s_j \in S$

3. $e_{ij} \in E$, representing an edge from pattern node $\alpha_i \in V$ to the sequence node $s_j \in S$, whenever the algorithm assigns $s_j$ to $D_i$ (see Definition 3). We will have $e_{ij} \in E$, $\forall \alpha_i \in \psi, s_j \in S$ such that $s_j$ is assigned to the block $D_i$ that corresponds to the $i^{th}$ pattern $\alpha_i \in \psi$.

For any edge $e \in E$, let $LB(e)$ and $UB(e)$ denote, respectively, the lower and upper bounds on the capacity of edge $e$. Given a coupled pattern $\psi$, the computation of its $\tau$-coverage, $\Gamma_{\epsilon}(\psi, \tau)$, reduces to the computation of max-flow for the network $G$ under the following capacity constraints:

1. $LB(e_{\ast i}) = \tau$, $UB(e_{\ast i}) = \infty$, $\forall \alpha_i \in \psi$

2. $LB(e_{j\#}) = 0$, $UB(e_{j\#}) = 1$, $\forall s_j \in S$

3. $LB(e_{ij}) = 0$, $UB(e_{ij}) = 1$, $\forall \alpha_i \in \psi, s_j \in S$

We can now use any max-flow algorithm, such as [86, 87] to obtain the max-flow in $G$ subject to the stated capacity constraints. The flow returned will give us $\Gamma_{\epsilon}(\psi, \tau)$. 
4.4.2 Complexity Analysis

The runtime for finding all possible coupled patterns depends on the number of sequences \( n \), the alignment length \( m \), the column-window threshold \( \epsilon \), and the maximum size of the indexed pattern \( \ell \). Let \( p \) be the number of semi-conserved columns found in level 1 indexed pattern mining. Then the running time for generating all possible coupled patterns is \( O(nm + l(p^3 + lp^2n\epsilon)) \). Since \( p \sim O(m) \), the running time is \( O(l(m^3 + lm^2n\epsilon)) \). Finding a \( \tau \)-coverage coupled pattern depends on the number of nodes \( O(n + K) \) and the number of edges \( q \) in the max-flow network for which the running time is \( O((n + K)q \log((n + K)^2/q)) \). [87].

4.4.3 Updating the Alignment

There are various ways to adjust the given alignment. One strategy that suggests itself is to modify the substitution matrix but this is not a good idea since this is a global approach and does not lend itself to the local shifting of columns as suggested by coupled pattern sets. We instead adopt a constraint-based alignment strategy, based on COBALT [81], which can flexibly incorporate domain knowledge. Constraints in COBALT are specified in terms of two segments from a pair of sequences that should be aligned with each other in the final result. To convert coupled patterns into constraints, we can adopt various strategies. One approach is to, for each pair of sequences, identify a pair of column positions that should be realigned based on the coupled pattern set. We then map these two positions in the
alignment to the corresponding positions in the original (ungapped) sequences. (These two positions in terms of the original sequences thus constitute a segment pair of size one that should be realigned.) Taking all pairs of sequences in this manner would generate a huge number of constraints. We can reduce the number of constraints by considering consecutive pair of sequences. Another approach is to take a subset of sequences, say $S_1$, for whom the residues match over a column in the coupled pattern. We then take each of the sequences for whom residues do not match over that column in the coupled pattern, and create constraints by pairing the sequence with each of the sequences from $S_1$. COBALT guarantees a maximal consistent subset of these constraints to be occurred in the final alignment. The runtime for an alignment using COBALT is data-centric [81]. DIALIGN [88] is another possible algorithm that can be used to realign sequences. It takes user-defined anchor points but might yield non-aligned residues in the alignment. Due to our desire for global alignments we focus on the COBALT strategy but ARMiCoRe can be easily incorporated into DIALIGN as well.

### 4.5 Experimental Results

In this section, we assess ARMiCoRe on benchmark datasets. Due to space limitations, we provide only representative results illustrating selective superiorities of ARMiCoRe. Our goals are to answer the following questions:

1. How is the discovery of coupled patterns influenced by the dominant residue conserva-
tion threshold ($\tau_d$), block coverage threshold $\tau$, and column window parameter $\epsilon$? (see Section 4.5.3)

2. How does ARMiCoRe fare against classical algorithms on benchmark datasets? Here we choose ClustalW and COBALT, two representative MSA algorithms. (see Section 4.5.4)

3. Can ARMiCoRe extract coupled patterns that capture evolutionary covariation in protein families? (see Section 4.5.5, Section 4.5.6, and Section 4.5.7)

4. Can domain expertise be used to drive the computation of improved alignments? (see Section 4.5.8)

### 4.5.1 Datasets

We use both simulated and benchmark datasets to evaluate our method.

