Automated Motif Discovery
in Protein Structure Prediction

by

Evan W. Steeg

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

© Evan W. Steeg 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.
Automated Motif Discovery in Protein Structure Prediction

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy, 1997

Evan W. Steeg

Graduate Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4

Abstract

The protein structure prediction problem (PSP) is one of the central problems in molecular and structural biology. A computational method that could produce a correct detailed three-dimensional structural model for a protein, given its linear sequence of amino acids, would greatly accelerate progress in the biomedical sciences and industries. This thesis presents PSP as a combinatorial optimization problem, the most straightforward formulations of which require search of an exponentially-large conformation space and are known to be NP-Hard. This otherwise intractable search can in practice be reduced or eliminated through the discovery and use of motifs. Motifs are abstractions of observed patterns that encode structurally important relationships among constituent parts of a complex object like a protein tertiary structure. Motif discovery is accomplished by particular combinatorial search and statistical estimation methods.

This thesis explores in detail two particular motif discovery subproblems, and discusses how their solutions can be applied to the overall structure prediction problem:

1. For a complex multi-stage prediction task, what makes a good intermediate representation language? We address this question by presenting and analyzing methods for the discovery of protein secondary structure classes that are more predictable from amino acid sequence than the standard classes of α-helix, β-sheet, and “random coil”.

2. Given a database of M objects, each characterized by values \( a_{ij} \in A_j \) for each of \( N \) discrete variables \( \{c_j\}_{j=1}^N \), return the list of “most interesting” higher-order features \( \gamma_i \), i.e., sets of \( k_l \) variables with highest estimated correlation, for any \( 2 \leq k_l \leq N \). In the PSP context, the problem is the detection of correlations between amino acid residues in an aligned set of evolutionarily-related protein sequences. We present and analyze a fast procedure, based on multinomial sampling and a novel coding scheme, that avoids the exhaustive search, prior limits on the order \( k_l \), and exponentially large parameter space of other methods.

The focus of this thesis is PSP, but the techniques and analysis are also aimed at wider application to other hard, multi-stage prediction problems.
I wish to thank my committee members. Professors Geoffrey Hinton, Tony Bonner, Allan Jepson, Ken Sevcik and David Tinker of the University of Toronto, for their careful reading of this long, interdisciplinary thesis and for their helpful comments, corrections and suggestions. In addition, I thank Geoff for his patience and for creating and maintaining an intellectually exciting and exacting research environment within the Neural Networks Research Group.

Sincere thanks also go to my external examiner, Dr. Terry Sejnowski of the Salk Institute, for taking time out of his busy schedule to participate in the Ph.D. examination process.

I gratefully acknowledge the collaboration of Drs. Alan Lapedes and Rob Farber, of Los Alamos National Labs and the Santa Fe Institute, on some of the work discussed in Chapter Three. I thank Alan, Geoff, and Melanie Mitchell for setting up the collaboration and the Institute for putting me up in beautiful Santa Fe for a couple of research visits. I also want to thank Geoff for key suggestions on the use of Method 3 in Chapter 3, and Drew van Camp for implementation help with that method.

I am indebted to my good friend Derek Robinson for inspiration, for early work on the Coincidence Detection methods of association mining, for reminding me that there are no intellectual boundaries and that intellectual excitement is possible even without the “carrot” of publications, patents, or profits. I thank Timothy Horton for reminding me who I am from time to time as needed, and Tony Chiverton for helping me to remember that there is much more to life than work, even for people who love their work. Dr. Larry Hunter has been a source of encouragement (“Finishing your Ph.D. is the greatest lifestyle change you can ever make!”) and insights into the computational molecular biology research community for many years.

I want to express my appreciation to Professors Janice Glasgow and Suzanne Fortier for providing a stimulating “postdoctoral” position within the Molecular Scene Analysis group, Departments of Computing Science and Chemistry at Queen’s University. I have enjoyed working with them and with group members Kim, Alan, Shishan, Tony, Peter, Ken, Ed, Hai, Laurence, Jean, David and Jennifer.

I am grateful to John Molloy for presenting an exciting opportunity for life after academia.

Many friends have made this long process survivable and sometimes even fun - lots of fun! Among them are Tim, Derek, Gara, Sherry, Demetri, Tony, Hai, Matt, Jean, Misun, Ed, Karen, Bob, Kim, Rich, Toni, Tom, Diane, Larry, Sarah, Craig, Mike, Brenda, Maggie, John, Raj, Sid, Kirsty, Teena and Sageev.
My mother and father, and Barbara, Yusuf, Nadia, the Miernickis and Laughtons have always been there for me, with a word of encouragement, a hot meal, a loan, or a laugh as needed. Finally, I thank Carol for love and support and strength beyond what anyone could ever reasonably expect.
Contents

List of Tables viii

List of Figures x

1 Introduction 1

1.1 The Topic and Scope of the Thesis 1

1.2 Overview of the Thesis 3

1.3 Translation from a Biophysical into a Mathematical Model 5

1.4 Organization of this Thesis 7

1.5 Contributions to Computer Science 7

2 Motivation, Background, and Mathematical Framework 9

2.1 Introduction 9

2.2 DNA, RNA, and Proteins 9

2.3 Protein Structure and the Folding Process 13

2.3.1 Levels of Organization in Protein Structure 13

2.3.2 What Drives the Folding Process (and Does It Matter)? 15

2.4 Prediction of Protein Secondary and Tertiary Structure 17

2.4.1 Why Predict? 17

2.4.2 What is Structure Prediction? 18

2.4.3 The Difficulty of the Problem 19

2.4.4 Approaches to PSP 21

2.5 The Sequence Alignment Problem 22

2.6 A Mathematical Framework for PSP 23

2.6.1 First Things First: Primary Structure 23

2.6.2 Protein Structures As Contact Graphs 23

2.6.3 Protein Structures as Self-Avoiding Chains on a Lattice 24

2.6.4 The Problem. Formally: Combinatorial Optimization 25

2.6.5 Intractability 26

2.6.6 Getting Around the Intractability 27

2.7 Empirical Potentials, Probabilities, and Machine Learning 28

2.7.1 What is Machine Learning? 30

2.7.2 Supervised and Unsupervised Learning 31

2.7.3 ML as Motif Discovery 31

2.8 The Need for Locality, Abstraction, and Selectivity 33

2.8.1 Recent Work on ML in PSP 36
2.9 Putting the Pieces Together: New Motifs and Discovery Procedures within an Overall PSP Methodology ................................................. 38

3 Automated Discovery of Predictable Secondary-Structure Classes 46
3.1 Introduction .......................................................... 46
3.2 The Success, and Failure, of Secondary-Structure Prediction .......... 47
  3.2.1 Prediction of Traditional Secondary Structure Classes ............. 48
  3.2.2 Why Can't Secondary Structure Predictors Do Better? ............. 49
  3.2.3 Measuring Predictive Accuracy — Matthews Correlation ....... 51
3.3 Method 1: Correlated Feed-Forward Networks ....................... 52
  3.3.1 Intuition and Background ....................................... 52
  3.3.2 The Objective Function .......................................... 54
  3.3.3 Learning Rules and Prediction Rules ............................ 58
  3.3.4 Experiments and Results ........................................ 62
  3.3.5 Possible Extensions ............................................... 70
3.4 Method 2: Joint Density Estimation with Gaussian Mixture Models .... 71
  3.4.1 Intuition and Background — The Use of MDL Objectives ........... 71
  3.4.2 The Objective Function .......................................... 77
  3.4.3 Learning Rules and Prediction Rules ............................ 78
  3.4.4 Experiments and Results ........................................ 80
  3.4.5 Possible Extensions ............................................... 81
3.5 Method 3: Conditional Mixture Models — Mixture of Competing Experts 83
  3.5.1 Intuition and Background ....................................... 83
  3.5.2 The Objective Function .......................................... 86
  3.5.3 Learning Rules and Prediction Rules ............................ 86
  3.5.4 Experiments and Results ........................................ 88
  3.5.5 Possible Extensions ............................................... 90
3.6 Comparison of the Three Methods .................................. 91
  3.6.1 The Use of Artificial Neural Networks .......................... 94
3.7 Comparison with Other Local Structure Motifs ........................ 94
  3.7.1 Use of Both Sequence and Structure Data ....................... 97
  3.7.2 Number of Classes, Number of Motifs .......................... 98
  3.7.3 Locality: Size of Input Fragments .............................. 99
  3.7.4 Representation, Shape, and Parameterization of Motifs/Classes .... 100
  3.7.5 Intrinsic versus Extrinsic Clustering Criteria ................. 102

4 Modelling Aligned Sequences, and Discovering Higher-Order Features in Discrete Datasets 103
4.1 Motivation: Correlations between Residue Columns .................. 104
  4.1.1 First-Order Features ........................................... 104
  4.1.2 Higher-Order Features ......................................... 106
  4.1.3 The Problem .................................................... 107
4.2 Connection with Other Database Modelling Problems .................. 109
  4.2.1 Three Goals of Probability Estimation ........................ 109
  4.2.2 Estimation/Approximation of the Complete Underlying Probability Distribution ......................................................... 110
    4.2.2.1 Modelling Reduces to Maximum Entropy Assumption and "Suspicious Coincidences" ......................... 111
4.2.3 Hypothesis Testing .................................................. 112
4.2.4 Data Mining .......................................................... 113
4.2.5 Computational Complexity ......................................... 114
4.2.6 Is Protein Sequence Family Analysis a Database Problem?  .................................................................................. 119
4.3 Importance in Protein Structure Prediction and Analysis ............................................................................. 120
4.3.1 Discovery of Evolutionarily-Conserved Structural Constraints ................................................................. 120
4.3.1.1 What Do We Expect to Observe? .................................................................................................................. 120
4.3.1.2 What Has been Observed? ......................................................................................................................... 121
4.3.1.3 How Can These Observations Be Used? ..................................................................................................... 122
4.3.1.4 Correlated Positions and RNA Structure Prediction .................................................................................. 123
4.3.2 Building More Sensitive Protein Classification Systems ........................................................... 123
4.4 Previous and Alternative Methods .............................................................................................................. 124
4.4.1 First-Order Models ....................................................... 124
4.4.2 Direct Calculation of Pairwise Probabilities .............................................................................................. 124
4.4.3 n-gram Methods .......................................................... 125
4.4.4 Hidden Markov Models and Induction of CFG’s and CSG’s ......................................................................... 126
4.4.5 Neural Networks ............................................................ 127
4.4.6 MacKay’s Density Networks ............................................. 128
4.4.7 What Kind of Method is Needed? ........................................ 130
4.5 Coincidence Detection: A Novel Data-Mining Method ...................................................................................... 131
4.5.1 The Basic Logic Behind the Method ............................................................................................................ 131
4.5.2 Definitions .............................................................................................................................. 133
4.5.3 Outline of Procedure ....................................................... 133
4.5.4 Expected and Actual Frequencies of Coincidences .......................................................................................... 134
4.5.4.1 Expected Size h of a Match ......................................................................................................................... 134
4.5.4.2 The Observed Sizes of Matches ............................................................................................................... 136
4.5.5 Hypothesis Tests ............................................................... 136
4.5.6 The Main Procedure ........................................................ 138
4.5.7 Computational Complexity of the Method .................................................................................................. 140
4.5.7.1 Overall Asymptotic Complexity .................................................. 140
4.5.7.2 Estimating E[L], the Expected Size of the Cset Table ............................................................................... 141
4.5.8 Analysis of Error in the Method — Types, Probability, and Bounds ............................................................ 141
4.5.8.1 Error at Level 1 ................................................................................................................................. 143
4.5.8.2 False Negatives and False Positives at Level 3 .......................................................................................... 144
4.5.9 Finding Other Correlated Attributes ....................................................................................................... 146
4.6 Coincidence Sampling: Experiments and Results ............................................................................................. 147
4.6.1 Tests on Specially-Constructed Datasets ...................................................................................................... 147
4.6.1.1 General Form of the Experiments .............................................................................................................. 148
4.6.1.2 Family0 Dataset ...................................................................................................................................... 148
4.6.1.3 Family1 Dataset ..................................................................................................................................... 148
4.6.2 Results of Family0 and Family1 Experiments .............................................................................................. 149
4.6.2.1 Family0 Results ..................................................................................................................................... 149
4.6.2.2 Family1 Results ...................................................................................................................................... 150
4.6.3 Tests of Time and Space Complexity on Independent-Column Datasets ...................................................... 151
4.6.4 Results of Experiments on the Size of L ....................................................................................................... 152
4.6.5 Tests on an HIV Protein Database .............................................................................................................. 154
4.6.6 Results of Experiments on HIV Protein Database ........................................................................................ 155
4.7 Comparison of Coincidence Detection with Other Methods ............................................................................. 156
List of Tables

3.1 Mathews correlation values ($M_k$) between sequence and structure network classifications, on training and prediction sets of examples, from Experiment 1. 65
3.2 Three representations of Actinidin (sulphydryl Proteinase). 66
3.3 Mathews correlation values ($M_k$) between sequence and structure network classifications, on training and prediction sets in Experiment 2. 68
3.4 Mathews correlation values, for trained network on prediction set of examples, for each of X1Bclass, X1Cclass, and X1Dclass as measured against each other and against traditional classes. 68
3.5 Mathews correlation values ($M_k$) between sequence and structure classifications, as measured by responsibility terms $r_{1k}^A$ and $r_{2k}^Z$, respectively, from the sequence and structure sub-networks: on training and prediction sets of examples. 82
3.6 Mathews correlation values ($M_k$) between sequence and structure classifications, as measured by Gating net outputs and Expert net responsibilities, respectively: on training and prediction sets of examples, compared with standard classes and Method 1 classes. 90
3.7 Mathews correlation values, for class X2Aclass produced by $Z$ half of joint density modelling network (Method 2) and for X3Aclass produced by structure portion (experts net) of conditional density modelling network (Method 3), as measured against each other and against traditional classes $\alpha$-helix, $\beta$-sheet, and coil. 92
3.8 Mathews correlation values, for class X2Aclass produced by $A$ half of joint density modelling network (Method 2) as measured against traditional classes $\alpha$-helix, $\beta$-sheet, and coil. 92
4.1 Seven sequences of the Crambin (1crn) protein family, aligned. 105

A.1 The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 5$. 169
A.2 The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 10,000$ and $r = 5$. 169
A.3 The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 6$. 170
A.4 The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 10,000$ and $r = 6$. 170
A.5 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 7$. .......................................................... 171

A.6 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family0. These results were produced with parameter settings $T = 10.000$ and $r = 7$. .......................................................... 171

A.7 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 10$. .......................................................... 172

A.8 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family0. These results were produced with parameter settings $T = 10.000$ and $r = 10$. .......................................................... 172

B.1 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 5$. .......................................................... 173

B.2 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 10.000$ and $r = 5$. .......................................................... 174

B.3 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 6$. .......................................................... 174

B.4 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 10.000$ and $r = 6$. .......................................................... 175

B.5 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 7$. .......................................................... 176

B.6 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 10.000$ and $r = 7$. .......................................................... 177

B.7 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 10$. .......................................................... 178

B.8 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the artificial dataset Family1. These results were produced with parameter settings $T = 10.000$ and $r = 10$. .......................................................... 178

C.1 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the HIV dataset. These results were produced with parameter settings $T = 10.000$ and $r = 5$. .......................................................... 180

C.2 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the HIV dataset. These results were produced with parameter settings $T = 10.000$ and $r = 6$. .......................................................... 180

C.3 The most likely correlated attributes. as estimated by the coincidence detection procedure. for the HIV dataset. These results were produced with parameter settings $T = 10.000$ and $r = 7$. .......................................................... 181
C.4 The most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 10,000$ and $r = 10$.

C.5 The thirty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 100,000$ and $r = 7$.

C.6 The first twenty-five of the fifty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 750,000$ and $r = 7$.

C.7 Continuation of the fifty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset: csets ranked 26 through 50. These results were produced with parameter settings $T = 750,000$ and $r = 7$.

C.8 The top thirty-five pairwise inter-column mutual information values for the HIV-V3 dataset, as estimated by our methodology as described in the main text.

C.9 The top seven pairwise inter-column mutual information values for the HIV-V3 dataset, as estimated by the Los Alamos group.
List of Figures

2.1 Five views of a protein molecule. ........................................ 12
2.2 Schematic of a protein backbone. ........................................ 20
2.3 Two Graph-Theoretic Representations of A Protein Structure. ...... 25
2.4 A two-stage approach to protein tertiary structure prediction. ...... 40
2.5 Inter-residue correlations and conserved structural constraints. ..... 43
2.6 The integration of multiple knowledge sources in protein structure prediction. . 45

3.1 Conventional neural network training for prediction of conventional secondary-structure classes. ........................................ 50
3.2 Use of a correlation-based agreement measure to train coupled neural networks. ... 53
3.3 Compatible clusterings in sequence and structure spaces. ............. 54
3.4 Illustration of the coupled feed-forward networks used in Method 1. ... 59
3.5 Simplified schematic of the neural network implementation of gaussian mixture modelling used in Method 2. ......................... 78
3.6 Illustration of the neural network implementation of conditional mixture modelling used in Method 3. ................................. 87

4.1 The power set of a set with $N = 6$ objects, depicted as a lattice. .... 115
4.2 Lattice-diagram depiction of the n-grams, for all sizes $n = 1, 2, \ldots, 6$, from the toy dataset. ........................................ 116
4.3 Lattice-diagram depiction of all possible pairwise correlations between columns (residue positions) in the toy dataset. .............. 117
4.4 Lattice-diagram depiction of a partition of the variables in the toy dataset. ... 118
4.5 Simplified schematic of two possible neural network latent variable models for the toy dataset. ........................................ 130
4.6 Operation of the Coincidence Detection Method. ......................... 139
4.7 3D plot of the estimated probability of a correlated $k$-tuple of attributes being "masked" by noise. ................................. 145
4.8 Plot illustrating the growth of the Csets data structure as a function of the number of iterations of the Coincidence Detection procedure, on a dataset with uncorrelated columns. . 152
4.9 Plot illustrating the growth of the Csets data structure as a function of the number of iterations of the Coincidence Detection procedure, on the Family1 dataset. ... 153
4.10 Plot illustrating the growth of the Csets data structure as a function of the number of iterations of the Coincidence Detection procedure, on the HIV dataset. ... 153
4.11 Plot illustrating the growth in Csets as a function of the number of columns, on a dataset of independent columns. ......................... 154
Chapter 1

Introduction

1.1 The Topic and Scope of the Thesis

The protein structure prediction (PSP) problem is a crucial one in the emerging field of computational molecular biology. In order to gain a comprehensive and deep understanding of the structure and evolution of biological organisms, and to make sense of the flood of sequence data — DNA, RNA, and protein — produced by the Human and other Genome Projects, we must be able to infer the relationship between macromolecular sequences and three-dimensional (3D) structures. In order to design (or artificially evolve) new, safer, and more effective drugs and vaccines that interact with particular targeted substances at the molecular level, we must understand how the insertion, deletion, or substitution of particular sub-elements in a protein chain will affect the overall structure and function of the molecule.

The analysis of protein sequences and prediction of their structures poses daunting and intellectually exciting problems for computational science. The growth of sequence databases and the need for fast detection of similarities between sequences is driving research into data compression, string-matching algorithms, and network and database theory and design. The prediction of native or thermodynamically optimal structural conformation through detailed modelling of the protein folding process at the atomic level requires sophisticated statistical and numerical methods and may also await breakthroughs in high-speed and parallel computation.

Where classical or quantum physical modelling becomes intractable, or where gaps exist in the theoretical understanding of protein folding, there is a need for statistical estimation and inference — machine learning — methods to assist. And where there exist datasets of known sequences and structures, there is an opportunity for such methods to assist.
Artificial Intelligence (AI) and machine learning (ML) researchers find in this domain several factors which provide challenging research topics: the high dimensionality of input and output representations, the huge sizes of some of the datasets (sequences) along with a paucity of other kinds of data (precisely-determined tertiary structures).

Even if one narrows one’s view of the protein structure prediction problem to focus on AI and ML methods, the topic is too broad and deep to be covered in one thesis. The research described herein therefore focuses on a few key aspects of the application of ML to the problem of predicting protein three-dimensional structure from amino acid sequence. The thesis presents a particular mathematical treatment of PSP and provides sufficient background on protein science to make clear how novel ML methods for particular subproblems can contribute to solution of the overall problem.

The main logical thread running through this text is as follows: Protein structure prediction, when expressed in its most general mathematical form, is an intractable problem in combinatorial optimization. It may usefully be viewed as the search for, or piecewise construction of, a graph that expresses inter-atomic relationships in a folded protein and that optimizes some thermodynamic or other objective function. There exist regularities in the domain — biophysical and evolutionary constraints — that enable the use of particular heuristics that reduce the search space and make the task feasible in practice. These heuristics are grouped into one or more of what we might characterize as learning, locality, selectivity, and abstraction. Consideration of these ideas leads to particular ways of decomposing the PSP problem into interesting, tractable subproblems. Two such subproblems serve as the foci of new research described in Chapters Three and Four, respectively.