**Simulated Datasets**

To evaluate our proposed method, we designed a simulation model to generate MSAs with embedded coupled patterns. We generated 27 synthetic protein families varying various parameters (see Table 4.1). Subsequently, the multiple sequence alignments were stripped of the gap (‘-’) symbols to obtain contiguous residue sequences. We used a standard multiple sequence alignment algorithm (in this case ClustalW) to align these sequences and used this new alignment to mine for coupled patterns.
Table 4.1: Description of simulated datasets. Each of the dataset from A0 to F2 has 100 sequences and 100 residues.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Parameter Value</th>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0–A2</td>
<td>{0.2, 0.4, 0.6}</td>
<td>Fraction of columns involved in couplings</td>
</tr>
<tr>
<td>B0–B3</td>
<td>{2, 3, 4, 5}</td>
<td>Number of columns in each embedded coupled pattern</td>
</tr>
<tr>
<td>C0–C3</td>
<td>{2, 3, 4, 5}</td>
<td>Number of partitions or blocks in each embedded coupled pattern</td>
</tr>
<tr>
<td>D0–D2</td>
<td>{0.2, 0.4, 0.6}</td>
<td>Fraction of sequences covered by the dominant or combination in each coupled pattern</td>
</tr>
<tr>
<td>E0–E2</td>
<td>{0.4, 0.6, 0.8}</td>
<td>Fraction of sequences covered by the conserved symbol in a given conserved column</td>
</tr>
<tr>
<td>F0–F2</td>
<td>{0.05, 0.1, 0.2}</td>
<td>Fraction of deletions (i.e. blanks '-') in a column</td>
</tr>
<tr>
<td>G0–G2</td>
<td>{50, 100, 150}</td>
<td>Number of columns in a simulated alignment</td>
</tr>
<tr>
<td>H0–H2</td>
<td>{50, 100, 150}</td>
<td>Number of sequences in a simulated alignment</td>
</tr>
</tbody>
</table>

The simulator generates an MSA by first randomly labeling residue positions as either a conserved column, randomly distributed column, or part of a coupled pattern. Each conserved column is then assigned a dominant symbols randomly drawn from 20 amino acid residues and each row of the MSA for that residue position gets the dominant symbol with high probability or one of the remaining amino acid symbols (including a gap) with remaining probability. A residue position labeled as random receives amino acid symbols with equal probability. Next, couplings are embedded over the set of columns allocated for this purpose. Each coupled pattern embedded into the MSA consists of two or more sets of symbols where all sets have the same number of distinct residue symbols. Each set of symbols in a coupled pattern is randomly assigned a sequence in the MSA and the symbols of the set are placed in the respective columns of the MSA assigned to that coupling. The number of columns in a coupled pattern, the number of sets or partitions and the total number of coupled patterns
to embed are input by the user. There is also a provision in the simulator to set probabilities of assignment to each of the symbol sets or partitions in a coupling. For example in our simulation, we designate one of the residue sets as the dominant combination which is used in a larger fraction of sequences in the MSA.

**Benchmark Datasets**

We evaluate our method using three well-known benchmark datasets: BaliBase3 [7], OXBench [89], and SABRE [90]. The BaliBase3 benchmark is created for evaluating both pairwise and MSA algorithms. We use only those alignments from BaliBase that have at least 25 sequences, which yields 48 alignments from three reference sets: RV12, RV20, and RV30. (We chose a threshold of 25 sequences in order to maintain the fidelity of couplings within a sequence family.) For reference set RV20 and RV30 we chose additional threshold of 400 residues for sequence length to reduce the number of alignments in the datasets. The alignments in the reference set RV12 are composed of sequences that are equidistant and have 20-40% identity. The reference set RV20 contains alignments that are composed of highly divergent orphan sequences. The reference set RV30 contains alignments that are composed of sequence groups each of whom have less than 25% identity. OXBench has 3 reference sets and the master set contains 673 alignments that have sequences ranges from 2 to 122. From the master set, we chose a subset that have at least 25 sequences (yields 20 alignments). SABRE contains 423 alignments that have sequences ranges from 3 to 25. We choose a subset of 6 sequences that have at least 20 sequences.
Other than these benchmark datasets, we use families of proteins couplings: GPCR, WW, and PDZ. G-protein coupled receptors are a key demonstrator of allosteric communication and serve to transduce extracellular stimuli into intracellular signals [39]. The entire GPCR family is subdivided into 16 subfamilies (alignments). We use 6 alignments from this set, each of whom involve at least 30 sequences: Amine, Rhodopsin, Peptide, Olfactory, Nucleotide, and Prostanoid. The PDZ family has only one alignment and the WW family has three subfamilies: native, CC, and IC.

A summary of the alignments that are used in the experiments are shown in Table 4.2.
4.5.2 Scoring Criteria

We use four different scoring criteria to assess the quality of a test alignment with respect to a reference alignment. The scores are as follows:

1. Q-Score \[61\]: This score, a.k.a. sum-of-pairs score, can be defined as follows. Let \( T \) be the number of aligned residue pairs in the reference alignment and \( L \) be the number of aligned residues pairs in the reference alignment that are also correctly aligned in the test alignment. Then, Q-score = \( \frac{L}{T} \).

2. Total Column Score (TC) \[7\]: This score is measured by the percent of the number of columns in the reference alignment that are identical with a test alignment. Let \( m \) be the number of columns in a reference alignment and \( m' \) be the number of columns that are identical in both of the reference and test alignments. Then, TC-Score = \( \frac{m'}{m} \).

3. Modeler Score \[91\]: This score is the same as the Q-score but with a different denominator. The score is the percent of pairs of residues in the test alignment that are present in the reference alignment. Let \( R \) be the number of aligned residue pairs in a test alignment and \( L \) be the number of aligned residue pairs in the reference alignment that are also correctly aligned in the test alignment. Then, Modeler score = \( \frac{L}{R} \).

4. Cline Shift Score \[92\]: While the above three scores evaluate only correctly aligned residues or residue pairs, the Shift score also penalizes misalignments. See \[92\] for more details.
5. Coupled Column Score (C-Score): None of the above four scores measure how many of the coupled columns (columns that are participating in the couplings) of a reference alignment are retained in the test alignment. We propose a new score to measure the fraction of retained coupled columns based on probabilistic graphical models (PGMs). PGMs can encode couplings of an alignment where each node denotes a column of the alignment and each edge denotes a coupling between two columns. To calculate the C-Score, we create a PGM for a reference alignment, and then count the number of columns \( V \) that are participating in couplings. For these \( V \) coupled columns in the reference alignment, we count how many \( V' \) of them are retained in the test alignment. A column in the reference alignment is considered to be retained in the test alignment if the number of mismatched residues are fewer than 10% of the residues in the particular column in the reference alignment. These two counts give us C-Score = \( \frac{V'}{V} \).

For all the above measures, higher values are better. The five measures yield a maximum score of 1. The first four measures yield a score of 1 when both the reference and test alignments are identical. The first three measures yield a score of 0 when the alignments are a complete mismatch. For the Shift score, the minimum possible score is \(-0.2\) by default.

### 4.5.3 Effects of Important Thresholds

The parameters that have the most significant impact on the number of coupled patterns discovered are the dominant residue conservation threshold \( \tau_d \), block coverage threshold \( \tau \),
and column-window size threshold ($\epsilon$). Based on the 27 synthetic alignments, we produced precision-recall curves using various values for $\tau_d$, $\tau$ and $\epsilon$. In synthetic alignments couplings are embedded. We run our methods on these datasets to discover coupled patterns based on various parameters and see how many of the discovered coupled patterns are matched (true positive) and how many are redundant (false positive). The precision-recall curve for $\tau_d \in \{0.2, 0.4, 0.6, 0.8\}$ which are illustrated in Fig. 4.6. We vary the block pattern threshold parameter $\tau$ in the set $\{0.4, 0.6, 0.8, 0.10, 0.12\}$ and generate precision-recall plots (see Fig. 4.7). Similarly, we produce precision-recall curves for $\epsilon \in \{1, 2, 3, 4, 5\}$ (see Fig. 4.8). As all the plots reveal, our method maintain consistently high levels of recall and precision across a wide range of thresholds.
Varying Block conservation Threshold from 0.04 to 0.12

Figure 4.7: Precision-recall plots for the coverage threshold \( \tau \) \([0.4,0.6,0.8,0.10,0.12]\).

Varying Column Span from 1 to 5

Figure 4.8: Precision-recall plots for the window size parameter \( \epsilon \) [1 to 5].
Table 4.3: Comparison of ARMiCoRe with ClustalW and Cobalt on synthetic dataset.