The major theme unifying research Chapters Three and Four is the idea of a motif, an abstraction of observed patterns or objects. The thesis first provides a definition of this term within the computational molecular biology domain, some background on the use of motifs in protein structure prediction, and an explanation of how motifs relate to supervised and unsupervised machine learning. Then the main research chapters examine criteria for choosing “good” motifs and present novel methods for motif discovery in the protein structure domain. The presented methods should prove useful in other domains as well.
1.2 Overview of the Thesis

The computational problems explored in this thesis are to be attacked within the problem domain of protein structure prediction and analysis. So let us examine the primary assumptions and results of the thesis from the biochemistry/biophysics viewpoint.

1. The "native" structure of a protein — that which one tries to predict — is a well-defined concept. This is an assumption, a working hypothesis, and is not beyond controversy (as discussed in Chapter 2), but good arguments can be made that it is a reasonable basis for continued work.

2. There is an important role for ML techniques in the PSP problem. The improvements gained already by ML-based methods over long-used "hard-wired" methods provide some evidence in favor of this view.

3. The means for applying ML methods exists, in the growing collections of data from which known (sequence, structure) training datasets can be prepared. Moreover, there is reason to hope for some success, because the complex thermodynamic and kinetic constraints operative in protein folding do leave their "footprints" in the datasets.

4. The complexity of the structure prediction problem, as well as the paucity of solved structure data relative to the high dimensionality of sequence and structure representations, mandate a "divide and conquer" approach. It is necessary to define (at least) one intermediate level of representation — "secondary structure" — between sequence and full 3D structure, and to perform the prediction in (at least) two stages. For example, in the first stage one builds a module to map sequence to secondary structure, and in the second stage one builds another module to map secondary to tertiary (global three-dimensional) structure.¹

5. The patterns of local hydrogen bonding, e.g., "α-helix" and "β-sheet", defined as secondary structure by structural biologists, provide a good intermediate level of representation. However, it is desirable and possible to define secondary structure classes that are more predictable from amino acid sequence. In general, one can propose particular criteria for local structure motifs, define information-theoretic objective functions that enforce such

¹One should not see a contradiction between items 3 and 4 in this list; there probably is enough data, just not enough to permit employing it unwisely.
criteria, and then optimize the objective function in order to discover motifs that fit the criteria.

6. There are interactions between sequence-distant amino acid residues in the protein chain, sometimes detectable as correlations between positions (columns) in a set of aligned sequences from a protein structural family, that play an important role in determining structure and function. Discovered correlations may represent an evolutionary history of compensatory mutations, and may provide useful features in models of protein structural/functional families, but are ignored or mishandled by most ML classification methods, in part because of the high computational complexity of searching for \( k \)-tuples of correlated positions.

7. Once predictions or estimations have been made of local structural pieces and of particular sequence-distant interactions, the search for global optimal tertiary structures can be greatly reduced, through the use of geometric (steric) and topological constraints as well as pattern-matching and pattern-retrieval methods that exploit the observed correspondence between most frequent configurations and most energetically favorable ones.

8. The protein folding process is best considered as an interaction between several levels of organization, and as a simultaneous satisfaction of both bottom-up and top-down constraints. Computational modelling and machine learning methods applied to the problem should reflect this. An emerging consensus in computational molecular biology argues for a broader view of structure prediction, in which protein representations contain several levels and several different "views" on the data, and in which "prediction" may occur between any two or more levels or views: top-down and bottom-up, global or local, sequence to structure or vice-versa. The methods presented herein are in accord with this broad interpretation of protein structure analysis.

The focus of the investigations described in the thesis is on items 5 and 6 of the list given above. New methods are proposed and evaluated for the automatic discovery of two kinds of protein motif: local structure motifs with associated sequence motifs (which may be viewed as novel secondary-structure classes), and motifs characterizing long-range correlations between amino acid residues in a set of evolutionarily-related sequences.

For the local structure motif problem, this thesis presents adaptive algorithms that evolve new secondary structure classes that are more predictable from local amino acid sequence in-
formation. Ongoing work focuses on defining secondary structure classes that are also more predictive of tertiary structure, and that might therefore provide a generally better representation language for use in a multi-stage structure prediction system.

For the motifs characterizing non-local interactions, the thesis presents some fast heuristic methods for correlation-detection, as well as bounds on their rates of error and computational complexity.

1.3 Translation from a Biophysical into a Mathematical Model

The thesis attempts to define and demonstrate several novel methods for two important subproblems of PSP, as well as surveying relevant work by other researchers, and showing how the different subproblems and approaches fit together. It is easiest to do this within one unified mathematical framework. Such a framework is developed, along with some necessary biological background, in the next chapter. A few conceptual steps are required to go from an idea of a protein as a biophysical entity to an understanding of the mathematical treatments in Chapters Three and Four. These steps are outlined here:

1. **Represent protein structures as graphs.** Taking the idea of a protein as a chain that folds back on itself, we let the nodes of a graph represent individual amino acids and let each edge between two nodes represent "contact", "spatial proximity", or "interaction" between the subunits in 3D space.

2. **Consider protein structure prediction as optimization over graphs.** Each graph has a cost, corresponding to the free energy of the represented molecular structure or to some heuristic objective function that approximates the free energy. Prediction is then a search for the lowest-cost graph(s).

3. **Use a probabilistic modelling framework to describe the optimization and results.** Instead of using "energy" per se, treat "optimal structure predictions" as "most likely structural models given the sequence (or other input) data". The approach makes use of the fact that the graphs representing existing protein molecules are a highly specialized subset of possible graphs. Particular, non-uniform distributions over the features of such true protein graphs may be empirically observed and exploited in order to cut down the otherwise exponential search for the optimal conformation for a given protein.
4. **Treat machine learning as the inference of parameter values in these structural models.** Such inference may be done in any of a number of ways, under Bayesian, maximum likelihood, maximum entropy, or other methodologies.

Within the mathematical framework defined in Chapter Two and outlined above, the overall PSP problem is seen (roughly) as one of finding the most likely structural model given the sequence data and any prior knowledge. The representation of possible structures as graphs makes clear the combinatorial complexity of the problem. This complexity, combined with the need for reliable estimation of model parameters, makes clear the need to divide PSP into subproblems.

Once formulated in terms of a contact graph or matrix of inter-segment distances, PSP becomes a problem of predicting which pieces will tend to be near which other pieces. The work described in Chapter Three aims at finding empirical folding tendencies over sequentially local regions of the protein, based upon joint probabilities in sequence-structure space. The work of Chapter Four aims at finding tendencies of a few sequentially distant amino acid residues to interact as spatial neighbours, based on covariance in pairs or $k$-tuples of sequence positions over the course of molecular evolution. The underlying basis for both methodologies, indeed for all ML approaches in this domain, is the evolutionarily-honed nature of all natural extant DNA, RNA, and proteins: features that are observed frequently are those that have been conserved throughout molecular evolution, and are often those that are most structurally stable and functionally efficient.

A formal mathematical statement of two crucial subproblems, addressed respectively in Chapters Three and Four, is given at the end of the next chapter, after the necessary definitions. At this point it should suffice to say that both problems concern the estimation, from datasets, of some higher-order joint probability densities $p(x_1, x_2, \ldots, x_k)$ over features $\{x_i\}$ in sequence-space and/or structure-space. The two proposed methodologies both employ latent variables in their respective estimation tasks. In one case, the latent variables correspond to protein secondary structure classes; in the other case, the latent variables correspond to putative evolutionarily-conserved 3D structural constraints between subunits distant in the protein sequence.
1.4 Organization of this Thesis

The structure of this document is as follows:

Chapter 2 sets the stage for the main results chapters by providing some background on computational molecular biology and the application of ML to protein structure prediction. An argument in favor of the general approach taken in this research is given in light of biophysical and mathematical theories and results. A single mathematical framework, combining graph theory and probabilistic models, is defined.

Chapter 3 describes research into the use of coupled learning modules in developing new secondary structure classes that might provide for a better intermediate representational language within the multi-stage protein structure prediction task. One of the novel approaches turns two supervised learning systems into one large unsupervised (or self-supervised) learning system. Other explored methods employ joint and conditional probability density estimations over two input spaces: sequence fragments and tertiary structure fragments. The chapter focuses first on finding secondary structure classes more predictable from sequence and then on finding secondary structure classes more useful in predicting tertiary structure.

Chapter 4 presents an analysis of higher-order features in protein sequence data, corresponding to \( k \)-ary correlations between sequence-distant amino acid residues. The chapter discusses the potential importance of such features in structure prediction. The prohibitive computational cost of traditional approaches is discussed, and new, efficient heuristic methods for the detection and use of such long-range correlations are presented and analyzed.

1.5 Contributions to Computer Science

Among the contributions this thesis makes to the field of computer science are the following:

1. In Chapter Three we report a couple new formulations and sets of experiments in the search for computational learning methods that combine information from two or more sources or modalities. In our case the sources are protein primary and tertiary structures, from which we derive novel "secondary structure" classes, but the underlying mathematical and computational problems are not wholly different if one is concerned instead with visual and auditory modalities, or with visual inputs from multiple cameras.
2. The problem of finding all significant correlations among pairs or $k$-tuples of attributes in a database is ubiquitous in the computational sciences and in medical, industrial, and financial applications. We propose and evaluate a probabilistic algorithm in Chapter Four that has the interesting property of finding significant higher-order $k$-ary correlations, for all $k$ such that $2 \leq k \leq N$ in an $N$-attribute database, for the same computational cost of finding just significant pairwise correlations. Moreover, $k$ need not be fixed in advance in our procedure, in contrast with other known procedures. The procedure was deigned for the task of finding conserved structural relationships in aligned protein sequences, but may have more useful application in other domains.

3. In addition to the two specific problems and proposed solutions, there are perhaps more general insights to be gained from our analysis of a complex, seemingly intractable combinatorial search problem and our decomposition of it into a set of smaller, simpler subtasks, some of which are amenable to heuristic solution based upon unsupervised machine learning and database probability density estimation.
Chapter 2

Motivation, Background, and Mathematical Framework

2.1 Introduction

This chapter is intended to provide sufficient background to enable the reader to understand the particular problems addressed in Chapters Three and Four and to assess the novel algorithms proposed in those chapters. The essential combinatorial nature of the protein structure prediction problem is first revealed through the molecular biophysics of protein folding and is then reviewed within a more formal mathematical framework of graph theory and combinatorial optimization. From these two perspectives, one can clearly see the both the fundamental intractability of the PSP in the general case and the plausibility of getting around this computational barrier with principled heuristic short-cuts. Consideration of two particular kinds of heuristic methods leads to the definition of and attack on the subproblems — secondary structure motif definition and long-range inter-residue correlation detection — explored in Chapters Three and Four respectively.

2.2 DNA, RNA, and Proteins

Deoxyribonucleic acid (DNA) is the medium of storage of genetic information in living cells. The "central dogma of molecular biology" states that genetic information flows from DNA to RNA (ribonucleic acid) to proteins. The units of genetic information contained in the DNA are called genes. DNA acts as a template for the creation of an RNA copy of a gene; this process is called
transcription. The RNA copy is in turn translated into a protein molecule. Proteins are the major actors in cells, where they have structural, signaling, and catalytic functions. Exceptions to the central dogma exist — for example, there are certain viruses that have RNA-encoded genes from which DNA copies are made, first reversing the flow of information before rejoining the pathway (Baltimore, 1970). Still, the central dogma serves as a useful generalization.

DNA is a long molecule displaying a double helical three-dimensional structure (Watson and Crick, 1953). The subunits that comprise each strand of the double helix are called nucleotides. Each of which is in turn composed of a sugar (deoxyribose), a phosphate, and a base (one of guanine, cytosine, adenine, and thymine). From an information-theory point of view, a strand of DNA is a linear sequence of symbols, over the alphabet G, C, A, T. A crucial aspect of DNA helix structure is that guanine (G) on one strand pairs with cytosine (C) on the other strand: and adenine (A) pairs with thymine (T). This base-pair complementarity allows fidelity in DNA replication and in the copying of a DNA template into an RNA molecule. An RNA molecule is similar to DNA, except that it is a single strand, the sugar is ribose, and it uses the base uracil instead of thymine. In RNA, uracil pairs with adenine. The length of a DNA molecule (or RNA molecule) is defined in terms of the number of bases (sometimes, for DNA, called “base-pairs” and abbreviated bp).

Each cell typically contains one or two copies of the entire set of genetic information particular to that organism, which is called its genome. To be precise: each germ. or sex, cell contains one copy of the genome and each somatic, or body, cell contains two copies. In eukaryotic organisms (almost everything besides bacteria), the bulk of the DNA is carried in the nucleus, and is organized into linear segments known as chromosomes. The approximately 3 billion bases in the human genome are spread out over 23 chromosomes.

All DNA is unbranched, and therefore all genomes can be organized into one-dimensional maps. A genetic map is the order of genes along a chromosome. The genes are the segments of DNA that code for particular proteins or structural RNA molecules. Preceding and following each gene are regulatory regions which control the gene’s expression, i.e., its transcription into RNA. Regulatory regions include promoters, enhancers, and terminators. It is estimated that the human genome contains more than 100,000 genes (Lipton, Marr and Welsh, 1989). the coding region of each having a size of about 1-2 kilobases (kb). However, much of the genome is noise. Sequence and mapping analysis carried out so far indicates that humans, like other mammals, have a genome which consists of at least 90% (and probably closer to 95%) uninterpretable.
possibly biologically meaningless sequence.

The coding sequences that make up a gene are not necessarily contiguous. In eukaryotes, an individual gene is typically interrupted by *intervening sequences,* or *introns.* The coding regions are called *exons.* During transcription, an RNA copy of a gene containing both exons and introns is synthesized. Subsequently, the introns are spliced out of the molecule, producing a shorter RNA, in which the coding regions are contiguous. During the next stage, translation, the spliced RNA is decoded and a chain of amino acids, a protein, is assembled. The bases of the RNA are read in serial order in groups of three. Each triplet of nucleotides, called a *codon,* specifies one amino acid. In principle, each mRNA sequence can be read in any one of three different *reading frames,* depending upon where on the molecule the decoding process begins. In almost every case, only one of these reading frames will produce a functional protein. Since there are no punctuation signals except at the beginning and end of the mRNA message, one reading frame is set at the initiation of the decoding process and is maintained throughout.

The end products of the process of genetic information flow, proteins, fall roughly into three different groups: enzymes, which catalyze the reactions responsible for nearly all biological activities; structural proteins, which give cells and subcellular compartments shape and motility; and small proteins like the polypeptide hormones, which function as messengers of physiological information.

In referring to particular units in a particular protein, one speaks of amino acid *residues,* as in “The 27th residue in this sequence is a lysine.” For proteins, length is defined in terms of the number of amino acid residues. An amino acid consists of an *alpha carbon* (α-carbon)\(^1\), to which the following four components are covalently bonded: an amino group, a carboxyl group, a hydrogen atom, and a particular “R-group” by which each of the commonly occurring amino acids is distinguished. Protein chains are formed when the amino group of one amino acid is joined to the carboxyl group of another amino acid by a covalent bond called a *peptide* (or *amide*) *bond.* Hence proteins are sometimes known as “polypeptides”. The polypeptide chain of a protein consists of a regular repeating part, called the *main chain* or *backbone,* and a variable part comprising the distinctive *side chains,* the R-groups, of the constituent amino acids.

Since 1953, when Sanger determined and published the complete amino acid sequence of insulin, it has been understood that an amino acid sequence uniquely determines a particular

---

\(^1\)This designation should be understood to mean, simply, “the first carbon”, and should not be confused with α-helix, a kind of secondary structure comprising many contiguous residues.
Figure 2.1: Five views of a protein. The protein is Crambin (1crn), a very small, mostly-hydrophobic protein found in cabbage seeds. 1) The upper-left quadrant features a space-filling model of the tertiary structure of the protein. The main carbon/nitrogen/oxygen backbone atoms are shown in grey/white. Each of the amino-acid side-chains is shown as one moulded piece in a particular color. Acidic residues are various shades of red; basic residues are shades of blue; hydrophobic residues are gray/green; amino acids serine (S), threonine (T), and Cysteine (C) are in the yellow to orange range. 2) The picture in the upper-right quadrant is a line drawing, with the same color scheme and from the same coordinate viewpoint. The line drawing allows one to see more of the interactions within the core. 3) The lower-left quadrant displays a ribbon diagram of the protein, again from the same angular viewpoint. The ribbon diagram clearly reveals the two main \( \alpha \)-helical segments of the 3D structure.

4,5) The lower-right quadrant presents both the primary structure (amino acid sequence) and the secondary structure. The amino acid sequence, represented with the 20-symbol amino acid alphabet, is broken into two lines, and the secondary structure encoding is written directly below it (also broken into two lines). The symbols used for the secondary-structure description are from Kabsch and Sander's DSSP program (Kabsch and Sander, 1983a). Basically, "E" and "S" mean \( \beta \)-sheet. 'H' and 'G' mean \( \alpha \)-helix. 'T' is turn and '-' is coil. The correspondence of the primary- and secondary-structure text sequences to the pictures may be seen by first noting that the beginning of the sequence corresponds to the darkened, partially-occluded "ribbon-end" in the ribbon diagram at lower-left.
protein. The function, the biological role or activity of a protein, derives from its specific conformation, the 3D arrangement of atoms. This conformation is possible because many of the covalent bonds in the polypeptide chain are rotationally permissive. Of course, there are limitations to the diversity of protein shapes: not all conformations are theoretically possible, not all theoretically possible structures actually occur in nature, and not all possible and naturally occurring structures are equally likely or equally frequent.

Under appropriate conditions, a protein molecule assumes a particular three-dimensional structure, its native (or functional) conformation. The finding (Anfinsen, 1973) that denatured proteins can re-fold into their native conformations, led to the working hypothesis that folding is dependent only upon the chemical properties of the amino acid side chains in the sequence, and the interactions deriving from their place in the sequence and the immediate environment of the protein molecule. It is further hypothesized that the native state corresponds to a thermodynamically-optimal state; whether it is a global or local optimum, and the degree to which the actual folding pathways are important, remain key and controversial questions (Creighton, 1988).

The assumption that a single native structure exists for each protein, and the relative emphasis placed on different forces at work on the folding protein, are subject to lively ongoing debate. These issues have direct bearing on the use of computational modelling and machine learning in structure prediction: they are discussed in the next section.

2.3 Protein Structure and the Folding Process

2.3.1 Levels of Organization in Protein Structure

A protein's primary structure is its amino acid sequence. Secondary structure refers to the local steric arrangement of the constituent amino acids and their side chains, and often to the pattern of hydrogen bonding (H-bonding) between atoms in different residues' side chains. Tertiary structure refers to the global three-dimensional (3D) arrangement of the amino acids.

Although it is assumed that all of the information required for the folding of a protein is encoded in the amino acid sequence, we are not yet able to decode this information and reliably predict the 3D structure from the sequence (hence the remarkably active research area).

---

2 Some proteins need the help of a chaperonin, another protein that facilitates folding (Gething and Sambrook, 1992).
Currently, the only known ways to determine a protein's structure are to perform either X-ray diffraction (which requires crystallizing the protein molecule, no mean feat) or nuclear magnetic resonance (NMR) imaging analysis (which has its own set of empirical limitations).

X-ray diffraction patterns served as the initial data from which protein structure pioneers like Linus Pauling constructed hypothetical models of conformation. Pauling (1951) discovered that, although each protein seems to have a unique conformation, several particular patterns occur repeatedly in most known proteins. Two prominent periodic structures formed through hydrogen bonding between relatively local residues are \( \alpha \)-helix and \( \beta \)-sheet (which is composed of one or more \( \beta \)-strands).

The \( \alpha \)-helix is characterized by a periodic pattern of hydrogen bonding between an \( i \)th residue in the sequence and the \( i + 4 \)th. It therefore appears in a molecular graphics rendering as an extended helical structure that seems to wrap around a central (but non-existent) rod. Helices, like \( \beta \)-strands, are found in many complex arrangements in different proteins. One common motif is a helix bundle, in which two or more helices are arranged parallel to each other.

In the \( \beta \)-sheet, extended strands of the polypeptide chain (of lengths ranging from 5 to about 14 residues) lie adjacent and parallel to each other and are cross-linked by hydrogen bonds between spatially (but not sequentially) very local residues. The strands may be arranged into sheets in a parallel or an anti-parallel manner. "Parallel" means that the residue numbers in two adjacent strands are both ascending or descending when viewed from the same direction. Two or more sheets may themselves be arrayed in parallel, forming sandwich-like global structures, or a sheet may "roll up", such that the strands form a barrel-like structure. Other, more complex arrangements are also common. Although there are all-\( \alpha \) and all-\( \beta \) proteins, many proteins have conformations combining these two forms.

Between well-defined \( \beta \)-strands and \( \alpha \)-helices are found \textit{turns} of various kinds, as well as longer segments of the protein chain that do not fit neatly into another category, and these are called \textit{random coil}. These relatively local arrangements of the amino acid residues — \( \alpha \)-helices, \( \beta \)-strands, turns, and coil — are known generally as the secondary structure types. (Turns are often included in the random coil category, as in Chapter Three.)

Although the term "local" is not unambiguous\(^3\), it would be a mistake to suggest that the

\(^3\)Here we speak primarily of "local" structure in terms of the sequential distance, which may be considered 5 residues or 25 residues by different people in different contexts; and this kind of locality is very different from "local" interactions in an interatomic distance matrix; then, too, there is the concept of "local minima" in an
distinction between secondary and tertiary structure is arbitrary.

2.3.2 What Drives the Folding Process (and Does It Matter)?

The still-controversial idea that secondary structures might form stable or meta-stable arrangements separate from, or temporally prior to, tertiary structures, is central to an understanding of protein folding and protein structure prediction. There seems to be a consensus supporting the idea that protein conformation is almost wholly a combination of two major components: a tendency of non-polar amino acid side-chains in the protein to avoid water molecules (the hydrophobic effect), and the tendency of some amino acid side-chains to form H-bonds with the side-chains of other amino acid residues. However, there is anything but a consensus on the relative importance of these two tendencies.

Evidence of the importance of H-bonds can be found in the observation (Chothia, 1976; Chothia, 1984) that in naturally-occurring proteins virtually every potential H-bond donor or acceptor is H-bonded. Most of the well-characterized and structurally significant hydrogen bonds are of the standard secondary structure types. However, H-bonding is probably not the dominant force driving a protein chain to fold into a particular native conformation. This may be in part because: (1) H-bonds are relatively weak and thus individually have little or no net beneficial effect on free energy, and (2) H-bonds act only over short distances.

Protein biophysicist Ken Dill writes. “A criterion for a fundamental driving force is that it must explain why the folded state is advantageous relative to the unfolded one. [It is] argued that hydrogen bonding would not satisfy this criteria because there [is] no basis for believing that the intra-chain H-bonds would have lower free energy than those of the unfolded chain to water.” Further, “the reason that only one native structure is encoded in the amino acid sequence may be largely attributable to the hydrophobic interaction: there are only a small number of ways to configure a chain [so as] to maximize the number of non-polar contacts” (Dill, Fiebig and Chan, 1993a).

When considering the importance of different forces and effects in protein folding, it is impossible to avoid the issue of kinetics. Put simply, does a protein really achieve a global thermodynamic minimum, or can it get caught in local minima? And if it can become trapped in local minima, what kinds of folding pathways — trajectories through state space — lead it
into, away from, or perhaps out of such local minima (Bryngelson et al., 1995)? If there is both local nucleation of folding (as in hydrogen bonding in secondary structure formation) and global nucleation of folding, do the local and global forces conflict? If so, then the protein folding is an example of a frustrated system; if not, then the system is characterized by cooperativity. Dill et al. (Dill, Fiebig and Chan, 1993a) and Srinivasan and Rose (Srinivasan and Rose, 1995) provide theoretical arguments and empirical results indicating a high degree of cooperativity in the folding of several diverse proteins.

In this thesis we shall take the easy way out of this particular arena of controversy by relying upon the following: (1) For at least some proteins, there is strong evidence for a dominant native folded state and fast, simple access to this state (Unger and Moult, 1993). (2) In any case, the forces driving a folding protein through any particular kinetic pathways produce different results on different proteins and at different regions of a single protein, and such selectivity must be somehow encoded (through evolutionary trial-and-error) in the amino acid sequence and in the relationship between the sequence and the set of accessible states. Therefore, (3) patterns detected in associated (sequence, structure) data can in principle be detected and used to assist in the prediction of some subset of the accessible conformational states from the amino acid sequence.

What does all of this imply for computational scientists looking for useful and meaningful representations of protein structures?

Suppose one accepts the Dill view that optimizing a pattern of H-bonding is easy compared to optimizing the burial of hydrophobic residues away from water in the protein structure interior, and, that therefore, the hydrophobic effect is more likely to be the driving force in protein folding. This does not mean that particular H-bonding patterns are not much more favorable than others, given a rough arrangement of substructures brought into proximity by the hydrophobic effect. Then, although individual residue-residue H-bonds may not be significantly more favorable than water interactions for either or both residues would be, nonetheless a particular pattern of H-bonding (particular secondary structures) might be significantly more favorable than other H-bonding patterns within the same overall, rough tertiary structural arrangement.

Such a view would seem to unite the theory that the hydrophobic effect nucleates and drives coarse-scale folding with the empirical observation of stable secondary structures in "molten globule" and partially denatured states. Such a view would also lend support to the use of secondary structure prediction in global tertiary structure prediction, while also explaining why
standard “local window”-based approaches to secondary structure prediction apparently cannot do better than about 75% predictive accuracy (but see Chapter 3). That is, secondary structure is important, and correlations between sequence and secondary structure even in local windows (fragments composed of sequentially-local residues) can indeed be detected and exploited, but only up to a limit imposed by the neglect of longer-range tertiary structural interactions (presumably driven by hydrophobic interactions and affected by packing constraints).

This is essentially the stance favored within this thesis project: (1) Predict local secondary structure as well as possible, and use these noisy predictions to help constrain predictions for tertiary structure, but (2) keep in mind that global tertiary constraints — manifest in particular sequence-distant, structure-proximal interactions — select between different locally optimal secondary structures. The new methods and experiments described in this thesis focus on automated discovery of exactly these two kinds of patterns: local secondary structures and sequentially-nonlocal interactions.

One need not take sides in any debate on the merits of a particular model of cooperative folding in order to recognize and exploit the benefits of cooperativity in general. To the extent that cooperativity holds true in protein folding, methods based on the use of partial and local "hints" in structure prediction (Sibbald, 1995) will succeed. We shall revisit this idea of hints in the context of more formal mathematical definitions of protein structures and prediction.

2.4 Prediction of Protein Secondary and Tertiary Structure

The previous sections told us what a protein is and how it (probably) folds. This section provides a brief introduction to the “why, what, and how” of the prediction of protein conformations.

2.4.1 Why Predict?

Structure prediction is necessary because the knowledge of protein structure (and hence function) is crucial to an understanding of biology and to progress in biotechnology and medicine, and because it is currently the only (relatively) quick and inexpensive way to discover the native structure of a newly discovered protein.

The deduction of protein structures from X-ray crystallographic and NMR imaging data is difficult, slow, and costly. A potentially useful application of scanning-tunneling microscopy to the problem is probably not feasible for at least several years. About 1200 to 2000 unique protein and poly-peptide structures have been solved, while over 200,000 sequences have been
archived into the global databases. All of this points to a need for fast and reliable algorithms for prediction of protein structure from amino acid sequence.

2.4.2 What is Structure Prediction?

Protein structure prediction is the process of inferring a partial or complete 3D conformation from an amino acid sequence. The term is usually taken to mean tertiary structure prediction, in which case a prediction system takes a representation of an amino acid sequence and returns a representation of a tertiary structure. Several possible representation schemes are presented below. One may also speak of secondary structure prediction, which is usually understood to mean the process of inferring, from an amino acid sequence, one of the traditionally-defined secondary structure classes, α-helix, β-sheet, or random coil. Secondary structure prediction may be viewed as a simpler task and an important precursor to tertiary structure prediction, and has become, fortunately or unfortunately, a popular "stand-alone" benchmark problem in AI/ML research.

In order to understand the computational task of PSP, it is necessary first to consider the inputs and outputs. The choice of representation of the inputs and outputs has crucial bearing on the success or failure of structure prediction.

A protein sequence may be represented as a finite sequence of symbols $\tilde{s} \in (\mathcal{A})^N$ where $N$ is typically between 60 and 400, and

$\mathcal{A} = \{A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y\}$ is the set of twenty commonly occurring amino acids.

For protein tertiary structure, the most obvious representation is perhaps a list of the coordinates in $\mathbb{R}^3$ of the protein's amino acid residues, in sequential order. This representation is of size $O(N)$. A serious drawback to this scheme is that a protein's coordinates are not absolute or invariant under rotation: two different views of the same protein will "look" very different.

Another possible representation of tertiary structure, which in contrast is invariant to rotation, is an $O(N)$-sized listing of the $\phi$ and $\psi$ angles (Schulz and Schirmer, 1979) describing the relative orientation of peptide units along the protein backbone. (See Figure 2.2.)

A much better, and widely used, scheme is an interatomic distance matrix, which represents 3D Euclidean distances (in angstrom units $\alpha A$) between the alpha-carbon atoms in all $i,j$ combinations of amino acid residues in the protein sequence. This 2-dimensional upper-right
triangular matrix is an array $A_{ij}$, $i = 1, 2, \ldots, N-1$, $j = i+1 \ldots N$, where $N$ is the length of the sequence. It is an important method for displaying aspects of the molecular structure of proteins and ribonucleic acids, for it makes visible the local 3D interactions between substructures that may be distant in the sequence.

A type of truncation of the distance matrix gives us a contact matrix, in which $A_{ij} = 1$ if residues $i$ and $j$ are within $\delta$ distance of each other in the 3D structure, and $A_{ij} = 0$ otherwise. Both types of inter-atomic matrices have size $O(N^2)$ (though there is much information-theoretic redundancy in such a representation (Chan and Dill, 1990)).

A useful way to think about the difference between the coordinate representation and $\phi\psi$ representations on the one hand and the distance matrix and contact map on the other hand, is to realize that the former describe the geometry of the protein molecule in 3-space, while the latter define its topology. As the distance threshold of a series of contact maps is set progressively higher, this topological representation becomes an increasingly compact and abstract representation of the protein conformation. A very coarse contact map might correspond to any of a number of different possible detailed geometric shapes. The issue of how much geometric information can be derived from a given topological description is addressed by Crippen and Havel (1988), among other sources.

2.4.3 The Difficulty of the Problem

Computational approaches to protein structure prediction face two basic and very difficult problems. First, the space of possible structural conformations available to a protein is enormous. For example, a protein of length 100 amino acid residues has on the order of 300 torsional angle degrees of freedom. If each torsional angle has access to 3 minimum energy states, then the entire protein has at least $3^{300}$ possible configurations.

Second, the degree of accuracy needed in theoretical folding models used for computer simulations and predictions is somewhat daunting. Small errors in parameter values can cause molecular dynamics simulations, for example, to diverge rapidly from the true state space trajectories. In principle, it is possible to predict atomic interactions and hence molecular structures by ab initio quantum mechanical methods. However, for large molecules such as proteins this is in practice intractable. Researchers therefore rely on classical mechanical models, complex formulations employing terms that represent covalent bonds, torsional angles, hydrogen bonds, van der Waals forces and electrostatic interactions. These parameters are theoretically calculated
Figure 2.2: This schematic drawing of a polypeptide (protein) backbone structure illustrates the significance of the $\phi$ and $\psi$ angles. Each amino acid molecule, or residue, in the chain features an $R$-group, or side chain, characteristic of the particular amino acid occurring in that position. The $R$-group is shown bonded with the $\alpha$-carbon atom. The covalent bonds comprising the backbone, and the bond angles, are fairly rigid. However, considerable rotation is possible around certain bonds. The angles $\phi_i$ and $\psi_i$ measure the torsion about the rotationally permissive bonds in the backbone of the $i$th residue. The backbone structure of a protein may be specified by the ordered list of coordinates in 3-space of the $\alpha$-carbons of the residues, or by a two-dimensional table of Euclidean distances between $\alpha$-carbons, for all pairs of residues, or by a list of the $\phi, \psi$ angles for all residues.

$ab initio$ from quantum models or estimated empirically, and the necessary approximations and inevitable estimation errors impair the reliability of the computational methods (van Gunsteren, 1988; Anfinsen and Scheraga, 1975; Creighton, 1988).

The importance of the hydrophobic effect in protein folding complicates the prediction problem by requiring one to account for the effects of the surrounding solvent (primarily water), and hence many more molecules than just a hundred or so amino acid residues.

Exactly how hard is PSP? Unger and Moult (1993) have shown that the prediction from sequence of a protein's thermodynamically optimal conformation (even ignoring the effect of water or other molecules in the environment) is $NP$-Hard (Garey and Johnson, 1979). (Other formulations and reductions have been given as well (Fraenkel, 1993; Ngo and Marks, 1992).) However, this result holds for the most general and abstract formulation of the problem, and ignores the possibility that the members of the set of naturally occurring proteins — a tiny
subset of the set of possible $N$-length strings over $A$ — share certain properties that might make their prediction more tractable. The result also applies only to exact and globally optimal solutions. In other words, no principle yet discovered rules out the development of *usefully* accurate and efficient algorithms and systems for protein tertiary structure prediction. This is discussed further in Section 2.6.5.

### 2.4.4 Approaches to PSP

Despite the difficulties, theoretical and computational approaches to protein structure prediction are useful and will assume an even larger role in the next few years. Many different methods have been applied to PSP: it is possible to define three broad approaches that encompass existing methods (Ponder and Richards, 1987b):

1. **Substructure condensation**: Predict secondary structure, then construct an approximate tertiary structure by packing secondary structure units together.

2. **Homology modelling**: Build a model of tertiary structure based on the recognizable sequence relationship between the new protein and another protein or protein family of known structure.

3. **Energetic methods**: Construct a potential function and follow the gradient to the optimal structure (energy minimization) or simulate the time-dependent motions of the chain during the transition from one conformation to an energetically more desirable state (molecular dynamics).

It is also instructive to distinguish three types of tertiary structure prediction problem, each successively an order of magnitude more difficult, and each currently approached by different classes of methods. (The reader should not infer a one-to-one correspondence between the three levels of difficulty and the three kinds of methods listed above.)

First, there is the problem of predicting a detailed conformation of a protein $P'$ from a protein $P$ with known structure, where the sequences are identical save for a few changes in amino acid residues. Various computer methods achieve a reasonable level of success on such cases.

Second, there is the problem of prediction of structures within a structurally or functionally related family of proteins, some of whose structures are known, even where few of the actual
amino acid identities are preserved among members of the family\(^4\). For both this task and the
easier task listed above, one may employ any of a range of methods, including *ab initio* calculations,
classical free energy minimization models, homology modelling and pattern-matching
from known structural pieces (Taylor, 1988; Zvelebil et al., 1987). The computationally intensive
energy-minimization and molecular dynamics methods have a fighting chance of success in
this case because one starts the search near in conformation space to the native fold, thereby
perhaps avoiding the notorious multiple minima problem.

Finally, there is the problem of *de novo* prediction of protein structures, without the use
of homologous or related proteins. There are currently no reliable methods for this task, though
it is believed by many that the use of some variant of substructure condensation is desired,
perhaps to be followed by a final refinement with molecular dynamics simulation.

Towards the more difficult end of the structure prediction spectrum is where AI and ML
methods become attractive. Heuristic rules and sophisticated pattern matching procedures can
be ideal complements to the energy-based methods — the former methods may provide good
clues regarding plausible long-range interactions and thereby constrain the initial state or search
space for the more physical methods, while the latter can refine the "good guesses" into detailed
conformation predictions.

### 2.5 The Sequence Alignment Problem

Several of the PSP subproblems and methods discussed in this thesis are formulated with the
assumption of a set of aligned sequences as a starting point. However, the reader is reminded
that the sequence alignment problem, that of finding an element-wise correspondence between
two or more sequences in the presence of insertions, deletions, and substitutions, is a non-trivial
and much-studied problem in computer science, bioinformatics, and speech recognition research.
We refer the reader to any of several good surveys of this research area (Sankoff and Kruskal,

\(^4\)The reader will note that, in the long run, the prediction of structure of most natural proteins may fall into
this class of manageable tasks. This is because eventually the crystallographers and NMR practitioners will likely
have "solved" a structure from nearly every protein family. This provides little hope for the foreseeable future,
however, and none for the prospect of designing and analyzing artificial proteins.
2.6 A Mathematical Framework for PSP

Following are some mathematical definitions and simple propositions that should enable the reader to understand, compare, and critique the methods described herein for various subproblems of the protein structure prediction problem.

Two different discrete models of protein conformation are presented here, each of which emphasizes different abstract features of molecular structure and both of which provide insights into the combinatorial nature of structure-space and the complexity of the search problem. The first ("contact graph model") emphasizes the topological constraints and the importance of pairwise and k-ary inter-residue interactions, whereas the second model ("lattice model") emphasizes the steric and geometric constraints on a folding protein.

2.6.1 First Things First: Primary Structure

A protein has a primary structure which is its amino acid sequence (called simply the "sequence" below) and a tertiary structure \( T \) (called simply the "structure" below). A sequence is represented formally as a finite string of symbols from a finite alphabet.

**Definition 2.1** A protein sequence (or amino acid sequence) \( \bar{s} = s_1s_2 \ldots s_N, s_i \in \mathcal{A} \).

Usually \( \mathcal{A} \) is taken to be the set of twenty naturally-occurring amino acids. However, it is often useful to create equivalence classes among the amino acids by grouping them according to various physico-chemical properties (Nakai, Kidera and Kanehisa, 1988). In such a case, \(|\mathcal{A}| < 20\).

2.6.2 Protein Structures As Contact Graphs

Next, a protein structure - the complete 3D global conformation - may be treated as a graph. The graph representation is derived from the contact map representation.

**Definition 2.2** A graph is a mathematical object \( G = (V, E) \) where \( V \) is a set of vertices (nodes) and \( E \) is a set of edges. An edge \( e_i \) connects exactly two vertices \( v_j, v_k \).

**Definition 2.3** The contact graph of a protein structure is a graph \( G = (V, E) \) such that: there is a vertex \( v_j \in V \) for each of the \( N \) amino acid residues in the protein: and for each pair of residues \( c_j, c_k \) such that their inter-atomic distance is less than some given \( \delta > 0 \), there is an
edge \( e_i \in E \) connecting \( v_j, v_k \). Associated with each node in the graph is a number \( i \in \{1 \ldots N\} \) indicating that the node represents the \( i \)th residue in the protein, and a letter \( a \in \mathcal{A} \) indicating the identity of the amino acid.

Note that our mathematical treatment of protein structures begins at a level of granularity coarser than a fully-specified atomic model. We characterize a protein molecule in terms of individual amino acid residues, rather than in terms of all the constituent atoms of the residues. The inter-atomic distances used in contact map representations may be calculated for just the alpha-carbon \((C_\alpha)\) coordinates, or for some estimated center of mass for each residue.

2.6.3 Protein Structures as Self-Avoiding Chains on a Lattice

The idea behind the lattice model is that the conformation of a protein, or of any polymer molecule, can be approximated by a representation of a chain of subunits — amino acids in this case — embedded in the discrete cells defined by a multi-dimensional lattice. This discrete approximation to a set of continuously-varying positions and hence bond angles allows one to analyze conformations by counting discrete states, and therefore allows simple formulations in terms of information theory and in terms of standard computational complexity theory. That the discrete lattice approximation is not too far from reality is supported by the following sorts of arguments: First, the discretization of dihedral bond angles can be justified by the empirical fact that real \( \phi \) and \( \psi \) angles in folded proteins do indeed cluster into narrow ranges of values (Schulz and Schirmer, 1979). Second, the modelling of each amino acid by only one point can be justified by the demonstration (Holm and Sander, 1991) that one can build a reasonable model for the full protein from the set of \( C_\alpha \) coordinates.

A lattice model for a protein is given as follows: We will consider a three-dimensional cubic lattice, of size \( n \times n \times n \), such that each face of the cube is an \( n \times n \) grid and each cell of the lattice is a unit \((1 \times 1 \times 1)\) cube with each of the eight vertices each being a lattice point. (See Figure 2.3.)

Each amino acid in the protein chain is represented by one element that can occupy one cell of the lattice. Only one element is allowed into each cell. The chain is folded on the lattice such that two contiguous amino acids in the sequence must reside in two neighboring cells of the lattice. Cells are neighbors if they share at least two lattice points. Note that each amino acid residue can therefore have 18 possible neighbors.

A lattice model can be formulated as a graph, wherein each node is a lattice cell, labelled
in some canonical way with $x, y, z$ coordinates, and with the directed edges connecting some set of pairs of nodes defining the sequence of amino acids in the embedded protein chain.