<table>
<thead>
<tr>
<th>Score</th>
<th>ClustalW</th>
<th>Cobalt</th>
<th>ARMiCoRe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-score</td>
<td>0.516</td>
<td>0.265</td>
<td>0.551</td>
</tr>
<tr>
<td>TC Score</td>
<td>0.0</td>
<td>0.009</td>
<td>0.017</td>
</tr>
<tr>
<td>Shift score</td>
<td>0.655</td>
<td>0.355</td>
<td>0.663</td>
</tr>
<tr>
<td>Modeler Score</td>
<td>0.512</td>
<td>0.474</td>
<td>0.592</td>
</tr>
</tbody>
</table>

Table 4.4: Comparison of ARMiCoRe with Cobalt on RV12 reference set of the BaliBase [7] benchmark.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Q-Score</th>
<th>TC Score</th>
<th>Shift Score</th>
<th>Modeler Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cobalt</td>
<td>ARMiCoRe</td>
<td>Cobalt</td>
<td>ARMiCoRe</td>
<td>Cobalt</td>
</tr>
<tr>
<td>BB12035</td>
<td>0.75</td>
<td>0.74</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td>BB12043</td>
<td>0.68</td>
<td>0.66</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>BBS12035</td>
<td>0.78</td>
<td>0.81</td>
<td>0.30</td>
<td>0.38</td>
</tr>
<tr>
<td>BBS12043</td>
<td>0.75</td>
<td>0.80</td>
<td>0.24</td>
<td>0.33</td>
</tr>
<tr>
<td>Avg</td>
<td>0.74</td>
<td>0.75</td>
<td>0.21</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 4.5: Comparison of ARMiCoRe against ClustalW over all BaliBase datasets (using only core regions). The average scores are shown here. RV20* is curated from RV20 by removing orphan sequences.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Q-Score</th>
<th>TC Score</th>
<th>Shift Score</th>
<th>Modeler Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClustalW</td>
<td>ARMiCoRe</td>
<td>ClustalW</td>
<td>ARMiCoRe</td>
<td>ClustalW</td>
</tr>
<tr>
<td>RV12</td>
<td>0.84</td>
<td>0.89</td>
<td>0.51</td>
<td>0.61</td>
</tr>
<tr>
<td>RV20</td>
<td>0.84</td>
<td>0.79</td>
<td>0.24</td>
<td>0.15</td>
</tr>
<tr>
<td>RV20*</td>
<td>0.88</td>
<td>0.90</td>
<td>0.54</td>
<td>0.57</td>
</tr>
<tr>
<td>RV30</td>
<td>0.68</td>
<td>0.58</td>
<td>0.23</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Table 4.6: Comparison of ARMiCoRe against ClustalW over the OXBench alignments.

<table>
<thead>
<tr>
<th>Alignments</th>
<th>Q-Score</th>
<th>TC Score</th>
<th>Shift Score</th>
<th>Modeler Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClustalW</td>
<td>ARMiCoRe</td>
<td>ClustalW</td>
<td>ARMiCoRe</td>
<td>ClustalW</td>
</tr>
<tr>
<td>12s107</td>
<td>0.99</td>
<td>0.98</td>
<td>0.80</td>
<td>0.86</td>
</tr>
<tr>
<td>12s108</td>
<td>0.97</td>
<td>0.99</td>
<td>0.85</td>
<td>0.94</td>
</tr>
<tr>
<td>12t109</td>
<td>0.96</td>
<td>0.96</td>
<td>0.76</td>
<td>0.80</td>
</tr>
<tr>
<td>12t113</td>
<td>0.95</td>
<td>0.91</td>
<td>0.82</td>
<td>0.56</td>
</tr>
<tr>
<td>12t116</td>
<td>0.94</td>
<td>0.87</td>
<td>0.53</td>
<td>0.33</td>
</tr>
</tbody>
</table>
| ...        | ...     | ...     | ...     | ...     | ...     | ...     | ...     | ...
| 588t28     | 1.00    | 0.99    | 0.97    | 0.97    | 0.89    | 0.89    | 0.80    | 0.81    |
| _22s38     | 0.95    | 0.95    | 0.82    | 0.81    | 0.81    | 0.81    | 0.69    | 0.69    |
| _22t50     | 0.96    | 0.95    | 0.86    | 0.83    | 0.79    | 0.78    | 0.64    | 0.64    |
| _588       | 0.98    | 0.98    | 0.83    | 0.8     | 0.88    | 0.89    | 0.80    | 0.81    |
| _12        | 0.86    | 0.87    | 0.00    | 0.10    | 0.5     | 0.53    | 0.34    | 0.37    |

4.5.4 Comparison with ClustalW

We evaluate ARMiCoRe on all the datasets described earlier: synthetic, benchmark and alignments with couplings. For each of these alignments, we remove gaps and realign with
Table 4.7: Comparison of ARMiCoRe against ClustalW over the SABRE alignments.