---

**Figure 2.3: Two Graph-Theoretic Representations of A Protein Structure**

**2.6.4 The Problem, Formally: Combinatorial Optimization**

Given the lattice formulation above, a general statement of the protein folding problem, with a general form for the cost $E$ ("energy") of a structure is one given by Unger and Moult (1993).
Definition 2.4 Given a sequence $\tilde{s} = s_1s_2\ldots s_N$, we seek a "folding function" $f : (\{s_i\}) \to \mathbb{Z}$ such that, given pairwise interaction costs $c_{s_i,s_j}$ and a positive function $g : \mathbb{Z}^3 \to \mathbb{R}^+$ of the 3-dimensional distance components between the folded elements: $g(\Delta_x(\tilde{l}, \tilde{k}), \Delta_y(\tilde{l}, \tilde{k}), \Delta_z(\tilde{l}, \tilde{k}))$, where $\Delta_x(\tilde{l}, \tilde{k}) = |\tilde{l}_x - \tilde{k}_x|$ and so on. and $\tilde{l} = \tilde{l}(s_i)$, $\tilde{k} = \tilde{k}(s_j)$ are lattice coordinates, $f$ would minimize the function:

$$E = \sum_{i=1}^{N} \sum_{j \neq i} c_{s_i,s_j} g(\Delta_x(\tilde{l}, \tilde{k}), \Delta_y(\tilde{l}, \tilde{k}), \Delta_z(\tilde{l}, \tilde{k}))$$

such that $\Delta_x(f(s_i), f(s_{i-1})) \leq 1$, $\Delta_y(f(s_i), f(s_{i+1})) \leq 1$, $\Delta_z(f(s_i), f(s_{i-1})) \leq 1$. for $i = 1, 2, \ldots, N$. (The latter distance constraints force contiguous chain elements to occupy neighboring lattice cells.)

The point here is not to meditate on the details of this particular definition, but rather to recognize the inherent combinatorics of the problem and perhaps to glimpse a means to circumvent the combinatorial obstacles. A central theme of our treatment is the replacement of theoretical, fine-grained substructure interaction costs (like the $c_{s_i,s_j}$ and $g(distance)$ terms above) with coarser-grained costs estimated from databases of known sequences and structures.

### 2.6.5 Intractability

Unger and Moult (1993) demonstrate that protein structure prediction is $NP$-Hard. First, they define the lattice-theoretic version of the prediction problem, and then they define the corresponding decision problem, DPF as follows: Given a sequence $\tilde{s}$ and a real parameter $B$, does there exist a folding $f$ such that $E \leq B$? They then exhibit a polynomial-time transformation of a known $NP$-Complete problem, Optimal Linear Arrangement (OLA), to DPF.

Fraenkel (Fraenkel, 1993) reduces the $NP$-Complete problem Three-Dimensional Matching (3DM) to a slightly different formulation of DPF. Other formulations may be given in terms of contact graphs.

Details aside, the essential exponential complexity of the protein structure prediction problem is intuitively fairly clear, whether one looks at structures in terms of lattices, contact maps, or other representations. For example, it has long been known that the number of possible foldings of an $N$-length chain onto a lattice — in other words, the number of self-avoiding walks on a lattice — is exponential in the length of the chain. It is given by $K\mu^N N^b$, where $K$ is a constant, $b$ depends on the dimensionality, and $\mu > 1$ is dependent on the type of lattice.
How do we reconcile these results with the observation in denaturation/renaturation experiments that globular proteins typically find their native structures very quickly and apparently without exhaustive search? Possible explanations include:

1. Nature can solve \( NP \)-Complete problems in polynomial time. Perhaps \( P = NP \) but we have not yet found the necessary reduction. Or perhaps quantum computation is the key, as suggested by Biafore (Biafore, 1993), among others.


3. The general, universal formulation is not applicable here. A problem is polynomial iff all of its instances are polynomial: it is \( NP - \text{Hard} \) if any of its instances are \( NP - \text{Hard} \), even if others may be polynomial. \( NP - \text{Hardness} \) reflects worst-case behavior, but the average case, under the assumption of some probability distribution, may be tractable. The exigencies of function and the selective processes of molecular evolution may preserve polynomial-time folding mechanisms while rejecting others.

4. The folding mechanisms, including particular kinetic folding pathways, may be encoded in the amino acid sequence, but the code remains unknown. Perhaps folding is not a search process at all, and exponential complexity is therefore not an issue.

These possibilities are not mutually exclusive.

2.6.6 Getting Around the Intractability

The protein molecule itself seems to avoid the exponential search that characterizes the most general and formal definition of the folding problem. How might we, as designers and users of prediction methods, avoid this fundamental intractability (and might we benefit from attempts to copy whatever tricks the molecule itself is using)?

We look into these questions in the following sections, and thereby come to understand the various structure prediction methods and how they fit together.

The crucial insight is that:

1. the evolutionary process has honed, over billions of years of trial and error, mutation and selection, the set of all possible amino acid sequences into a comparatively small set of
existing functional proteins made up of a comparatively small set of local domains and conformational patterns (Murzin et al., 1995): and

2. the work thus far in solving the primary, secondary, and tertiary structures of hundreds of proteins has resulted in a knowledge base from which we can identify this comparatively small set of patterns and use them to pare down the search space we face in structure prediction.

The methodology suggested by this insight goes loosely by the name of "empirical potentials" in the protein science culture, and by the name of "machine learning" and "motif extraction" within computer science.

2.7 Empirical Potentials, Probabilities, and Machine Learning

In the formal definition of PSP as a combinatorial optimization problem, we defined an energy function on protein structures. That function $E$ is a very general formulation, one among many possible ones. It captures the important idea that global potential energy is defined in terms of the geometric relationships between and interactions among subcomponents.

In the protein structure prediction literature, the art and science of defining "good" component structure costs is known as the problem of potentials. The cost function is often called the "potential" after the concept in physics, which is defined as the work required to bring objects (e.g., charges or masses) into particular positions within a field (e.g., electric or gravitational). Most research into protein structure prediction methods may be seen as part of a quest for (1) the best potential function, and (2) the best way to search for a structure which optimizes the potential function (Crippen and Maiorov, 1994; Grossman, Farber and Lapedes, 1995; Ngo and Marks, 1992). In this context, the two main obstacles to protein structure prediction are seen clearly (Crippen and Maiorov, 1994):

1. **Realism**: The potential function must resemble the free energy of a protein molecule in solution at least as far favoring the native conformation over all others.

2. **Feasibility**: The potential function $E$ must be defined so that calculation of $E(T)$ for a single structure $T$, and the search through the structure-potential space for the optimal structure(s), are computationally tractable.
In theoretically-defined potential energy formulations, for use in energy minimization as in molecular dynamics simulation, the internal geometry and inter-residue distance terms are derived from the strains represented by bond lengths and bond angles and the forces present in non-bonded interactions (primarily electrostatic and van der Waals). In contrast, empirical potential formulations simply use the empirical distributions of internal geometries and inter-residue distances.

It has been known since the time of Boltzmann that the energy of a state of a physical system can be related to the probability of the state:

\[ p_{hij} = \frac{1}{Z} \exp\left(\frac{-E_{hij}}{kT}\right), \]

where \( h, i, j \) correspond to variables of the system, \( K \) and \( T \) are Boltzmann’s constant and the system temperature, respectively, and \( Z = \sum_{hij} \exp\left(\frac{-E_{hij}}{kT}\right) \) is the partition function.

This relationship opens the door to methods that replace theoretically determined energy functions with empirical ones, by inverting the Boltzmann law and turning observed probabilities into pseudo-energies:

\[ E_{hij} = -kT \ln(f_{hij}) + kT \ln(Z), \]

where the \( f_{hij} \) terms denote empirical frequencies of particular values for the system variables.

Empirically-based objective functions invite the use of probability density estimation methods, and hence machine learning methods, to go from database frequencies to probabilities. Further, if empirical probabilities are used to define pseudo-energy terms, there is no reason to use only atomic-level descriptions — observable patterns obtain at other, coarser levels of description, too. We therefore have access to the efficiencies that can derive from multi-resolution search and from search through comparatively smaller sets of larger objects than atomic coordinates and bond-angles.

Sippl puts it succinctly in some pioneering papers on empirical potential functions (Sippl, 1990a; Sippl, 1993):

“The energy of interaction of atomic groups depends on their separation. The shapes of the potentials are obtained by calculating the distribution of distances between interacting atoms from a data base of of protein structures available from the Brookhaven protein data bank. The observed frequencies are transformed with the help of Boltzmann’s law to yield the
potentials of mean force of the interactions as a function of distance. These potentials contain the knowledge acquired from the data base."

Of course, the collection and use of empirical relationships in potential function need not stop at estimations of such things as the probability of interaction at 2 angstroms of an Alanine with a tryptophan 30 residues downstream. One can estimate the probability of an interaction between any "small" amino acid and any "large" amino acid at a particular distance, or indeed an interaction between an entire α-helix and an entire β-strand in a particular configuration. Such attempts to model the distribution of structures, sequences, and structures given sequences, take one into the realm of machine learning.

The intuitive explanation of empirical methods is that the most likely patterns of structure, given a sequence, correspond roughly to the the most commonly observed patterns in a database of real proteins: hence the particular subsequence-substructure associations presumably correspond to biophysically plausible, if not optimal, combinations. In fact, it has been commonly observed that when one plots certain structural parameters of real proteins, as in a Ramachandran plot of the Φ and Ψ backbone dihedral angles, the resulting image of frequency distributions looks nearly identical to a plot of $-Energy$. In other words, the most common configurations are the most energetically favorable, and hence perhaps the most likely to be the native configurations. Despite the apparent intractability of the structure optimization search problem, we have reason to believe that the actual distribution of graph structures within our domain is far from uniform. Thus we can observe the regularities and exploit them to form heuristic rules: \((sequence) \Rightarrow (structure)\). Such rules can, in principle, be used to side-step large amounts of search. Of course, it is not quite this simple, as discussed in the next sections.

### 2.7.1 What is Machine Learning?

Machine learning might be described as the set of theories and methods concerned with automated knowledge acquisition, or with automated transformations of data structures into more useful representations, or with the automated discovery of implicit but recurrent patterns in a set of data. A very general definition might be: Machine learning research is concerned with the search for models of phenomena. In current practice, the field of machine learning, construed broadly, includes subfields which focus on statistical inference, approximation and regularization theory, artificial neural networks, classification, and automated induction, abstraction, and analogy. Roughly, one might divide all the approaches into either the numerical/statistical or
the symbolic/logical camps, though the boundaries are fuzzy, there is much work on hybrid approaches, and important issues cut across both sides of the divide.

2.7.2 Supervised and Unsupervised Learning

Neural network learning in particular, and most ML paradigms in general, can be divided roughly into supervised and unsupervised learning. In supervised learning, the learning system adjusts its internal state in order to decrease the error in associating input examples with assigned target values. The target values may be viewed as interaction with a "teacher". In supervised classification, for example, a system learns to assign each member of a training set of examples to a pre-designated class. In unsupervised learning, by contrast, one defines some task-independent measure of the quality of representation of the input data, and then adjusts the representation in order to optimize this quality measure.

One can often view a supervised learning task as the problem of estimating parameters of some model of the distribution $p(y|x)$ for inputs $x$ and outputs $y$. Similarly, unsupervised learning can sometimes be usefully formulated as the estimation of parameters for a model of $p(y)$. In our domain of interest, $x$ and $y$ might correspond to (sub-)sequences and (sub-)structures, respectively.

2.7.3 ML as Motif Discovery

A major focus of machine learning in the computational molecular biology domain is the automatic discovery of motifs. A motif is an abstraction of a set of observed patterns in the data, and consists of a set of features, perhaps associated with a set of probabilities or frequencies over the features. That is, a motif is a model of some phenomenon or relationship. A feature is a variable associated with a particular value. The variable often represents a component of a larger structure. For example, a simple motif describing a particular kind of restriction site in DNA sequences might be expressed as:

$$(\text{NumPositions} = 6, \text{Pos1} = \text{G}, \text{Pos2} = \text{A}, \text{Pos3} = \text{C}, \text{Pos4} = \text{G}, \text{Pos5} = \text{T}, \text{Pos6} = \text{C}).$$

Here, the variables, components of a larger sequence of variables, are Pos1, Pos2, etc., and they take on values from the alphabet of DNA base symbols. No explicit probabilities are attached, and so the probabilities are assumed to be 1 — those variables must always have those
values. in the described motif. Well-known motifs in computational molecular biology include
the "zinc-fingers" (Vallee, Coleman and Auld, 1991), and bacterial promoters (Nakata, Kanehisa
and Maizel, 1988), as well as sequence motifs found for particular protein families (Baldi and
Chauvin, 1994).

The utility of motif discovery in the molecular biology domain is predicated on two related
ideas. The first is that form follows function (and vice versa), and the second is that molecular
evolution is selective of function. Hence the conservation of a feature throughout a large set of
evolutionarily-related protein sequences is often assumed to bespeak important structural and
functional constraints. Such conserved features may include amino acid identity ("W appears at
position 3 in 90% of these sequences") or physico-chemical properties ("a large-side-chain amino
acid appears at position 3 in 100% of these sequences") or multi-position relationships ("the
amino acids at positions 41 and 108 always have opposite electric charges").

Motifs can be described in terms of their intension, their "concept" or representation, or in
terms of their extension, i.e., the set of objects that fit the motif. The essence of motif discovery
and rule induction in ML is exactly the generation of intensions from extensions. Motifs may be
combined, using the AND, OR, and NOT operations (equivalently, the set intersection, union,
and complementation operations) into more complex motifs. A motif might be implemented
explicitly in some logical language, as in the example above, or may be implemented implicitly,
as in a set of network connection weights or other statistical parameters. It is often possible in
principle to translate one format into another, though in practice this can be difficult (Denker
et al., 1987; Towell and Shavlik, 1991; Maclin and Shavlik, 1992).

In the protein analysis domain, motifs may describe patterns within protein sequences,
within protein 3D structures, or a combination thereof (Conklin, 1995).

This thesis takes a statistical modelling view of learning tasks, one in which the major
computational effort in learning is the estimation or approximation of particular probability
distributions over objects and sets. Consider, for example, the standard protein secondary
structure prediction problem task. Here the learning subtask might be viewed as the estimation
of key parameters $\theta_\alpha, \theta_\beta, \theta_{coil}$ in a parameterized model of the traditional secondary structure
classes. The performance subtask is then the calculation, for some local window of adjacent
residues $s_{test}$ of a protein, $\max(p(CLASS = \alpha|\theta_\alpha, s_{test}), p(CLASS = \beta|\theta_\beta, s_{test}), p(CLASS =
coil|\theta_{coil}, s_{test}))$. The parameter vectors $\theta$ might represent the connection weights in a neural
network, for example, or the means, variances, and mixing proportions of multidimensional
Gaussian distributions in a mixture model. The representation of a motif describing "α-helical regions" then would be the parameter vector $\theta_\alpha$, because its components can be used to generate the particular ranges of values of input variables that characterize the $\alpha$-classified subsequences as against the $\beta$- and coil-classified ones.

Motif-discovery is the focus of this thesis, and motif-discovery is essentially an unsupervised learning task. The goal in motif discovery is the development of a "better representation" of the input data, a representation with more explanatory power, that organizes the data and simplifies it. Usually this is done through estimation of the parameters of a model of the probability density over the data, where the model employs latent variables. These latent variables, and their most likely or best-fitting parameter values, provide the basis for the new and better representation of the data.

While the new methods presented in the thesis are unsupervised methods, they are to be used and analyzed within the larger task of protein structure prediction. PSP, if it is to be approached by ML methods at all, is essentially a supervised learning task — an attempt to assign structure targets to sequence inputs. This learning task is too difficult for simple, one-step approaches. The next few sections present one view on how to decompose the huge task into tractable subtasks, and on how unsupervised motif-discovery procedures may play a major role.

2.8 The Need for Locality, Abstraction, and Selectivity

We know how to conduct machine learning, we have sequences with known 3D structures, so why can't we just train a module to map protein sequences to 3D structures? The problem is one of combinatorics, complexity and statistical support — the objects are too big and occur too infrequently.

Let us examine this in some depth. Suppose we have a set $\mathcal{S} = \{S_1, S_2, \ldots, S_M\}$ of sequences and a set $\mathcal{T} = \{T_1, T_2, \ldots, T_M\}$ of tertiary structures.

Each $S_i = s_{i1}s_{i2}\ldots s_{in_i}$ and each $T_i = t_{i1}t_{i2}\ldots t_{in_i}$, where the $s_{ij}$ are amino acid symbols, say, and the $t_{ij}$ are pairs of $\phi, \psi$ angles.

We would like to train a system with this data, such that when we input a new sequence $S_{new}$ to the system, we obtain a predicted 3D structure $T_{new}$ as output. We might view the
prediction aspect of the system as an implementation of Bayes’ Rule:

\[ p(T_{new}|S_{new}) = \frac{p(T_{new})p(S_{new}|T_{new})}{p(S_{new})} \]

But whence come these estimates on the right-hand side? If the protein in question is, for example, Phospholipase A2, isolated from the venom of the Western Diamondback Rattlesnake, then \( S_{new} = \)

\[ SLVQFETLIMKIAGRSGLLLWYSAYGCYCGWGGLPQDATDR \]
\[ CCFVHDCCYGBKATDCNPKTVSYTYSEENGEIIICGGDDPCGTQ \]
\[ ICECDKAAAICFRDNIPSYDNKLYWLFPPKDRCREEPEC. \]

and one is faced with the calculation of

\[ p(Pos1 = S, Pos2 = L, \ldots, Pos98 = C). \]

This particular sequence only occurs once in the entire database. Thus, direct estimations of such probabilities, from the corresponding frequencies, are impossible — there is not sufficient statistical support. The same holds true for the probabilities of tertiary structures, when represented in their entirety and at such an exact level of detail.

Note, too, how this paucity of data is exacerbated by the size of the necessary representations — the number of degrees of freedom in the learning task. In considering tertiary structure, one really must consider at least pairwise interactions between elements distant from each other in the sequence, as in a distance matrix representation. The size of a distance matrix grows as \( O(N^2) \) where \( N \) is sequence length (though, there are many fewer than \( \frac{N(N-1)}{2} \) actual degrees of freedom). A naive attempt to learn all the values in a distance matrix therefore requires the optimization of nearly 5000 parameters for a sequence of length 100. Proteins can be much larger than 100 residues in length.

Clearly, any formulation of the learning problem in PSP that is so naively direct is doomed to failure. It is necessary to find representations of sequences and structures that circumvent the problems of size and statistical support. To achieve this goal, it is possible to identify three basic strategies employed in the design of motifs and of motif-discovery methods:
- **Abstraction** is the modification of a motif so that the new version has a larger extension — the motif fits more objects, and is therefore more general. A more abstract motif "occurs" more frequently in the data, and hence \( p(motif) \) is higher, and its estimation is less likely to be troublesome. One simple kind of abstraction is the clustering of a set of similar objects into classes, followed by the replacement of each of the objects by its class label, class centroid or exemplar. This operation can be combined with a shift to a larger granularity of representation. For example, one can represent interactions between "chunks" of sequence instead of between individual amino acid residues. A generic "α-helix" occurs much more frequently than do any of its many particular instantiations. Another common use of abstraction is the use of physico-chemical property codes (Nakai, Kidera and Kanehisa, 1988) for amino acids — the feature "large side-chain", corresponding to the set of amino acids \( \{W, Y, F\} \), occurs more frequently than do any of \( W, Y, \) or \( F \).

- **Locality** refers to the restriction of a motif to smaller regions of the protein, either in terms of sequence distance or 3D structural distance. Smaller pieces of large objects occur more frequently than the large pieces, e.g., \( p(ABCD) \geq p(ABCDE) \). Empirical contact potential terms based on small interaction radii have fewer parameters to be learned than do potentials based on larger radii. Note that locality in time can be as useful as spatial locality: molecular dynamics simulations avoid exhaustive search of conformation space by following Markov-chain transitions, which define a local topology on the possible intermediate states of a folding process. Hidden Markov Models (HMMs) in sequence analysis also, as their name implies, rely on the use of Markovian locality constraints.

- **Selectivity** is employed when one chooses to represent only a subset of possible features, preferring features which meet some particular information-theoretic or domain-specific criteria for "interestingness". For example, one might choose to represent only the tiny subset of pairwise residue-residue interactions that correspond to significant correlations discovered in a set of aligned and related sequences (as discussed in Chapter Four).

In the next subsection, brief mention is made of some of the dominant areas of work in machine learning in protein structure prediction. The emphasis is on the application of the three principles stated above to the development and use of particular kinds of protein motifs. This skeletal survey is followed by the final subsection of this introductory chapter, in which the kinds of motifs developed in the remainder of the thesis are discussed.
2.8.1 Recent Work on ML in PSP

Perhaps the most straightforward approach to predicting the structure of a sequence $s_{new}$ is to try to find a similar sequence $s_{known}$ with known structure, and then to use that known structure as the prediction. This approach, with some extensions and embellishments, is known as homology modelling. Homology modelling is essentially the application of nearest-neighbor classification to PSP, with the evolutionary distance measure (a.k.a., string-edit distance, weighted Levenshtein distance) as the metric (Taylor. 1988: Nishikawa. 1986: Sankoff and Kruskal. 1983). One might say here that the motif is the matching sequence itself. To the extent that the metric and/or the neighborhood function is adaptive, one can say that ML is being applied. To the extent that the mapping from known sequence to known structure is modular — that is, localized — the methodology shades into other methodologies discussed below.

The recognition that sequence similarity can provide at least a first approximation to structural similarity has fueled many efforts in sequence modelling. Because the member sequences of a protein family, by definition, share a common structure, at least in coarse outline, one can abstract the idea of sequence-matching into motif-matching, and thereby extend the homology modelling method. The earliest motifs used in this way were simple consensus sequences. A consensus sequence is computed from a simplistic voting scheme — winner take all — among the sequence elements at each position. This may be embellished by using wild-card symbols, etc. Below is shown a simple consensus sequence motif (on the right) for a toy example family of protein sequences (on the left).

\[
\begin{align*}
\text{Example 2.1} & \\
\text{col1} & \text{col2} & \text{col3} & \text{col4} & \rightarrow \text{col1} & \text{col2} & \text{col3} & \text{col4} \\
A & C & C & F & \Rightarrow & A & C & * & F \\
A & C & C & F \\
A & C & M & F \\
V & C & L & G \\
V & C & L & G \\
A & C & M & F \\
\end{align*}
\]

A real improvement comes from allowing insertions and deletions in aligning the modelled sequences, as in the dot matrix methods and their successors (Staden, 1982; Gribskov. McLachlan and Eisenberg, 1987a). Another sensible extension is to use statistical analysis to maintain an ordered, weighted list of most frequent elements at each position in the class motif, and to weight
the positions in accord with information-theoretic criteria for "importance". (For example, a position is more important if highly conserved, i.e., if the distribution of elements found in it across all of the sequences has low Shannon entropy. A position may also be tagged as more important if this probability distribution mentioned above is very different, as measured in Kullback divergence, from that of a set of sequences in a particular different class or family (Schneider and Stephens, 1990).)

The extension of consensus sequence methods in these ways suggests the use of general and sophisticated computational linguistics approaches to sequence family modelling. Searls (Searls, 1988) has advocated and initiated such a research program. ML in this framework becomes grammar induction, for some appropriate class of deterministic or stochastic grammars. Recent successful work in this area includes the application of Hidden Markov Models (HMMs) (Haussler et al., 1992; Baldi and Chauvin, 1994; Krogh et al., 1994), long a standard in the automatic speech recognition domain.

An obvious tradeoff is encountered in computational linguistic and adaptive approaches to modelling: more powerful languages are needed to model some kinds of sequences and structures, but more powerful languages are harder to learn. The set of possible $\beta$-sheets can be shown to be beyond the representational power of context-free grammars (Searls, 1988; Jiménez-Montaño, 1984) (and therefore far beyond the capabilities of finite state machines and standard HMMs). The error-free induction of significantly more powerful grammars in this domain is intractable. However, these limitations do not bar further research or progress, because error-free induction is not required for such programs to be useful.

The methodologies described above all aim at the development of sequence motifs. One can also of course define structure motifs, and this is an area of much research activity. Most of the work on the definition and discovery of new structure motifs concerns sequence-localized substructures, all of which are in Chapter 3 grouped under the broadened definition of "secondary structure". Numerous references to and comparisons among the different proposed motifs can be found in that chapter. Most machine learning (particularly neural network) projects in PSP are devoted to learning how to predict the existing, traditional secondary structure motifs ($\alpha$, $\beta$, coil), rather than the discovery of new ones. This, too, is discussed in Chapter 3.

In closing this section, three important approaches that do not focus on localized motifs are noted. First, Nussinov, Wolfson and others (Nussinov and Wolfson, 1991) use a geometric hashing scheme adapted from machine vision research (Lamdan and Wolfson, 1988) to define
large-scale, non-local structure motifs. Second, some researchers (Taylor and Thornton, 1983; Taylor and Thornton, 1984; Orengo, Jones and Thornton, 1994) focus on the definition and matching of motifs at an hypothesized super-secondary structure level. Third and finally, there is sequence threading. Sequence threading is an attempt to understand protein folding by working on the inverse folding problem: Given a particular tertiary structural motif, find the set of all possible sequences \( s \) that will assume the described conformation. Machine learning methods can be employed at both ends of this process. First, a motif is created by the automated abstraction over a set of known sequences and/or structures, and then a genetic algorithm, for example, can be used to generate and test candidate sequences against the motif. The motif is meant to encode the crucial features of a protein's biochemical/physical environment (Bowie et al., 1990a; Bowie et al., 1990b; Bowie, Luethy and Eisenberg, 1991a) and may be designed by hand instead of learned. It has been shown that the threading problem, intuitively a much simpler problem than PSP itself, is also \( NP\text{-Hard} \) (Lathrop, 1994).

2.9 Putting the Pieces Together: New Motifs and Discovery Procedures within an Overall PSP Methodology

The approach to protein tertiary structure prediction proposed in this thesis unites the following ideas: unsupervised learning (statistical estimation) employed in discovering sequence and structure motifs; supervised learning employed judiciously in mapping motifs at one level to motifs at another level (as in secondary structure prediction); and the combination of predictions and partial predictions from several modules into the prediction of tertiary structure for a new sequence. All of this fits within the broader "substructure condensation" paradigm of protein structure prediction (though it would probably be necessary to combine this with homology-modelling and energy-minimization into a broader system that would integrate all sources of knowledge bearing on PSP).

This section discusses how the particular motif-discovery methods studied in the following chapters fit into such a "big picture" view of PSP. The use of abstraction, localization, and selection in motifs is at the heart of the overall approach.

Chapter 3 discusses the development of local structural motifs that correspond to patterns over fragments of 13 contiguous amino acid residues. The standard secondary structure classes implicitly define one such set of motifs — \( \alpha \)-helix, \( \beta \)-sheet, and some number of others — but
these motifs and their corresponding classes are not the only, nor necessarily the best, possible choices.

How would one use a newly-discovered set of secondary structure classes? If one has a set of such classes, and the classes are highly predictable from amino acid sequence, then the intractable global sequence-to-tertiary learning problem can perhaps be decomposed into two or more stages, wherein the problems of combinatorics and statistical support become manageable. The key is the use of localization and abstraction in re-representing a sequence of amino acids as a much shorter sequence of secondary structure labels.

The method is presented below in outline (also see Figure 2.4). Steps 1-4 comprise the task of discovery, or definition of the secondary structure classes/motifs. Steps 5-8 comprise the use of secondary structure in prediction of tertiary structure. This process is analogous to the discovery and use of “phonemes” in speech recognition or of “morphemes” in higher-level natural language understanding systems.

1. Obtain data sets \( \tilde{S}, \tilde{T} \) of sequences and structures, respectively.

2. Choose \( N_w \), the size of the window — the locality of first-stage prediction.

3. Create data sets \( A, Z \) and \( X = A \times Z \) by cutting every \( S_q \in \tilde{S} \) and \( T_q \in \tilde{T} \) into \( N_w \)-sized fragments. (For simplicity, assume for now that each \( S_q \) and each \( T_q \) is cut into non-overlapping sequence or structure fragments that, when concatenated, span the entire sequence or structure.)

4. Perform clustering\(^5\) within \( Z \), assigning each \( z_i \in Z \) to some class \( C_j \); and choose a symbol \( y_j \) for each such \( C_j \). Thus is a new alphabet defined. \( Y = \{y_1, y_2, \ldots, y_m\} \).

5. Train an ML module to implement or approximate the mapping \( F_{sec} : A \rightarrow Y \) that assigns the correct \( y_j \) for each \( z_i \) to the corresponding \( a_i \). The trained module performs secondary structure prediction.

6. For each sequence \( S \in \tilde{S} \), create the new sequence \( S^Y \) of \( Y \)-symbols such that each sequence fragment \( \tilde{a}_i \) making up \( S \) is replaced by its correct \( y_j \). Call the set of new, shorter sequences \( \tilde{Y} \).

\(^5\)We use the term clustering to mean, very broadly, the assignment of every member of a set into some subset, based upon some notion of similarity or distance. The subsets — clusters — may or may not be disjoint, may or may not be fuzzy. The reader should not interpret the usage as implying any particular metric or any particular clustering method; in particular, the usage is meant to include the use of probability-based latent-class modelling. Thus, except in the sections detailing particular methods, “cluster” and “class” and “latent class” are synonymous.
7. For each structure $T \in \mathcal{T}$, create a new representation $T^Y$ defined in terms of structural/spatial relationships among $y_j$'s. (That is, construct a coarse-grained representation of an arrangement of structure fragments rather than an arrangement of individual amino acid residues.) This set of new structure representations is $\mathcal{T}^Y$.

8. Design a module to implement the mapping $F_{\text{ter}} : \mathcal{Y} \to \mathcal{T}^Y$. This may be done using some combination of pattern-matching, distance-geometry and packing analysis, and energy-minimization and molecular dynamics simulation.

Two Stages of Tertiary Structure Prediction

![Diagram of two stages of tertiary structure prediction]

Stage 1: Predict Secondary from Local Sequence

Stage 2: Predict Tertiary from Secondary

Figure 2.4: This figure depicts a two-stage approach to protein tertiary structure prediction. First, the protein sequence is segmented into short subsequences, and the secondary structure labels are predicted for each of the segments. Then the sequence (or other configuration) of secondary structure labels is used to predict an approximate, coarse-grained representation of tertiary structure. The box inset at upper-right reminds the reader that the secondary classes are defined, manually or automatically, to be compatible with both sequence and tertiary structure. In the methods presented in this thesis, the definitions are produced automatically, in unsupervised learning (probability density estimation) procedures.

How does this two-stage reformulation help us? Instead of trying to estimate such quanti-
ties as \(p(T_{new}|S_{new})\) directly, where both the sequence and structure have length \(N\). One is trying to estimate \(p(y_{j1}|a_1).p(y_{j2}|a_2)\ldots .p(y_{jn}|a_n)\) in Stage 1 of learning, and \(p(T^V|y_{j1}, y_{j2},\ldots , y_{jn})\) in Stage 2. The objects of interest in the two stages are of sizes \(n\) and \(N_w\), respectively, where \(n \ll N, N_w \ll N\). The new objects are smaller, occur more frequently, and their probabilities are easier to estimate from finite databases.

The choice of the secondary structure classes, upon which is based the representation language of the crucial intermediate level of description between sequence and tertiary structure, largely determines the success or failure of both main stages and thus of the entire operation. The fragment-size \(N_w\), the degree of overlap between fragments in sequence- and structure-representation, the number of classes to be used, and the means of incorporating sequence and structure information in the clustering step, all bear heavily on the resulting class structure. Chapter 3 focuses primarily on the last issue, the combination of information from both \(A\) and \(Z\) in selecting secondary structure classes.

The work described in the other main research chapter, Chapter Four, is aimed at bringing another kind of information to bear on tertiary structure prediction. This information is embedded in the coincidence of particular amino acids over particular pairs or \(k\)-tuples of residues (positions) separated by (perhaps long) distances within a sequence, observed over a set of aligned sequences. An efficient method for detecting and modelling such correlations can be useful in at least two distinct ways.

First, there is information to be extracted at the level of individual sequences, in the form of joint symbol frequencies. It is well-known that an abnormally high observed frequency of a particular single-position pattern (e.g., “G occurs at residue number 3 in 98% of these sequences”) can reveal an important physico-chemical constraint on secondary or tertiary structure. This is also true of surprisingly-frequent joint symbol occurrences (e.g., “G at position 3, L at position 5, and M at position 87 occurs much more often than would be predicted by the individual marginal frequencies”). Such long-distance co-occurrences might be especially indicative of tertiary constraints, because the designated positions may be nearby each other in the 3D structure to which all of the modelled sequences correspond. (This detection of “suspicious coincidences”, as when \(p(A, B) \gg p(A)p(B)\), is at the heart of pattern recognition and learning, as noted long ago by Barlow (1972), and others.)

Second, there is information to be extracted at the “next level up”, of statistical relationships between the positions (columns in an alignment of homologous sequences). If the existence
of frequently occurring joint symbol \( k \)-tuples can be used to infer 3D structural interactions, such an inference is even better supported by certain information-theoretic relationships between positions (columns) over a set of \textit{many different} joint symbol occurrences. This is because such symbolic relationships can signify evolutionarily conserved physical or structural relationships between different parts of the protein chain. (Please see Figure 2.5.) The observation of high values of mutual information (Shannon and Weaver, 1964) and other correlation measures between columns has been used successfully to predict 3D structural interactions in RNA (Gutell et al., 1992) and in HIV proteins (Korber et al., 1993). While these previously reported efforts have focused on \textit{pairwise} residue-residue interactions, the work described in Chapter Four, along with that of David MacKay (MacKay, 1994), aims at the detection of \( k \)-ary interactions for \( 2 \leq k \leq N \).

In closing this chapter, let us review the different ways in which local predictions can help narrow the search for the best global structure predictions.

First, there are distance geometry constraints. Secondary structure prediction, and the discovery of \( k \)-ary long-distance interactions, give evidence for \textit{presumed contacts}, of the form \textit{contact}(i, j) for the \( i \)th and \( j \)th amino acid residues in a protein. Using distance geometry theory (Chan and Dill, 1990; Crippen and Havel, 1988; Ciani and Goel, 1985), one can derive a set of \textit{inferred contacts}. One can also derive sets of \textit{inferred blocks}, contacts that are forbidden by a given set of presumed or inferred contacts. Essentially, given a model of a polymer chain constrained to exist within a fixed volume, the assumption that two particular pieces are brought into contact implies that some other pieces are also brought into proximity and that still other pieces are moved further apart. Indeed, Chan and Dill (1990) conclude that "considerable amounts of internal architecture (helices and parallel and anti-parallel sheets) are predicted to arise in compact polymers due simply to steric restrictions. This appears to account for why there is so much internal organization in globular proteins."

Second, as discussed throughout the previous sections, one can infer and exploit empirical relationships between local and global configurations. Local stretches of sequence, or selected non-local pairs of residues, can be found to occur, with some high probability, in particular global configurations. Heuristic rules, in whatever form, can be used to avoid large parts of conformation space. The inference of particular models of cooperativity in folding is a special case: knowledge of "rules" such as \( p(\text{contact}(i, j) | \text{contact}(i + 1, j - 1)) > p(\text{contact}(i, j)) \) can help significantly.
Correlated Positions Suggests Conserved Structural Constraints

Figure 2.5: This figure illustrates how correlations between positions — columns — within an aligned set of $M$ protein sequences might represent an evolutionarily-conserved structural constraint. In this toy example, the conserved relationship between a large-side-chain amino acid (tryptophan ($W$) or tyrosine ($Y$)) at position $i$ and a small-side-chain amino acid glycine ($G$) or Alanine ($A$)) at position $j$ might represent a packing constraint, as the two residues (positions) are in contact in the protein’s 3D structure.
The larger view taken within this thesis is that all of these sources of information ought to be integrated within a multiple-stage, multiple-view PSP system. (Please see Figure 2.6.) Sibbald (1995) has taken steps towards quantifying the computational advantage gained from disparate sources of partial prediction knowledge. He found, using a lattice model of structure and an expert system model of constraint-satisfaction, that the number of possible conformations can be severely limited with relatively few constraints. Of course, many different architectures and decision methods may be used for such a system, and a very sophisticated version would also integrate energy-minimization and molecular dynamics simulations into the prediction process, as has been done (Major et al., 1991) in RNA structure prediction and modelling. Such integrated methods can be extended beyond structure prediction per se to general molecular modelling tools that combine prediction with structure determination, as in the crystallographic interpretation approach taken within the Molecular Scene Analysis project (Baxter et al., 1996).
Some Sources of Information in Tertiary Structure Prediction

Figure 2.6: Above is a schematic drawing suggesting the integration of multiple knowledge sources in PSP. The sources include predictions from secondary structure packing (top), homology-based prediction (middle), and important features detected through analysis of large numbers of related sequences (bottom). In a sophisticated system, the arrows ought to be two-headed — information might be generated and used in both bottom-up and top-down manners.
Chapter 3

Automated Discovery of Predictable Secondary-Structure Classes

3.1 Introduction

This chapter describes a set of related methods for the automated discovery of novel protein secondary-structure classes \(^1\). The research project outlined herein began as an attempt to define secondary structure classes — and hence local structure motifs — that are more predictable from amino acid sequence than the standard classes \(\alpha\)-helix, \(\beta\)-sheet, and so-called random coil. The research continues as an effort to define criteria for "good" secondary structure classes and motifs and to use unsupervised learning methods to develop classifications that meet these criteria. Predictability remains an important criterion.

The recognition that a class structure on a set of objects implicitly defines a set of motifs, and vice versa, ties this work into the large body of work on protein structure motifs. A major distinguishing feature of the work presented in this chapter is the emphasis on sequence-structure predictability, and the simultaneous use of both structure fragments and corresponding sequence fragments in the motif-discovery procedures.

The structure of this chapter is as follows: First, the next section presents some background on standard secondary-structure prediction and its limitations. Then, three main motif-discovery procedures.

\(^{1}\)Some of the work presented in this chapter has been published earlier, in a conference (Lapedes and Steeg, 1993) and a journal (Lapedes, Steeg and Farber, 1995). That work, which corresponds to the Method 1 section of this chapter, was joint work with Alan Lapedes and Robert Farber of Los Alamos National Laboratories and the Santa Fe Institute. Dr. Lapedes organized the collaborative effort and Drs. Lapedes and Farber obtained the original input data and performed the simulation experiments described as "Experiment 1" in the Method 1 section.
methods are presented, each in its own section. Each method is associated with an unsupervised learning procedure defined by the minimization of a particular objective function. The methods are presented in a particular order that reveals a logical progression in their derivations. The first method — maximization of correlation between two feed-forward networks — can be derived from a basic definition of predictability combined with simple constraints against trivial classifications. The second method — joint sequence-structure density modelling — represents an attempt to trade off some predictability for more informative structure classes. A plausible definition of "informative", in terms of Minimum Description Length (MDL), is provided in Section 3.4. The third method — a conditional mixture modelling approach — gives up on the goal of imposing class structure on both the sequence and structure spaces, and concentrates only on structure space, where there seems to be more natural class structure. For each method, a series of subsections details, respectively: the intuitive justification; objective-function definition; gradients, update rules, and prediction methods; experiments and results; and possible extensions.

A note: Some of the analysis and most of the results presented in this chapter are based on a two-classes secondary structure model, as previously used in some studies (Kneller, Cohen and Langridge, 1990; Stolorz, Lapedes and Xia, 1991a). This restriction facilitated the design, testing, and comparison of many different algorithms and variants, and permits a clearer and simpler presentation of the main ideas. The extension to multiple classes is not especially difficult, and should be pursued in further experiments in the near future.

3.2 The Success, and Failure, of Secondary-Structure Prediction

The conventional classes of protein secondary structure, α-helix and β-strand, were first introduced in 1951 by Linus Pauling and Robert Corey (1951) on the basis of molecular modeling. The researchers found that two periodic peptide structures, which they called α-helix and β-strand, could be built as detailed molecular models that satisfied experimentally-determined constraints on bond angles and distances. The α-helix secondary structure was later observed in the X-ray diffraction reconstruction of the hemoglobin protein (Perutz, 1951). Examples of β-strand secondary structure may be found in, for example, silk. These two classes of structure are visually quite apparent in modern molecular graphics representations of X-ray diffraction.
models of proteins. As more crystallographically-determined protein structure models became available, their secondary structures were annotated (often, only on the basis of visual inspection) as apparent helices and strands. To remove confusion and conflicting classifications, formal definitions of α-helices, β-strands, and other secondary-structure classes were constructed on the basis of certain characteristics of the local geometry of proteins. For example, a hierarchical definition that is in widespread use today first defines potential hydrogen bonds based on structural coordinates, and subsequently defines α-helices and β-strands as particular patterns of the potential hydrogen bonds. The Kabsch and Sander definitions are precise, algorithmic, and have been implemented in the DSSP program (Kabsch and Sander, 1983a); they serve as the standard for most protein researchers today.