<table>
<thead>
<tr>
<th>Alignments</th>
<th>Q-Score</th>
<th>TC Score</th>
<th>Shift Score</th>
<th>Modeler Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ClustalW</td>
<td>ARMiCoRe</td>
<td>ClustalW</td>
<td>ARMiCoRe</td>
</tr>
<tr>
<td>sup_038</td>
<td>0.82</td>
<td>0.88</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sup_092</td>
<td>0.20</td>
<td>0.30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sup_108</td>
<td>0.89</td>
<td>0.93</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sup_126</td>
<td>0.51</td>
<td>0.59</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sup_167</td>
<td>0.61</td>
<td>0.50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sup_215</td>
<td>0.11</td>
<td>0.18</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

ClustalW in default settings (using the PAM matrix). We then run ARMiCoRe on each of the ClustalW alignments to generate coupled patterns and use the coupled patterns to generate constraints, which are used by COBALT to create an improved alignment. We then compare our scores with ClustalW and with COBALT (without any constraints input).

As shown in Table 4.3, ARMiCoRe excels in all four traditional measures of MSA quality for synthetic datasets. Performance of ARMiCoRe on all reference sets of the BaliBase benchmark is given in Table 4.5. ARMiCoRe shows superior performance over ClustalW on all of the four measures in the RV12 reference set. The sequence identity in this benchmark is about 20–40%. Note that the performance of ARMiCoRe on RV20 and RV30 is worse than that of ClustalW in all four measures. This is because RV20 and RV30 pool together sequences with poor similarity and thus coupled patterns are not a driver for obtaining good alignments. The effect of an orphan sequence on the similarity structure of an alignment is illustrated in Fig. 4.9. To test this hypothesis, we removed the orphan sequences from RV20 (RV20*) and as Table 4.5 shows, the performance of ARMiCoRe is better along three of the four measures. Table 4.6 describes the results of ARMiCoRe for the OXBBench benchmark, once again revealing a mixed performance on a dataset with high sequence diversity. Finally, Table 4.7 depicts the superior performance of ARMiCoRe over ClustalW in 5 alignments out
Figure 4.9: Pairwise sequence similarity analysis of an alignment ‘BB20006’ from RV20 dataset that contains an orphan sequence. We use SCA [5] for this analysis. Fig. 4.9a has a peak for similarity score around 0.12 that indicates that the orphan is distant from the other sequences. Fig. 4.9b shows a reasonably narrow distribution without the orphan sequence with a mean pairwise similarity between sequences of about 27% and a range of 20% to 35%, which suggests that most sequences are about equally dissimilar from other.

Table 4.8: Comparison of ARMiCoRe against ClustalW and COBALT over the CC subfamily of WW protein family.

<table>
<thead>
<tr>
<th>Score</th>
<th>ClustalW</th>
<th>COBALT</th>
<th>ARMiCoRe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-Score</td>
<td>0.89</td>
<td>0.85</td>
<td>0.96</td>
</tr>
<tr>
<td>TC Score</td>
<td>0.51</td>
<td>0.35</td>
<td>0.51</td>
</tr>
<tr>
<td>Shift Score</td>
<td>0.93</td>
<td>0.91</td>
<td>0.97</td>
</tr>
<tr>
<td>Modeler Score</td>
<td>0.90</td>
<td>0.91</td>
<td>0.97</td>
</tr>
<tr>
<td>C-Score</td>
<td>0.71</td>
<td>0.88</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 4.9: Comparison of ARMiCoRe against ClustalW and COBALT over the PDZ family.

<table>
<thead>
<tr>
<th>Score</th>
<th>ClustalW</th>
<th>COBALT</th>
<th>ARMiCoRe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-score</td>
<td>0.85</td>
<td>0.82</td>
<td>0.87</td>
</tr>
<tr>
<td>TC Score</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>Shift score</td>
<td>0.89</td>
<td>0.87</td>
<td>0.9</td>
</tr>
<tr>
<td>Modeler Score</td>
<td>0.85</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>C-Score</td>
<td>0.67</td>
<td>0.81</td>
<td>0.81</td>
</tr>
</tbody>
</table>

of 6 alignments in SABRE dataset.
Table 4.10: Comparison of ARMiCoRe against ClustalW and COBALT over the Nucleotide subfamily of GPCR protein family.