As discussed in the opening chapters, it has long been an outstanding problem of computational molecular biology to be able to predict these classes of secondary structure from the amino acid sequence. A major reason for the interest in secondary-structure prediction is that a successful prediction of secondary structure from amino acid sequence may be used in tertiary-structure prediction algorithms to constrain their search space. This was discussed in Chapter Two with respect to the theories of distance geometry and cooperativity. Empirically, Skolnick's research group (Skolnick and Kolinski, 1991) has found that biasing amino acids towards assuming the correct, measured secondary structure, when coupled to their global tertiary-structure prediction algorithms, greatly increases the agreement of the global tertiary-structure prediction with the experimentally-determined structure. However, their test of the value of knowing the secondary-structure classes used the actual secondary-structure classes as determined from experimental data, and not error-prone algorithmic predictions of secondary-structure classes from amino acid sequence. Their method, like others, is not generally successful if one attempts to use predictions of secondary-structure classes at current levels of accuracy.

3.2.1 Prediction of Traditional Secondary Structure Classes

There have been numerous attempts to predict locally-defined secondary-structure classes using only a local window of sequence information. The prediction methodology ranges from a combination of statistical and rule-based methods (Chou and Fasman, 1978) to neural network methods (Qian and Sejnowski, 1988; Kneller, Cohen and Langridge, 1990; Stolorz, Lapedes and Xia, 1991a; Denker et al., 1987; Maclin and Shavlik, 1992). Figure 3.1 illustrates the Kabsch and Sander program defining secondary-structure classes, depicted as a "black box" on the right,
and also a neural network that attempts to learn the secondary-structure classes from the amino acid sequences, on the left. The Kabsch and Sander “black box” first defines hydrogen-bonding patterns from the structural information, and then uses the hydrogen-bonding patterns to define classes of secondary structure. This picture represents the standard approach to training a neural network to classify secondary structure from amino acid sequence (Stolorz, Lapedes and Xia, 1991a). A local window of structure information obtained from, for example, X-ray diffraction data in the Brookhaven (Abola et al., 1987) database, is input to the right-hand Kabsch and Sander black box. The box outputs the secondary-structure class of the fragment using the Kabsch and Sander definitions. For example, if one were dichotomizing all the windows of structure information into “α-helix” and “not-α-helix”, then the right-hand box will emit a “1” if the fragment is α-helix, and emit a “0” otherwise. The left-hand neural network “sees” the corresponding window of sequence information as input, and attempts to adjust its synaptic weights so that the output neuron of the neural network agrees with the output state of the Kabsch and Sander black box. Hence, if the input sequence adopts an α-helix state according to Kabsch and Sander, then the output neuron of the network should change state to “1”. Conversely, an input sequence fragment not in an α-helix state should cause the state of the output neuron to change to “0”.

How do such methods perform? The datasets, testing methodologies, and measures of predictive accuracy vary, but it is fair to say that nobody has reported better than 70% correct secondary-structure prediction on a testing set of significant size and diversity (Kabsch and Sander, 1983b; Biou et al., 1988; Fasman, 1989). (More details, and a more informative accuracy measure, are presented in later sections.) Evidence thus far indicates that this quality of prediction is not sufficiently useful in tertiary structure prediction. (We note here that better-than-70% accuracy has been obtained with the use of multiple alignments in combination with other standard methods (Rost and Sander, 1993; Rost et al., 1995). This corresponds essentially to a homology-modelling approach to structure prediction, and we distinguish this important subproblem from the more general secondary structure prediction framework in which the training set consists not of protein families but rather of a large set of diverse protein sequences.)

3.2.2 Why Can’t Secondary Structure Predictors Do Better?

In recent years there has been much discussion in the computational molecular biology community on possible explanations for the ubiquitous “70 percent limit”.
Figure 3.1: Conventional neural network training for prediction of conventional secondary-structure classes.

One obvious constraint on secondary structure prediction performance, upon which everyone agrees, is the limitation of the local window approach. As discussed in Chapter 2, one-step learning of a global sequence-to-structure mapping is infeasible: hence locality must be employed. However, examination of the protein folding process indicates that non-local forces and interactions play a role in determination of local structure: hence purely-local prediction cannot be perfect. As yet there has been no authoritative answer, theoretical or empirical, that explains exactly how much of the accuracy gap is due to this omission of global forces.

A second hypothesis about the predictive accuracy gap posits that the input representations used in standard secondary structure prediction are inadequate for capturing crucial information in the amino acid sequences. It is certainly true that the representation of inputs and outputs is crucial to the performance of any learning system. However, experiments reported in some of the neural network structure prediction studies (Qian and Sejnowski, 1988; Kneller, Cohen and Langridge, 1990), and especially that of the Los Alamos group (Stolorz, Lapedes and Xia, 1991a), provide strong evidence against this second hypothesis. First, a large number of diverse and clever representation methods have been tried — from physico-chemical property representations of amino acids (Nakai, Kidera and Kanehisa, 1988) to representations...
of pre-computed correlations between residues (Kneller, Cohen and Langridge, 1990). Second, it was demonstrated (Stolorz, Lapedes and Xia. 1991a) that simple one-shot Bayesian learning, based on the very unrealistic simplifying assumption of statistical independence between adjacent residues (in other words, almost the least-clever representation possible) performed almost as well as much more sophisticated representations. These results suggest strongly that researchers may have hit a fundamental "wall" in the pursuit of ever better prediction of traditional secondary structure classes with local methods.

There remains another possible explanation for the limited success of structure prediction. It may be that the traditional classes themselves are the problem. Although the α-helix, β-sheet, and coil classes are historically important and visually apparent, they might not be optimal for prediction. There may be slight variations of the Kabsch and Sander definitions, or perhaps classes that represent strikingly different ways of partitioning the set of secondary-structure-sized fragments, that can be better predicted from amino acid sequence.

This last hypothesis is the basis for the three methods presented in this chapter. In each of the following methods, the idea is to build in predictability from the very beginning of the classification-discovery (motif-discovery) process.

3.2.3 Measuring Predictive Accuracy — Matthews Correlation

It is important to note here that the "percentage correct" measure of predictive accuracy for secondary structure prediction (or other classification tasks, for that matter) has a serious flaw. The measure fails to account for the relative sizes of the m classes into which a dataset is divided, and thus can be very misleading. For example, if a dataset were composed of 90% "random coil" segments, then a trivial predictor module that output the answer coil on every input would achieve 90% predictive accuracy. What is required is a measure that penalizes over- and under-prediction of each class. i.e., false positives and false negatives, differently according to the relative size of the class. The Matthews ("four-point") discrete correlation measure (Matthews, 1975) does this:

\[ M_k = \frac{p_k n_k - u_k o_k}{\sqrt{(n_k + u_k)(n_k + o_k)(p_k + u_k)(p_k + o_k)}} \]

where \( p_k \) ("predictions") is the number of examples where the left-hand network and right-hand network both predict class \( k \), \( n_k \) ("non-predictions") is the number of examples where neither network predicts \( k \), \( u_k \) ("under-predictions") counts the examples where the left network predicts
and the right network does not. and \( o_k \) ("over-predictions") counts the reverse.

Note that if only two classes are used, then \( M_1(\text{source}_a, \text{source}_b) = M_2(\text{source}_a, \text{source}_b) \), and so the subscript designating the class may be dropped: \( M(\text{source}_a, \text{source}_b) \).

The Matthews measure has become the standard in recent years in the protein structure literature.

### 3.3 Method 1: Correlated Feed-Forward Networks

#### 3.3.1 Intuition and Background

The key idea in this section is illustrated in Figure 3.2. In it, the right-hand black box implementing the Kabsch and Sander rules (from Figure 3.1) is replaced by a second neural network. This right-hand neural network therefore sees a window of structural information, while the left-hand neural network sees the corresponding window of sequence information. Note that the right-hand neural network can implement extremely general definitions of secondary structure. For example, if the weights in the right-hand network are set to arbitrary values, then the right-hand network will correspondingly produce an arbitrary classification of the structures that are input to it. On the other hand, one could train the weights of the right-hand network to perform structure classification according to, say, the Kabsch and Sander rules.

To define new rules one merely changes the synaptic weights in the right-hand network. Arbitrary synaptic weights would define arbitrary rules, and there would be little chance that these new classes would be either predictable or meaningful.

In order to develop predictable classes, the rules embedded in the right-hand network's weights must be made to agree with the rules embedded in the left-hand network's weights, thereby ensuring agreement between the classifications being defined and the classifications being predicted.

To achieve this, both networks are trained simultaneously, starting from random initial weights in each net, under the sole constraint that the outputs of the two networks agree for each pattern — or for as many patterns as is possible — in the training set. The mathematical implementation of this constraint is described in various versions below. The two networks become mutually self-supervising and implement a type of unsupervised clustering on the data.

This clustering viewpoint is illustrated in Figure 3.3. Assume one is given a training set, considered as a set of points \( x_s = (a_s, z_s) \), as in Chapter 2, where \( a_s \in A \), \( z_s \in Z \), where \( A \) is
Figure 3.2: Use of a correlation-based agreement measure to train coupled neural networks. Such training can produce novel and highly-predictable protein secondary-structure classifications.

A space of short subsequences of amino acids, and where Z is a space of local tertiary-structure parameters (\(\phi, \psi\) angles) corresponding to the amino-acid residues. The left-hand and right-hand networks may be viewed as nonlinearly transforming the spaces \(A\) and \(Z\), mapping points into new spaces \(A'\), \(Z'\), respectively, such that clusters are formed. These clusters are interpreted as secondary-structure classes.

The objective functions, described below, that force “agreement” between outputs of the left and right nets enforce a compatibility constraint between the clusterings in the two different spaces, as well as enforcing other constraints on the sizes and shapes of the clusters. In general, the compatibility goal is achieved to the extent that neighboring points in sequence space \(A'\) are also neighbors in the structure space \(Z'\), and vice versa. That is, for \(\vec{x}_r = (\vec{a}_r, \vec{z}_r)\) and \(\vec{x}_s = (\vec{a}_s, \vec{z}_s)\), if \(\vec{a}_r\) and \(\vec{a}_s\) are in the same cluster, then \(\vec{z}_r\) and \(\vec{z}_s\) should be in the same cluster. and conversely (as shown in Figure 3.3). To the extent that this constraint holds, we say that the structure classification is compatible with the sequence classification, or in other words that the structure-classes are predictable from sequence.
This co-evolution of the two networks is clearly a more difficult computational problem than the conventional approach (Figure 3.1) that employs fixed targets. Each network now chases a moving target during training, and numerical difficulties in the form of local minima occur. These difficulties are surmountable, and new definitions of secondary structure may be found.

3.3.2 The Objective Function

The derivation of plausible objective functions begins with a careful consideration of predictability. Given $X = A \times Z$ as above, one proposes to define classes $C_j^A$, $j = 1 \ldots m_A$ in $A$ and $C_k^Z$, $k = 1 \ldots m_Z$ in $Z$. The objective is to build in a relationship between the $\{C_j^A\}$ and the $\{C_k^Z\}$ that can be used for prediction. The prediction will work as follows: Given $\tilde{a}$, classify it into some $C_j^A$, then use the relationship between the $\{C_j^A\}$ and the $\{C_k^Z\}$ to "guess" a particular $C_k^Z$: that is the secondary structure prediction for the amino acid subsequence $\tilde{a}$. Given this scenario, intuition suggests that to maximize predictability is to minimize the uncertainty in guessing $C_k^Z$ from $C_j^A$: that is, minimize $H(C_k^Z|C_j^A)$. (Here, notation is stretched so that $C_k^Z, C_j^A$ refer to the random variables whose values are indices of the actual classes $C_k^Z, C_j^A$ over the appropriate ranges of $k$ and $j$.) The intuition is formalized and verified in Fano's Inequality.
First, some useful information-theoretic definitions and facts are presented below. These will be used in the derivations and analysis of Methods 2 and 3 as well. The reader is assumed to be familiar with random variables and probability density and mass functions.

**Definition 3.1** Entropy of a discrete random variable: $H(X) = -\sum_{x_i \in X} p(x_i) \log p(x_i)$

**Definition 3.2** Conditional entropy: $H(Y|X) = -\sum_{x_i \in X} \sum_{y_j \in Y} p(y_j|x_i) \log p(y_j|x_i)$

**Definition 3.3** Joint entropy: $H(X,Y) = -\sum_{x_i \in X} \sum_{y_j \in Y} p(x_i,y_j) \log p(x_i,y_j)$

**Definition 3.4** Relative Entropy (Kullback-Liebler Distance, Asymmetric Divergence) between two distributions $p()$ and $q()$: $D(p||q) = \sum_x p(x_i) \log \frac{p(x_i)}{q(x_i)}$

**Definition 3.5** Mutual information between two random variables: $I(X;Y) = H(X) + H(Y) - H(X,Y)$

**Proposition 3.1** The mutual information between two discrete random variables $X$ and $Y$ is equal to the Kullback-Liebler distance between the joint and product distributions over $X$ and $Y$.

**Proof:**

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

$$= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i,y_j) \log p(x_i,y_j) - \sum_{x_i \in X} p(x_i) \log p(x_i) - \sum_{x_i \in X} p(x_i) \log p(x_i)$$

$$= \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i,y_j) \log p(x_i,y_j) - \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i,y_j) \log p(x_i)$$

$$= \sum_{x_i \in X} \sum_{y_j \in Y} \log \left[ \frac{p(x_i,y_j)}{p(x_i)p(y_j)} \right]$$

$$= D(p(X,Y)||p(X)p(Y))$$

Define $g(C^A_j)$ as the function implementing the guessing of the $Z$-class from the $A$-class: say $g(C^A_1) = C^Z_2$ for the guess of $Z$-class 2, for example. Define $P_e = \text{Prob}(g(C^A_j) \neq C^Z_k)$, the probability of error in guessing. Then this error can be shown to grow with $H(C^Z_k|C^A_j)$. 
Proposition 3.2 Fano's Inequality: \( P_e \geq \frac{H(C'_k|C'_j) - 1}{\log m_Z} \). where \( m_Z \) is the number of classes in \( Z \).

Rewriting \( H(C'_Z|C'_j) \) as \( H(C'_Z, C'_j) - H(C'_j) \), one can see that minimizing prediction error entails minimizing the joint entropy of the two classifications while maximizing the entropy of the classification in \( A \), that is, the responsiveness of the variable \( C'_A \) to differences among the \( \tilde{a} \in A \). Intuitively, each of these two subgoals seems sensible, but something is missing. A trivial way to achieve zero prediction error is to define only one class in \( Z \) — then one will never guess wrongly! But a definition of "protein secondary structure" in which there is just one secondary structure type is useless. It is therefore desirable to add another constraint, another term, to the objective: maximize the variation in the \( Z \)-classes, not just the \( A \)-classes. Hence, a reasonable objective is the minimization of

\[
H(C'_Z|C'_j) - H(C'_Z) = H(C'_Z, C'_j) - H(C'_j) - H(C'_Z) = I(C'_Z : C'_j) \tag{3.4}
\]

Thus the maximization of the mutual information between the \( A \)-classification and the \( Z \)-classification aims at achieving a high predictability along with some assurance against trivial secondary structure definitions. Mutual information is one of the three objective functions (agreement measures) investigated within the studies of Method 1. Use of mutual information in this context is related to the IMAX algorithms for unsupervised detection of regularities across spatial or temporal data (Becker, 1992; Becker and Hinton, 1992). The objective function is defined for the coupled sequence and structure networks by

\[
I = \sum_{j,k} p_{jk} \log \frac{p_{jk}}{p_j \cdot p_k} \tag{3.6}
\]

where \( p_{jk} \) is the joint probability of occurrence of the states of the left and right networks. The quantity \( p_j \), is defined as \( p_j = \sum_k p_{jk} \) and the quantity \( p_k \) is defined as \( p_k = \sum_h p_{jk} \). Minimizing \( E = -I \) maximizes \( I \).

While \( I \) has many desirable properties as a measure of agreement between two or more variables (Cover and Thomas, 1991; Abremski, Sirotkin and Lapedes, 1991; Korber et al., 1993; Gutell et al., 1992) preliminary simulations showed that maximizing this formulation of \( I \) over discrete classes, is often prone to poor local maxima. Therefore other objective functions were
sought.

Consideration of the simple case of two classes within each of A and Z. and networks with one real-valued output unit, suggests the use of a correlation measure on the two streams of values output by the nets processing the dataset examples. Two candidate correlation functions are the Pearson correlation and Matthews correlation functions: the experiments reported in this section employed objective functions based on these two measures.

The Pearsonian ("standard" or "product-moment") correlation measure between two sources, LeftO<sub>i</sub> and RightO<sub>i</sub> is:

\[ \rho = \sum_i (LeftO_i - \bar{LeftO})(RightO_i - \bar{RightO}) \]  \hspace{1cm} (3.7)

where \( \bar{LeftO} \) denotes the mean of the left net's outputs over the training set examples \( i \) and respectively for the right net. The expression \( \rho \) is zero if there is no variation, and is maximized if there is simultaneously both individual variation and joint agreement. In our situation it is equally desirable to have the networks maximally anti-correlated as it is for them to be correlated. (Whether the networks choose correlation or anti-correlation is evident from the behavior on the training set.) Hence the minimization of \( E = -\rho^2 \) would ensure that the outputs are maximally correlated (or anti-correlated).

Alternatively, since one ultimately measures predictive performance on the basis of the Mathews correlation coefficient, it is also reasonable to simultaneously train the two networks to maximize this measure. The Mathews coefficient, \( M_k \), for the \( k^{th} \) state or class, which again, is defined as:

\[ M_k = \frac{p_k n_k - u_k o_k}{[(n_k + u_k)(n_k + o_k)(p_k + u_k)(p_k + o_k)]^{1/2}} \]  \hspace{1cm} (3.8)

where \( p_k \) is the number of examples where the left-hand network and right-hand network both predict class \( k \). \( n_k \) is the number of examples where neither network predicts \( k \). \( u_k \) counts the examples where the left network predicts \( k \) and the right network does not. and \( o_k \) counts the reverse. Minimizing \( E = -M_k^2 \) maximizes \( M_k \).

Initial simulation experiments indicated that \( M_k \) and \( \rho \) are both less susceptible to local minima than \( I \). However, these other objective functions suffer the defect that predictability is emphasized at the expense of utility. In other words, they can be maximal for the peculiar situation where a structural class is defined that occurs very rarely in the data. but when
it occurs, it is predicted perfectly by the other network. The utility of this classification is therefore degraded by the fact that the predictable class only occurs rarely. Fortunately, this effect did not cause serious difficulties in the simulations we performed.

These three measures of agreement — mutual information and Pearson and Matthews correlation — are closely related. A variant of the mutual information objective function for real-valued outputs can be shown equivalent to maximization of Pearson correlation (Becker, 1992). Matthews correlation, also known as “four-point correlation”, is equivalent to Pearson correlation when the latter is restricted to binary-valued variables. Moreover, the use of any of these objective functions on pairs of feed-forward neural networks with sigmoidal-activation units can be shown to be a non-linear extension of a standard statistical method, Canonical Correlation Analysis (CCA) (Mardia, Kent and Bibby, 1979). CCA is used to find linear functions $L_1, L_2$ of two multi-dimensional variables, $X$ and $Y$, that maximize the correlation $\rho(L_1(X), L_2(Y))$. Our neural networks and objective functions extend CCA by replacing simple linear combinations with nonlinear transformations (Becker, 1992). Nonlinearity has been shown in many classification and regression problems to provide better results, thereby justifying its extra computational cost. In Section 3.3.4 below there is evidence of the non-linearity’s usefulness on the task at hand.

The experiments reported below used the Matthews and Pearson correlation-based objective functions. Figure 3.4 illustrates the use of either of the two correlation functions with the two networks and the data used in the experiments. ($LeftO$ and $RightO$ are now expressed in terms of their input datasets, as $O^A$ and $O^Z$ respectively.)

### 3.3.3 Learning Rules and Prediction Rules

In the optimization-based approach to unsupervised learning used in this thesis, the learning (or weight-update) rules follow naturally from the choice of objective function and the optimization algorithm. Typically the weight-update rules for connection weights $w$ are derived from $\frac{\partial E}{\partial w}$. Two main methods are reported here, after a larger number of methods were investigated.

Similarly, the prediction rules follow from the choice of architecture of the learning module — in this case feed-forward neural networks — and from the type of class to be used in classification, as well as from the objective function.

In both the Matthews-based and Pearson-based versions of Method 1, two multi-layer feed-forward neural networks, with sigmoidal activations in the processing units, were used.
Maximizing Correlation between $A$, $Z$ Classification Outputs

\[ E = -[\text{Corr}(O^A, O^Z)]^2 \]

\[ O^Z_i = \text{sigmoid} \left( \sum_{h=1}^{3} w_{h,i} u_h \right) \]

\[ u_{3,i} = \text{sigmoid} \left( \sum_{j=1}^{N_w} w_{j,3} z_{j,i} \right) \]

Figure 3.4: Illustration of the coupled feed-forward networks used in Method 1. The pair of nets is trained with a correlation-based objective function to produce correlated classifications. Simultaneously, in the space of short amino-acid subsequences $A$ and the space of short tertiary structure fragments $Z$. For simplicity, the input layers are shown to have $N_w$ units each; however, the actual input dimensions are $20N_w$ for $\bar{a}_i$ (for the unary encoding of each amino acid) and $4N_w$ for $\bar{z}_i$ (for the $(\cos \phi, \sin \phi, \cos \psi, \sin \psi)$ encoding of each residue). In the experiments performed thus far, $N_w = 13$. Each network employs a small layer of hidden units and a single output unit, all with sigmoidal activations.