<table>
<thead>
<tr>
<th>Score</th>
<th>ClustalW</th>
<th>COBALT</th>
<th>ARMiCoRe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-Score</td>
<td>0.74</td>
<td>0.68</td>
<td>0.79</td>
</tr>
<tr>
<td>TC Score</td>
<td>0.46</td>
<td>0.34</td>
<td>0.45</td>
</tr>
<tr>
<td>Shift Score</td>
<td>0.79</td>
<td>0.74</td>
<td>0.83</td>
</tr>
<tr>
<td>Modeler Score</td>
<td>0.74</td>
<td>0.77</td>
<td>0.80</td>
</tr>
<tr>
<td>C-Score</td>
<td>0.52</td>
<td>0.50</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Figure 4.10: An overview of user interaction with ARMiCoRe.

4.5.5 Modeling Correlated Mutations

We describe the effect of ARMiCoRe on three families that are known to exhibit correlated mutations. We focus on the CC subfamily of the WW domain, the PDZ family, and the Nucleotide subfamily of the GPCR family. Based on C-Score, we evaluate the Performance of ARMiCoRe against ClustalW and COBALT. As shown in Tables 4.8, 4.9, and 4.10, ARMiCoRe is consistently better on at least three measures.
4.5.6 Evaluation using Global Statistical Model for Residue Couplings

Couplings are often employed to predict 3D structure of proteins from sequences. A global statistical method for residue couplings for predicting 3D structure of proteins is proposed by Marks et al. [8]. The proposed method first calculates pairwise coupling scores and then uses high scoring pairs to find a 3D structure. The way of calculating couplings scores is global which is different from the method given by Thomas et al. [28]. We use their method to calculate pairwise couplings scores for reference, ClustalW, and ARMiCoRe alignments. We then identify how many of the coupled pairs (true positive) for the reference alignment are also retained in ClustalW and ARMiCoRe alignments for various thresholds. In Table 4.11 and Table 4.13 we see that the alignments for CC and Nucleotide subfamily given by ARMiCoRe is much better than that of ClustalW. But the alignment for PDZ family...
Table 4.11: Comparison of ARMiCoRe against ClustalW over the CC subfamily of WW family using the global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.

<table>
<thead>
<tr>
<th>Score Threshold</th>
<th>Number of couplings in Ref MSA</th>
<th>Number of couplings in ClustalW MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
<th>Num of couplings in ARMiCoRe MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.40</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0.20</td>
<td>0.13</td>
<td>0.15</td>
<td>5</td>
<td>3</td>
<td>0.60</td>
<td>0.38</td>
<td>0.46</td>
</tr>
<tr>
<td>0.35</td>
<td>27</td>
<td>19</td>
<td>13</td>
<td>0.68</td>
<td>0.48</td>
<td>0.57</td>
<td>29</td>
<td>21</td>
<td>0.72</td>
<td>0.78</td>
<td>0.75</td>
</tr>
<tr>
<td>0.30</td>
<td>64</td>
<td>51</td>
<td>36</td>
<td>0.74</td>
<td>0.56</td>
<td>0.63</td>
<td>61</td>
<td>53</td>
<td>0.87</td>
<td>0.83</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Table 4.12: Comparison of ARMiCoRe against ClustalW over the PDZ family using the global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.

<table>
<thead>
<tr>
<th>Score Threshold</th>
<th>Number of couplings in Ref MSA</th>
<th>Number of couplings in ClustalW MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
<th>Num of couplings in ARMiCoRe MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.20</td>
<td>15</td>
<td>13</td>
<td>4</td>
<td>0.27</td>
<td>0.31</td>
<td>0.29</td>
<td>17</td>
<td>3</td>
<td>0.20</td>
<td>0.18</td>
<td>0.19</td>
</tr>
<tr>
<td>0.15</td>
<td>64</td>
<td>71</td>
<td>21</td>
<td>0.33</td>
<td>0.30</td>
<td>0.31</td>
<td>82</td>
<td>13</td>
<td>0.20</td>
<td>0.16</td>
<td>0.18</td>
</tr>
<tr>
<td>0.10</td>
<td>396</td>
<td>405</td>
<td>190</td>
<td>0.48</td>
<td>0.47</td>
<td><strong>0.47</strong></td>
<td>474</td>
<td>145</td>
<td>0.37</td>
<td>0.31</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table 4.13: Comparison of ARMiCoRe against ClustalW over the Nucleotide subfamily of GPCR family using the global residue coupling model defined in [8]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.

<table>
<thead>
<tr>
<th>Score Threshold</th>
<th>Number of couplings in Ref MSA</th>
<th>Number of couplings in ClustalW MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
<th>Num of couplings in ARMiCoRe MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>-</td>
<td>9</td>
<td>1</td>
<td>0.11</td>
<td>0.14</td>
<td><strong>0.13</strong></td>
</tr>
<tr>
<td>0.12</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>24</td>
<td>3</td>
<td>0.13</td>
<td>0.21</td>
<td><strong>0.16</strong></td>
</tr>
<tr>
<td>0.10</td>
<td>21</td>
<td>23</td>
<td>2</td>
<td>0.09</td>
<td>0.10</td>
<td>0.09</td>
<td>37</td>
<td>5</td>
<td>0.14</td>
<td>0.24</td>
<td><strong>0.17</strong></td>
</tr>
</tbody>
</table>

Table 4.14: Comparison of ARMiCoRe against ClustalW over the CC subfamily of WW family using the statistical coupling analysis defined in [9]. Here ‘TP’ is used for true positive, ‘P’ is used for precision, and ‘R’ is used for recall.

<table>
<thead>
<tr>
<th>Cut-off Threshold</th>
<th>Sector Size in Ref MSA</th>
<th>Sector Size in ClustalW MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
<th>Sector Size in ARMiCoRe MSA</th>
<th>TP</th>
<th>P</th>
<th>R</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>1.00</td>
<td>0.71</td>
<td>0.83</td>
<td>7</td>
<td>6</td>
<td>0.86</td>
<td>0.86</td>
<td><strong>0.86</strong></td>
</tr>
<tr>
<td>0.80</td>
<td>11</td>
<td>7</td>
<td>7</td>
<td>1.00</td>
<td>0.64</td>
<td>0.78</td>
<td>9</td>
<td>9</td>
<td>1.00</td>
<td>0.82</td>
<td><strong>0.90</strong></td>
</tr>
<tr>
<td>0.75</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>1.00</td>
<td>0.67</td>
<td>0.80</td>
<td>12</td>
<td>11</td>
<td>0.92</td>
<td>0.92</td>
<td><strong>0.92</strong></td>
</tr>
</tbody>
</table>

given by ARMiCoRe is not better than that of ClustalW in terms of couplings calculated in global settings (see Table 4.12).
Figure 4.12: Pairwise sequence similarity analysis using SCA [5]. Histograms for reference, ClustalW, and ARMiCoRe are drawn for the same number of bins. This figure shows that the ARMiCoRe alignment retains most of the sequence similarity structure of the reference alignment.

4.5.7 Evaluation using Statistical Coupling Analysis

Lockless and Ranganathan [9] proposed statistical coupling analysis (SCA) as a method for analyzing coevolution in protein families represented by MSAs. The SCA tool [5] performs sequence similarity analysis to get an idea of the number of subfamilies in the MSA. We perform similarity analysis for the reference, ClustalW, and ARMiCoRe alignments of CC family (see Fig. 4.12). There are two subfamilies for the CC family: folded and non-folded. Fig. 4.12 shows that for both reference and ARMiCoRe alignments there are two peaks in the histogram which is an indication that there are two subfamilies, whereas the ClustalW alignment indicates that the similarity structure for two subfamilies are distorted.

SCA also allows us to identify protein sectors, which are quasi-independent groups of correlated amino acids [93]. We identify protein sectors of reference, ClustalW, and ARMiCoRe alignments for various cut-off thresholds (0.85, 0.80, and 0.75). We then calculate precision, recall, and F1-score for ClustalW and ARMiCoRe alignments with respect to the reference
alignment. For all of the cut-off thresholds one protein sector is identified. Table 4.14 shows that much of protein sector in the reference alignment is retained in the ARMiCoRe alignment.

### 4.5.8 User Interaction in Choosing Couplings

We have developed GUIs for ARMiCoRe that allow users to interactively choose patterns from a set of significant coupled patterns and use them to realign sequences. This enables biologists to bring specific domain knowledge in deciding which coupled pattern sets should be exposed as couplings in the new alignment. We have integrated ARMiCoRe with the JalView [94] framework, which has a rich set of sequence analysis tools. A typical workflow with ARMiCoRe is illustrated in Fig. 4.10. A user begins an experiment by loading an initial alignment (see Fig. 4.11(a)). He or she can evaluate the input alignment by measuring various scores with respect to a reference alignment. Based on the evaluation, he or she may decide to improve the alignment using the coupled pattern mining module. The coupled pattern mining module facilitates tuning various parameters prior to the pattern mining and gives a set of significant coupled patterns as output. From the pool of coupled patterns, a domain expert can choose meaningful patterns (see Fig. 4.11(b)) and use them in the realignment module. The realignment module gives a new alignment, which can be evaluated in the evaluation module. A user may repeat the realignment step by choosing different patterns or the mining step by tuning the parameters.
4.6 Discussion