Historical and mathematical analyses of this kind of neural network may be found in any of several sources (Hertz, Krogh and Palmer, 1986; Hinton, 1989; Lippmann, 1987). The single output unit of each network takes on continuous values in $[0, 1]$ when each of the input examples $\bar{x}_i = (\bar{a}_i, \bar{z}_i)$ are presented to it.

The objective functions may be defined either in terms of the networks' outputs or in terms of frequencies of the possible classification decisions observed over the example set. It is necessary to relate the two, in order to understand the learning task and the performance (prediction) task. There are many known ways to translate output unit activations $O_i$ into classifications, including, for the simple 2-class case under discussion:

1. "soft classes", with $\text{Prob}(\text{example}_i \in \text{CLASS}1) = O_i$, where $O_i \in [0, 1]$;
2. "hard classes", with $\text{example}_i \in \text{CLASS}1$ iff $O_i > 0.5$ ("hard threshold");
3. "hard classes", with example $i \in CLASS1$ iff $O_i > \bar{O}$, where $\bar{O}$ is the mean output over all examples ("floating threshold").

The third option was used in these experiments. No further assumptions were made about the distribution of objects (short subsequences or substructures) within each class. (This is in sharp contrast to the assumptions made for Methods 2 and 3.) The use of the floating threshold within an objective function can tend to enforce equality of class sizes. However, this effect was overcome in our optimizations of our overall objective functions, so that, for example (0.8, 0.2) partitions resulted on some runs.

**Prediction of structural class using sequence network:**

The following prediction rules were used for both the Pearson and Matthews experiments. Two-class secondary-structure prediction on the $p$th example is defined for the (left-hand) sequence network in terms of the network output $O_L$ on example $p$, the mean output over all training examples, and the post-training correlation between the outputs of the two networks, as follows.

Run example $i$ through network:

```
if ($O_i^L < \bar{O}^L$) then class := CLASS1
else class := CLASS2:
    if ($\rho(O_L^L, O_R^R) \geq 0$) then
        if (class = CLASS1) then predict := CLASS1
        else predict := CLASS2:
    else
        if (class = CLASS1) then predict := CLASS2
        else predict := CLASS1:
return predict;
```

Essentially, this says that if a network's output is below average, then the corresponding input is in one class; if it is above average, then the input is in the other class. Then, in making the prediction of the other network's output, use the first network's classification as the prediction, unless the correlation between the networks is negative, in which case predict the opposite.
This can be extended to handle \( k > 2 \) classes or different correlation or “agreement” measures.

**Training of two-net system using \( M_k \)-correlation objective function:**

The Mathews correlation function was designed for use on dichotomous data, and thus the most natural implementation for two networks employs binary decisions — hard classes. The hard classification introduces discontinuities in the overall objective function, making true gradient techniques impossible. Our initial tests of gradient optimization procedures for a smoothed approximation to Mathews showed that local minima was a significant problem. Local minima seemed to be much less of a problem using the gradient-free Powell procedure, although this point was not intensively investigated. Training with the Matthews-based objective function was accomplished with the gradient-free Powell minimization procedure (Press et al., 1988). The Powell algorithm works by performing successive line searches in conjugate directions, and does not require a gradient. It is generally slower, however, than gradient methods, and requires \( O(N^2) \) storage, as compared with \( O(N) \) for conjugate-gradient methods.

**Training of two-net system using \( \rho \)-correlation objective function:**

The Pearson experiments employed gradient-based optimization methods; therefore the learning rules for this method depend on the derivatives of the objective function with respect to the network’s connection weights.

\( E \) is the objective function, which in this case is \( E = -\rho^3 \). \( O_L \) is, as above, the output of the left-hand network. \( w \) is a synaptic weight in the network. We wish to train the network by adapting the weights in order to minimize the value of the objective function, over all of the training examples \( p \).

Training is accomplished by a conjugate-gradient optimization algorithm (Press et al., 1988), where the gradients are computed as follows.

\[
\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \rho} \frac{\partial \rho}{\partial O_L^{(p)}} \frac{\partial O_L^{(p)}}{\partial w}
\]

\[
\frac{\partial E}{\partial \rho} = -2\rho
\]
The standard product-moment correlation and its derivative are:

\[
\rho(y_1, y_2) = \sum_p (y_1^{(p)} - \bar{y}_1)(y_2^{(p)} - \bar{y}_2)
\]

\[
\frac{\partial \rho(y_1, y_2)}{\partial y_1^{(p)}} = \frac{1}{V_1 V_2}(y_2^{(p)} - \bar{y}_2) - \frac{V_2}{V_1} V_{12} \frac{1}{(V_1 V_2)^2}(y_1^{(p)} - \bar{y}_1)
\]

where \( V_1 = \sum_p (y_1^{(p)})^2 - (\sum_p y_1^{(p)})^2 \) and \( V_{12} = (\sum_p y_1^{(p)} y_2^{(p)}) - (\sum_p y_1^{(p)})(\sum_p y_2^{(p)}) \).

Sigmoidal output functions, weight-cost terms, and their derivatives may be found in any good neural network reference, such as that by Hertz, Krogh and Palmer (1986). Note that in the Pearson-based training, the network outputs are not forced into binary decisions — the “soft”, continuous values are employed directly in the optimization procedure.

The training of the right-hand network is entirely analogous. The two networks are typically trained simultaneously, step-for-step.

### 3.3.4 Experiments and Results

Reported here are the results from two sets of experiments with the coupled neural networks system described above. In Experiment 1 a number of training runs were performed for a very computation-intensive version of our method, from which our best predictability results were obtained. In Experiment 2 a much larger number of training runs were performed with a very fast variant of the algorithm, permitting a more detailed study of the dynamics and results of the mutually-supervised classification methodology. The best results to date, for each of the two sets of experiments, are presented below.

The criterion for choosing the “best” result in these experiments was based on how well the networks could be trained using different objective functions and not by selecting the algorithm and architecture that performed best on the predict set. If one chooses the architecture and training termination time based on the predict set, then contamination of the train and predict sets occurs, and prediction accuracy can be over-estimated. Use of a cross-validation set to choose the network architecture, as well as the optimal amount of training, is an acceptable procedure. However since the training times were already quite long (hours of CM5 time)
in Experiment 1, it was not possible to perform a full cross-validation study. The difference between accuracies reported on the train set, and on the predict set, does indicate that some degree of over-training occurred, and that the predict set results might possibly be improved by cross-validation or similar schemes. A validation set and simple cross-validation stopping criterion were used in Experiment 2; this scheme, along with a straightforward quadratic model cost ("weight-decay") (Hertz, Krogh and Palmer, 1986), did seem to produce more regularized models, decreasing the performance gap between train and predict sets.

The main dataset, from which were derived all the datasets used in the experiments described in this chapter, consisted of 105 proteins and is identical to that used in previous investigations (Kneller, Cohen and Langridge, 1990; Stolorz, Lapedes and Xia, 1991a). The proteins were divided into two groups: a set of 91 "training" proteins, and a distinct "prediction" set of 14 proteins. The resulting database is similar to the database used by Qian & Sejnowski (Qian and Sejnowski, 1988) in their neural network studies of conventional secondary-structure prediction. The training and prediction sets were chosen in such a way as to contain little homology between the two sets. When comparison to predictability of conventional secondary-structure classes was needed, we defined the conventional alpha, beta and coil states using the Kabsch and Sander definitions and therefore these states are identical to those used in previous work (Kneller, Cohen and Langridge, 1990; Stolorz, Lapedes and Xia, 1991a). A window size of 13 residues resulted in 16,028 training set examples and 3005 predict set (test set) examples. Effects of other windows sizes have not yet been extensively tested. All results, including conventional back-propagation training of Kabsch and Sander classifications, as well as two-net training of our new secondary-structure classifications, did not employ an extra symbol denoting positions in a window that extended past the ends of a protein. Use of such a symbol could further increase accuracy. The window size 13 was used throughout the experiments presented in this chapter because it was found to be effective in previous work on secondary structure prediction (Qian and Sejnowski, 1988; Kneller, Cohen and Langridge, 1990). In such studies it was found that smaller windows failed to provide sufficient local contextual information for prediction of the secondary structure of the central residue in the window; for windows of length larger than 13, the marginal gains in extra contextual information were swamped by noise. Other choices for this key variable were not extensively tested in our studies. The issue of length of discovered motifs is discussed later in this chapter.

Given that we chose algorithms and training and testing data without bias as to quality of
test-set results, and that we likewise based our training-time decisions on such uncontaminated
knowledge, we can report the "best" results on the prediction (test) set as a meaningful quantity.
It is a quantity that may be interpreted as an empirical upper bound on the performance of
our methods as tested thus far. This upper bound information may be augmented by any
reporting of: number of trials, mean values and standard deviations from the mean for any error
or predictability measures, and so on.

Networks yielding new classifications of secondary structure were obtained from random
initial weight values chosen from the uniform distribution between -0.2 and 0.2. However,
random initial conditions suffer to a certain extent from shallow local minima. We treated this
problem by repeated runs from different, random, initial weight values. One could attempt
to ameliorate the problem by first separately training both the sequence and structure nets to
predict the standard Kabsch and Sander classes (using conventional back-propagation), and then
using these synaptic weights as the initial values of a two-network run. However, we found that
the initial minimum is so deep that nothing new develops — the Kabsch and Sander definition
remains unless one initializes the network with random initial weight values.

**Experiment 1:**

The best results have been obtained with the Mathews objective function using an ar-
chitecture of five hidden units in each network. Therefore the “left-hand” (sequence win-
dow) and “right-hand” (tertiary-structure window) networks had feed-forward architectures of
(260 → 5 → 1) and (52 → 5 → 1) respectively\(^2\). Here, 260 = 13 × 20 for the unary encoding
of each of 13 contiguous amino-acid residues. and 52 = 13 × 4 for the \(\cos(\phi), \sin(\phi), \cos(\psi), \sin(\psi)\)
representation for each of the same 13 residues. Adjacent layers were fully interconnected. Hidden
and output units (“neurons”) employed sigmoidal activation functions (Hertz, Krogh and
Palmer, 1986).

Training was accomplished with the gradient-free Powell minimization procedure (Press
et al., 1988) and not by conventional back-propagation that employs gradients.

If one assigns the name “X1Aclass”\(^3\) to the newly-defined structural class, then we found

\(^2\)Each of these networks trained in Experiment 1 actually employed 2 output units. The two outputs in each
network were independent, and defined two independent subnetworks, in the sense of feed-forward processing and
prediction. As for learning, the first output of the sequence network was trained towards correlation with the
first output of the structure network, and likewise for the second output units of the respective networks. The
weights on the connections between the input and hidden layers were responsible to the optimization of both sets
of correlations. Interestingly, this architecture produced better learning and prediction results, for at least one of
the two sets of corresponding outputs, than the simpler single-output case tried in other experiments.

\(^3\)“X1Aclass” means “the first class (hence ‘A’) found in Experiment 1”. Similarly, “X3BClass” would mean
that one can evolve paired networks that classify local windows of structure into a "X1AcIass versus NotX1AcIass" dichotomy with higher predictability than the predictability of the conventional. alpha. beta. coil secondary-structure classes. Results of the two network training using the protocol of Experiment 1 is reported in Table 3.1. It may be seen that the Mathews coefficient on the prediction set of the newly-defined secondary-structure classes is -0.43. This result is for a two-state dichotomy. To compare the predictability of these new two-state classes to the conventional three-state secondary-structure classes of alpha. beta and coil. it is necessary to train three back-propagation networks to perform three separate dichotomies into $\alpha$/not-$\alpha$. $\beta$/not-$\beta$ and coil/not-coil. These results are also reported in Table 3.1. It may be seen that the predictability of the new two-state dichotomies are significantly higher than any of the $\alpha$/not-$\alpha$. $\beta$/not-$\beta$ or coil/not-coil dichotomies. Adding hidden units gives negligible accuracy increase for predicting the traditional classes: however they are crucial for accurately predicting the new classes.

The negative sign of the two-network result indicates anti-correlation — a feature allowed by our objective function. The sign of the correlation is easily assessed on the train set and then can be trivially compensated for during prediction.

Table 3.1: Mathews correlation values ($M_k$) between sequence and structure network classifications. on training and prediction sets of examples. For networks trained in Experiment 1. corresponding values for X1AcIass classifications are compared with values for neural networks trained for dichotomous prediction of traditional classes $\alpha$-helix. $\beta$-sheet. and coil.

<table>
<thead>
<tr>
<th></th>
<th>$M_k$ on Train Set</th>
<th>$M_k$ on Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1AcIass</td>
<td>-0.51</td>
<td>-0.43</td>
</tr>
<tr>
<td>$\alpha$-helix</td>
<td>0.37</td>
<td></td>
</tr>
<tr>
<td>$\beta$-sheet</td>
<td>0.31</td>
<td>0.26</td>
</tr>
<tr>
<td>Coil</td>
<td>0.41</td>
<td>0.39</td>
</tr>
</tbody>
</table>

A natural question to ask is whether the new classes are simply related to the more conventional classes of alpha-helix. $\beta$-strand. and coil. A simple answer is to compute the Mathews correlation coefficient of the new secondary-structure classes with each of the three Kabsch and Sander classes. for those examples in which the sequence network agreed with the structure network’s classification. The correlation with Kabsch and Sander’s $\alpha$-helix is highest: a Mathews coefficient of 0.25 was obtained on both the train set and predict set. There is therefore a significant degree of correlation with the conventional classification of $\alpha$-helix, but

"the second class found in Experiment 3", and so on.
significant differences exist as well. The new classes are a mixture of the conventional classes, and are not solely dominated by any of alpha, beta or coil.

Conventional α-helices comprise roughly 25% of the data (for both train and predict sets), while the new X1Aclass comprises 10%. It is quite interesting that an evolution of secondary-structure classifications starting from random initial conditions, and hence completely unbiased towards the conventional classifications, results in a classification that has significant relationship to conventional helices but is more predictable from amino-acid sequence than conventional helices. In Table 3.2 we compare the assignment of structural features into X1Aclass/NotX1Aclass categories, with the conventional assignment of structural features into α-helix, β-strand and coil, for the protein Actinidin (which is in the predict set). The similarity, and differences, of X1Aclass secondary structure to conventional α-helices is apparent.

Table 3.2: Three representations of Actinidin (sulfhydryl Proteinase) are displayed below. (Brookhaven Protein Designator: 2ACT.) The four groups of three lines illustrate the Kabsch and Sander’s definition of secondary structure (Helix, Beta, and Coil) in relation to the predicted X1Aclass secondary structure, and the target X1Aclass secondary structure, for protein 2ACT. Top line in each group: H=Helix, B=Beta chain, “=”=Coil, representing the conventional Kabsch and Sander’s secondary-structure classes. Second line in each group: “1”=X1Aclass, “.”=NotX1Aclass, representing the predicted (left-hand network) X1Aclass secondary-structure categories. Third line in each group: “1”=X1Aclass, “.”=NotX1Aclass, representing the target (right-hand network) X1Aclass secondary-structure classes.
Experiment 2:

The fastest experimental implementation of our method employed gradient-based optimization of the standard correlation, $\rho$, between two feed-forward networks. Discrete ("hard") classes were implemented for prediction, though the objective function was computed over real-valued network outputs, thereby ensuring smoothness. A conjugate-gradient procedure (Press et al., 1988) with a sophisticated line-search component performed the optimization.

The same basic network architectures as in Experiment 1 were used, with varying numbers of hidden units. The specific predictive accuracy results quoted below were produced by networks with two hidden units each. A somewhat unique cross-validation method was used to determine when to stop training. Because the Mathews correlation measure was to be used to assess the success of training and prediction, even though $\rho$-correlation was used as the objective function, training was terminated when the Mathews correlation between networks, on the validation set, began to decrease. The validation set comprised 1000 examples chosen randomly and removed from the training set.

The experiments with this implementation proceeded much faster than the Experiment 1 simulations. On a Silicon Graphics Iris 4D machine, slower and significantly less parallel than the CM5 system, it was possible to train each network to acceptable predictability levels within an hour or two at most.

This speedup allowed us to perform a larger number of neural network simulations and in fact to explore in some depth a particular region of classification space. Of 75 training runs from different random initial weight configurations with the same $(260 \rightarrow 2 \rightarrow 1)$ and $(52 \rightarrow 2 \rightarrow 1)$ network architectures, nine resulted in prediction set correlations $|M_k| \geq 0.39$. (Again, $260 = 13 \times 20$ for the unary encoding of each of 13 contiguous amino-acid residues, and $52 = 13 \times 4$ for the $\cos(\phi), \sin(\phi), \cos(\psi), \sin(\psi)$ representation for each of the same 13 residues. Adjacent layers were fully interconnected. Hidden and output units ("neurons") employed sigmoidal activation functions (Hertz, Krogh and Palmer, 1986).)

Of these nine well-trained systems, four were very similar, in terms of producing classifications very highly correlated ($|M_k| > 0.8$) with each other. Two of these, which we call X1Bclass and X1Cclass, had the highest predictability values in Experiment 2 and also display closer relationships with two of the traditional Kabsch and Sander classes than does X1Aclass. Closest to the traditional classes is X1Dclass, which is also slightly less predictable than the other X1class classifications. These results are summarized in Tables 3.3 and 3.4.
As the tables indicate, the X1Bclass and X1Cclass classifications are more predictable than the standard secondary-structure classes, with Mathews values of -0.42 and 0.42 as compared to the maximum of 0.39 for coil. These prediction set correlation values are almost equal to the -0.43 computed for X1Aclass, though the training set values are not nearly in the X1Aclass range.

Table 3.3: Mathews correlation values ($M_k$) between sequence and structure network classifications, on training and prediction sets of examples. For networks trained in Experiment 2, corresponding values for X1Bclass, X1Cclass, and X1Dclass classifications are compared with values for neural networks trained for dichotomous prediction of traditional classes $\alpha$-helix, $\beta$-sheet, and coil.

<table>
<thead>
<tr>
<th></th>
<th>$M_k$ on Train Set</th>
<th>$M_k$ on Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1Bclass</td>
<td>-0.46</td>
<td>-0.42</td>
</tr>
<tr>
<td>X1Cclass</td>
<td>0.44</td>
<td>0.43</td>
</tr>
<tr>
<td>X1Dclass</td>
<td>0.43</td>
<td>0.39</td>
</tr>
<tr>
<td>$\alpha$-helix</td>
<td>0.37</td>
<td>0.33</td>
</tr>
<tr>
<td>$\beta$-sheet</td>
<td>0.31</td>
<td>0.26</td>
</tr>
<tr>
<td>Coil</td>
<td>0.41</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 3.4: Mathews correlation values, for trained network on prediction set of examples, for each of X1Bclass, X1Cclass, and X1Dclass as measured against each other and against traditional classes $\alpha$-helix, $\beta$-sheet, and coil.

<table>
<thead>
<tr>
<th></th>
<th>X1Bclass</th>
<th>X1Cclass</th>
<th>X1Dclass</th>
<th>$\alpha$-helix</th>
<th>$\beta$-sheet</th>
<th>coil</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1Bclass</td>
<td>1.00</td>
<td>-0.89</td>
<td>-0.82</td>
<td>-0.32</td>
<td>0.08</td>
<td>0.37</td>
</tr>
<tr>
<td>X1Cclass</td>
<td>1.00</td>
<td>0.84</td>
<td>0.32</td>
<td>-0.00</td>
<td>-0.30</td>
<td>-0.34</td>
</tr>
<tr>
<td>X1Dclass</td>
<td>1.00</td>
<td>0.46</td>
<td>-0.01</td>
<td>-0.34</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Controls

To enable a better understanding of the significance of the predictability results, some control experiments were performed with different input datasets.