Evolutionary constraints on genes and proteins to maintain structure and function are revealed as conservation and coupling in an MSA. The advent of cheap, high-throughput sequencing promises to provide a wealth of sequence data enabling such applications, but at the same time requires methods such as ARMiCoRe to improve the alignments and inferred constraints upon which they are based. The alignments obtained by ARMiCoRe can be leveraged to design or classify novel proteins that are stably folded and functional [95, 96, 9, 31], as well as to predict three-dimensional structures from sequence alone [44, 8, 97]. Our work also demonstrates a successful application of pattern set mining where the goal is not just to find patterns but to cover the set of sequences with discovered patterns such that an objective measure is optimized. The ideas developed here can be generalized to other pattern set mining problems in areas like neuroscience, sustainability, and systems biology.
Chapter 5

Conclusion

The goal of this dissertation is to develop data mining techniques for modeling correlated mutations or couplings in proteins. We have developed methods, learning structures with graphical models and mining frequent episodes, that are applicable to problems concerning couplings. We believe that the developed methods would provide new insight (structural and functional) into protein and could be extended to infer coevolving structures in other domains.

In this dissertation, we deal with three bioinformatics problems and connect them with the common theme of couplings or correlated mutations. The developed framework brings with it a collection of algorithms addressing following challenges on evolutionary constraints analysis:

1. *Can we model and infer pairwise couplings that explicate the underlying coevolving*
To address this challenge, we define a novel type of coupling based on amino acid classes such as polarity, hydrophobicity, and size, and present two approaches for learning probabilistic graphical models to represent such couplings. These models can take optional structural priors into account for building graphical models. Couplings represented with graphical models can be used in many applications such as predicting protein structures, creating synthetic protein, and classification of new proteins. Our proposed models discover couplings that are richer and have mechanistic explanations, which are absent in standard methods.

2. How to model and infer higher-order couplings between residues? Existing research on coupling primarily focuses on identifying pairwise coupling. As more than two residues can interact with each other in a 3-D structure of a protein, it is interesting to examine whether a generalization of pairwise coupling is possible. This type of higher-order could offer us deeper insight into structures and functions of proteins. In this study, we define higher-order coupling in proteins, and identify and express such couplings with two probabilistic graphical models: Bayesian network and factor graph model. We evaluate our methods with nickel-repressor and GPCR protein families. We observe that both models capture higher-order couplings between residues that are critical to the functional activities of this family.

3. Can the quality of multiple protein alignment be improved by exposing embedded couplings in the sequences? This question addresses an inherent problem in classical multiple sequence alignment algorithms, which overlook coevolution between residues.
To alleviate this problem, we develop a two phase algorithm: using frequent episode mining we infer coupled patterns in a traditional alignment and exploit the coupled patterns to realign the sequences that are better than the traditional alignments. This algorithm allows optional user interactions in the realignment phase to bring specific domain knowledge. This research is one of the early steps towards auto-correction of an alignment measured in terms of exposition of couplings. The proposed method can be viewed as a novel application of the pattern set discovery where the goal is not just to mine interesting patterns (which is the purview of pattern discovery) but to select among them to optimize a set-based measure.

This dissertation opens up many opportunities for future exploration from theoretical and application perspectives. The problem of modeling couplings (Ch. 2–3) in proteins can be seen as a specific instance of modeling coevolving entities in many-body systems, which are prevalent in biology, physics, sociology, and computer networks. The area of learning granular structures for coevolving entities has great research potential. We can explore more formal classes of algorithms that would learn coevolving entities with their fine-grained interactions.

RNA exhibits couplings between sites in spatial conformation, which largely determines the functions. The secondary and tertiary structures of RNA form self-complementary base pairs, which yield different structural motifs such as stem and loop. Stem regions of RNA contain covarying residues and show greater sequence diversity compared to other regions, whereas loops in RNA exhibit conservations. The presence of covarying residues in RNA poses a
challenge to align RNA sequences correctly using only sequence data \[98, 99\]. We can extend our ARMiCoRe (Ch. 4) to mine coupled patterns of bases in multiple RNA alignments and exploit the patterns for guiding alignment algorithms in improving the quality of alignments (possibly with a realignment step). In future, we intend to evaluate our method using a large collection of benchmark datasets.

Analyzing opinion dynamics in social networks as well as news outlets is a fledgling research topic and can help answering questions in social dynamics. A natural extension of the proposed coupling algorithms (Ch. 2–3) is to adapt the model to infer dynamic coupled relationships between entities and actors in both spheres—news and social media. Particularly, we can investigate how polarization occurs in discussion threads, how entities influence each other, and what is the underlying structure of mutual influence. We aim to adapt our models and develop predictive algorithms to capture and characterize such occurrences and relationships between actors in the real world using surrogate data generated in social network sites.
Bibliography


[24] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-