First, one might wonder whether the configuration of coupled nets with the correlation-based objective functions might find a high between-spaces correlation in almost any dataset: if such were true, it would lessen the significance of the values attained on the A, Z data. In order to investigate this possibility, a set of 20 Pearson-training runs were performed on a dataset $X_{rand} = A_{rand} \times Z$, in which all of the $N_w$-sized sequences entered into the left-hand network were generated randomly from a single-amino-acid probability distribution identical to that estimated from the real A data. In other words, a memory-less Markov source using the same $p$(Alanine), $p$(Cysteine), ..., $p$(Tryptophan) probabilities as found in A was used to generate $A_{rand}$. The absolute values for the Matthews correlations of trained networks for the two-class
outputs over the randomized \( A' \) and original \( Z \) datasets are summarized by: \( \text{mean} = 0.017, \sigma^2 = 0.0035, \max = 0.031 \). all measured on the test set. A perhaps better test of the discovery of correlation in randomized data is one that employs real \( A \) and \( Z \) input datapoints but without matching them correctly — that is. \( X_{\text{rand}} = \{(\bar{a}_i, \bar{z}_j)\}_{ij} \), where the \( i, j \) pairings are chosen randomly by permuting the \( A \) and \( Z \) cases. This was done. and the Matthews correlation results for the Pearson-trained system given these inputs are as follows: \( \text{mean} = 0.028, \sigma^2 = 0.0046, \max = 0.039 \) over 20 trials.

Second, it was investigated whether an encoding of \( A \) in terms of physico-chemical property vectors (Nakai. Kidera and Kanehisa. 1988: Greller. Steeg and Salemme. 1991) would lead to better predictability results than those attained using the standard unary encoding of the twenty amino acid symbols. A key reason for testing this hypothesis is the significant decrease in network size and connectivity enabled by such a recoding. and the regularization benefits (better generalization to testing set) that would possibly follow.

Four properties were chosen: Charge. Hydrophobicity. Side-Chain Volume. and Side-Chain Entropy (Nakai. Kidera and Kanehisa. 1988). Each property index assigns values to each of the twenty amino acids: the indices were normalized to \([0.1]\) ranges. The modified data required a modified network architecture. The input layer of the \( A \)-network contained \( 4N_w = 52 \) input units. like the \( Z \)-network. 4 hidden units were used in each network. and a single output. In the 20 simulations performed. the modified input format did not yield improvements in predictability of the resulting classifications. The values obtained on the test set: \( \text{mean} = 0.18, \sigma^2 = 0.0147, \max = 0.21 \). On the training set: \( \text{mean} = 0.20, \sigma^2 = 0.097, \max = 0.23 \). One might infer from these results that in the re-representation of the sequence data some important information was lost. Whether the inclusion of additional physico-chemical properties (molecular weight? accessible surface area?) might improve upon these results is worthy of further inquiry.

Finally. it was investigated whether the nonlinearity of the sigmoidal activations in the coupled networks are necessary to finding predictable classifications in the data. The Canonical Correlation method was tested to see how high a (Pearson) correlation could be obtained between linear combinations of \( a \) and \( z \) vectors. (See the text by Mardia \textit{et al.} (1979) for details of the algorithm: see Becker’s Ph.D. dissertation (1992) for a comparison of canonical correlation to IMAX on other tasks.) The resulting optimal linear combinations were then used to generate outputs \( O_a \) and \( O_z \) which were thresholded into \([0.1]\) and tested for Matthews correlations between transformed \( \bar{a} \) and \( \bar{z} \) vectors. The results indicate that the nonlinear sigmoidal units in
the original feed-forward networks are finding $A, Z$ correlations not detectable through purely linear modelling: Matthews value on training set = 0.22; on test set = 0.20. One can also compare the *Pearson* correlation values obtained (0.27 and 0.23) with the best Pearson values obtained with the Pearson training on the original data (0.69 and 0.63).

### 3.3.5 Possible Extensions

The most obvious next step for future work with Method 1 is the move to $m > 2$ classes. Both discrete and continuous versions of the mutual information objective function. and of the correlation objective functions. can be extended to multiple classes.

Additionally, more effort can be made and more techniques tried in attempting to work around the local minima problem, especially for the mutual information objective function.

A fairly major drawback to the correlation-based objective function is that it does not readily permit an analysis of the resulting classes beyond whatever can be said about how the $A$-classes and $Z$-classes interrelate. That is, it does not permit the sort of Minimum Description Length or Bayesian analysis applied to the classifications produced by Methods 2 and 3 (see below). unless very particular assumptions are made about the shapes of the $A$ and $Z$ classes. Parameterized distributions. such as multi-dimensional gaussians. would enable us to say more about the quality of classifications produced by the method: but one of the motivations for using Method 1 in the first place was the freedom to avoid such assumptions. In this regard. it must be noted that the use of Pearson correlation can be said to make implicit assumptions about the input data: Correlation is a useful measure of (multi-)collinearity primarily when the scatter of points is elliptical. i.e., when the data is distributed in a roughly gaussian distribution. When the gaussian condition holds. the correlation coefficient for the $\{O_i^L\}$ and $\{O_i^R\}$ is zero if and only if $O^L$ and $O^R$ are independent. When the condition does not hold. the correlation may be zero even when $O^L$ and $O^R$ are not independent.

Method 2 and Method 3 represent attempts to bring the imposition and analysis of both predictability and optimal class structure together within a unified framework. Further. Method 3 attempts to do this without making possibly impractical assumptions about the structure of classes within the $A$ space.
3.4 Method 2: Joint Density Estimation with Gaussian Mixture Models

3.4.1 Intuition and Background — The Use of MDL Objectives

The classifications produced by Method 1 are more predictable from amino-acid sequences than are traditional secondary-structure classifications, at least for the two-class case and as measured by the Mathews correlation function. The constraint of predictability, i.e., of agreement between the outputs of the sequence network and the structure network, was explicitly built in via the objective function. What other useful properties should a secondary-structure classification have, and how may these be implemented in an objective function? For example, to effectively use secondary-structure predictions as an aid to tertiary-structure prediction, the secondary-structure classification must significantly constrain the \( \phi, \psi \) angles. It is a difficult and still-open problem to define, and maximize, an objective function that quantifies the structural constraints induced by a secondary structure classification. However, towards this end, it is plausible that the same Bayesian/MDL framework which provides the basis for modeling a single space (Zemel, 1994; Rissanen, 1985) can be used to advantage in managing the within-space versus between-spaces tradeoffs inherent in joint-space modeling. An exploration of this idea might begin with an investigation of a joint \( A, Z \) probability density modelling approach to defining secondary structure classes: this section presents such a method and the results of experiments with the method.

Rissanen proposed the Minimum Description Length (MDL) principle (Rissanen, 1985) for modelling and machine learning as a formalism that captures our intuition about what makes a good model of a set of data (in other words, what makes a good motif). He noted the ubiquitous tradeoff between accuracy and succinctness that one faces when trying to define or infer a model that encodes a set of data, and he developed the well-known communication paradigm in which to address the tradeoff. In the communication game, there is a Sender \( S \) and a Receiver \( R \), and \( S \)'s goal is to transmit to \( R \) a dataset \( X = \{x_i\}_1^M \) over a finite-bandwidth communication channel. Accuracy refers to the degree to which the information in \( X \) is preserved by the encoding, and succinctness is motivated by the idea that a model that captures the underlying regularities in the data facilitates a compact encoding of it. The communication game brings the tradeoff into clear focus: one wants to compress the message as much as possible to save time and cost, but one also wants to transmit as much of the information as is possible. The same opposing
motivations are at work when modelling a set of data. or when designing a learning algorithm to automatically infer a model: the model should be parsimonious and should not contain unnecessary features that "overfit" the data. but it should also explain as much as the data as possible. as accurately as possible. (The succinctness requirement is a computational version of Ockham’s Razor. which favors simple theories over complex ones with the same explanatory power. The MDL framework may be recast into the Bayesian learning framework (MacKay. 1992; Zemel. 1994) where its relation to regularization and to approximation theory (Poggio and Girosi. 1990) is more evident.) The MDL principle asserts that the best model for a dataset \( X \) is one which minimizes the sum of two cost terms:

1. the length (in bits) of the model. and

2. the length of the dataset when encoded using the model as a predictor for the data. i.e., the size of the dataset re-represented in the new code. plus the degree of misfit. of "truncation error" between the encoded data and the original data.

The two components listed above follow easily from the rules of the communication game. The Sender \( S \) must first send a (relatively) few bits telling the Receiver \( R \) what kind of code is to be used. Then she must send the encoded data items. Finally she must send whatever bits are necessary for \( R \) to be able to recover the data items exactly from the encoded versions.

If one applies the MDL principle to the search for good protein secondary structure classes. the data is \( Z \). and the model is the mapping from the \( \{z_i\} \). to the labels of the \( m_Z \) classes. \( C^Z_1. C^Z_2. \ldots. C^Z_{m_Z} \). The length of the model is a small fixed number of bits for specification of the model’s parameters. The cost of the data is the size of the set \( Z^c = \{z^c_i\} \) of data items encoded in terms of the class label codes. plus the cost of the misfit. i.e., the reconstruction cost. the additional bits of information needed to reconstruct each original \( z_i \) from its encoding \( z^c_i \).

The link between the MDL formulation and the Bayesian and Maximum Likelihood formulations used in most research on machine learning (supervised and especially unsupervised) is provided by Shannon’s Optimal Coding Theorem (Shannon and Weaver. 1964). The theorem states that the ideal code. generated from an optimal data description language \( L^* \) based on a known probability distribution \( p() \) over discrete values (events) \( x \) for a random variable \( X \). will have an expected length given by

\[
|L^*(x)| = -\log_2 p(X = x).
\]
The relation is made even more dear in the case of parameterized models and continuous data. Consider a model $\mathcal{M}$ with parameters $\tilde{\theta}$. The probability of the data $X$ given the model can be computed by integrating over possible values of $\tilde{\theta}$:

$$p(X|\mathcal{M}) = \int p(X|\tilde{\theta}, \mathcal{M}) \, p(\tilde{\theta}|\mathcal{M}) \, d\tilde{\theta}.$$ 

As discussed in Zemel's Ph.D. dissertation (Zemel, 1994), it is common in many problems for the posterior distribution for $\tilde{\theta}$ to have a sharp peak at the most probable value $\tilde{\theta}^\ast$, allowing the integral to be approximated the product of its height at $\tilde{\theta}^\ast$ and its widths $\Delta\tilde{\theta}$:

$$p(X|\mathcal{M}) \approx p(X|\tilde{\theta}^\ast, \mathcal{M}) \, p(\tilde{\theta}|\mathcal{M}) \Delta\tilde{\theta}.$$ 

The first factor in this product on the right-hand side is the maximum likelihood fit of the model to the data, and the second factor $p(\tilde{\theta}|\mathcal{M}) \Delta\tilde{\theta}$ is the "Ockham factor" (MacKay, 1992).

Ignoring for now the Ockham factor and concentrating on the asymptotic costs of representing the data given the model, we find our MDL goal to be the maximization of the log likelihood of the data given the model, that is, the minimization of:

$$Cost(X) \approx -\log p(X|\mathcal{M})$$

(All summations and products over $i$ are $i = 1, 2, \ldots, M$ in all equations in this section.)

The reader will note from this formulation that the expected cost, per data item $x_i$, is therefore simply:

$$E[Cost(x_i)] = -\sum_i p(x_i|\mathcal{M}) \log p(x_i|\mathcal{M})$$

A particularly common and useful model for probability density estimation is the mixture of gaussians model (Nowlan, 1991; McLachlan and Basford, 1988; Nowlan, 1989). Here the
clusters, the classes emerge from latent class modelling in which each possible class corresponds to a parameterized multivariate gaussian distribution in the input space: each data item $x_i$ is considered to have been generated by any of the gaussians with some particular probability, based on the proximity of $x_i$ to the mean of the gaussian as well as on the variance of the gaussian:

$$Cost(X) \approx -\log p(X|M)$$

$$\approx - \sum_i \log p(x_i|M)$$

$$= - \sum_i \log \sum_{k=1}^m \pi_k p_k(x_i)$$ \hspace{1cm} (3.16)

where $\pi_k$ is the prior probability of class $k$, and the class-conditional probability density for class $k$ at $x_i$ is given by

$$p_k(x_i) = p(x_i|\bar{\theta}_k)$$

$$= \exp\left(-\frac{1}{2} \sum_j^n \frac{(x_{ij} - \mu_{kj})^2}{\nu_{kj}}\right) \hspace{1cm} (2\pi)^{n/2} \prod_{j=1}^n (\nu_{kj})^{1/2}. \hspace{1cm} (3.18)$$

In the above equation for the probability of $x_i$ having been generated by the $k$th gaussian, $j$ indexes the $n$ dimensions of the data $x_i$ and of the gaussian's mean vector $\mu_k$ and variance vector $\nu_k$. It is often assumed that the dimensions of the vectors may be treated as independent, given the particular class $k$, that is

$$p(x_{i1}, x_{i2}, \ldots, x_{in}|\bar{\theta}_k) = \prod_{j} p(x_{ij}|\bar{\theta}_k). \hspace{1cm} (3.19)$$

A more general formulation would have:

$$p_k(x_i) = p(x_i|\bar{\theta}_k)$$

$$= \frac{\exp\left(-\frac{1}{2} (x_i - \bar{\mu}_k)^T V_k^{-1} (x_i - \bar{\mu}_k)\right)}{(2\pi)^{n/2} |V_k|^{1/2}}. \hspace{1cm} (3.20)$$
where $V_k$ is the covariance vector for the dimensions of the class $k$ gaussian, and $|V_k|$ is its determinant.

It is important, for understanding the motivation behind both Methods 2 and 3, to understand the connection between probability density estimation within a space $Z$ and the modelling of class structure within $Z$. They are in fact two views of the same computational goal, or perhaps two overlapping regions along a continuous spectrum of goals. In probability density estimation, the goal is the production of a model $M$ of the probability density $p(\tilde{z})$ at each possible point $\tilde{z} \in Z$. For continuous spaces $Z$ and with finite training data this goal is unattainable. What is attainable is the production of a model $M$ such that, given any $\tilde{z} \in Z$, we can use $M$ to estimate a probability that $\tilde{z}$ is nearby (similar to) some particular exemplar point $\tilde{z}'$. This is to say, we can use $M$ to estimate the probability that $\tilde{z}$ is in some class of points, of which $\tilde{z}'$ is the exemplar (class centroid, etc.) In the limit of a very large number of classes and exemplars, the goals of class structure modelling and density modelling merge (just as parametric density estimation methods such as gaussian mixture modelling give way to non-parametric approaches such as nearest-neighbour methods (Parzen, 1962; Huang and Lippmann, 1987)). In the extreme, every point is in its own class, and the nearest exemplar to every point is itself. Thus, loosely speaking, one may say when the $j$th gaussian unit of a mixture of gaussians network "wins" on a particular input $\tilde{z}$ that $\tilde{z}$ is in the $j$th class, or alternatively that $\tilde{z}$ was most likely generated by the $j$th gaussian, or alternatively that $\tilde{z}$ is nearest the $j$th exemplar (in terms of the Mahalanobis distance (Nowlan, 1990)).

One could apply the method of maximum likelihood fit of gaussians to the discovery of classes in $Z$. Indeed, Hunter and States (1992) have done this (for their own, different set $Z$ of short tertiary structure pieces) and can claim to have found some kind of "intrinsic" class structure in their structure space, using a version of the AutoClass latent class modelling system (Cheeseman et al., 1987). It can be said that they discovered some kind of "natural" secondary structure classification and one that, given certain assumptions, achieves at least a local optimum in terms of compression and retention of tertiary structural constraints.

The Hunter and States approach has little to say about predictability of the discovered structure classes. The secondary structure classes were evolved, automatically, subject only to constraints on and input from the tertiary structure side of the picture. Might we be able to build in sequence-structure predictability, while retaining a degree of "naturalness" to the classes? Our Method 2 aims to achieve this by doing essentially the same thing to $X = A \times Z$
as Hunter States did to just $Z$ alone.

It is clear why one expects such a method might be able to find and extract some kind of class structure immanent in $Z$, and in $A$; but why might it be expected to find \textit{predictable} classes?

An answer is suggested by considering again the mutual information function $I(A:Z)$ and recalling the MDL communication game:

\begin{align}
I(A;Z) &= D(p(\tilde{a}, \tilde{z})||p(\tilde{a})p(\tilde{z})) \\
&= \sum_w p(\tilde{a}, \tilde{z}) \log \frac{p(\tilde{a}, \tilde{z})}{p(\tilde{a})p(\tilde{z})} \\
&= \mathbb{E}_{p(\tilde{a}, \tilde{z})}[\text{Cost}(p(\tilde{a}, \tilde{z})) - \text{Cost}(p(\tilde{a})p(\tilde{z}))]
\end{align}

Here, the \textit{Cost} is assumed to be defined relative to some implicit model and encoding language.

As Cover and Thomas (1991) put it, "The relative entropy [Kullback measure $D(p||q)$] is a measure of the inefficiency of assuming that the distribution is $q$ when the true distribution is $p$." In this case, the mutual information is the measure of the inefficiency of assuming $A$ and $Z$ are independent, and then coding and transmitting them separately, versus coding and sending the joint distribution. The mutual information thus quantifies the information shared by $A$ and $Z$ that can be exploited in sending $(A, Z)$ together more cheaply than sending them separately.

Therefore, if we use a maximum likelihood procedure that minimizes the expected value of $- \log p(\tilde{a}, \tilde{z})$ given a model $\mathcal{M}$, then it tends to minimize

$$H(A, Z|\mathcal{M}) = H(A|\mathcal{M}) + H(Z|\mathcal{M}) - I(A;Z|\mathcal{M})$$

The algorithm will therefore tend to increase the mutual information between the $A$-classes and the $Z$-classes, under the model. It will therefore \textit{tend to} increase the predictability of $Z$ classes from $A$ classes. In fact, there is a tradeoff between what we might call \textit{within-space} optimality, for the spaces $A$ and $Z$, and a \textit{between-spaces} optimality. Some low-$E$ results of the joint log-likelihood optimization might display high correlation between the states but a relatively poor compression of the $\phi, \psi$ representation within $Z$, for example; and others might
display good encodings of A and Z separately, but fail to exploit $I(A: Z)$ in the joint encoding, resulting in lower predictability.

### 3.4.2 The Objective Function

The general formulation of the objective function for maximum log-likelihood estimation, with a gaussian mixture model of the joint ($A, Z$) probability density, is (omitting the "conditional on the model $M$" notation for brevity):

$$E = - \sum_{i=1}^{M} \log p(\tilde{a}_i, \tilde{z}_i).$$

For the case of $m_A$ classes in $A$, $m_Z$ classes in $Z$, and covariance across the dimensions within each gaussian, the tied-mixture formulation of $p(\tilde{a}_i, \tilde{z}_i)$ is given by

$$p(\tilde{a}_i, \tilde{z}_i) = \sum_{j}^{m_A} \sum_{k}^{m_Z} \pi_{jk} p(\tilde{a}_i | \theta_j^A) p(\tilde{z}_i | \theta_k^Z)$$

$$= \sum_{j}^{m_A} \sum_{k}^{m_Z} \pi_{jk} \exp\left( -\frac{1}{2}(\tilde{a}_i - \mu_j)V_j^{-1}(\tilde{a}_i - \mu_j) - (\tilde{z}_i - \nu_k)W_k^{-1}(\tilde{z}_i - \nu_k) \right) \frac{(2\pi)^{n/2}|V_j|^{1/2}|W_k|^{1/2}}{(2\pi)^{n/2}|V_j W_k|^{1/2}}$$

where $\mu_j, \nu_k$ are the means and $V_j, W_k$ are the covariance matrices for the $j$th $A$-class and the $k$th $Z$-class, respectively.

The use of general covariance matrices adds many degrees of freedom to the estimation (learning) task, and therefore poses problems for both speed and regularization. The work reported in this section employs the conditional independence assumption outlined above. Also, for the sake of simplicity and in order to complete a timely study of the methodology, we have assumed $m_A = m_Z$. These two assumptions combine to allow for a simple formulation in which there are only $m_X$ joint classes, and the vectors are composed of $\text{dim}(A) + \text{dim}(Z)$ independent (given the model and class) dimensions:

$$p(\tilde{a}_i, \tilde{z}_i) = \sum_{j}^{m_A} \sum_{k}^{m_Z} \pi_{jk} p(\tilde{a}_i | \theta_j^A) p(\tilde{z}_i | \theta_k^Z)$$

$$= \sum_{j}^{m_A} \sum_{k}^{m_Z} \pi_{jk} \exp\left( -\frac{1}{2} \sum_{h}^{N_W} \frac{(z_{ih} - \mu_{ih})^2}{v_{ih}^2} \right) \exp\left( -\frac{1}{2} \sum_{g}^{N_Z} \frac{(z_{ig} - \mu_{ig})^2}{v_{ig}^2} \right) \frac{(2\pi)^{N_w/2}(\prod_{h}^{N_W} v_{ih}^2)^{1/2}}{(2\pi)^{N_Z/2}(\prod_{g}^{N_Z} v_{ig}^2)^{1/2}}$$
Figure 3.5 illustrates the use of the simple gaussian-mixture joint-density-estimation objective function, implemented with a feed-forward network.

**Joint A, Z Density Modelling with Mixture of Gaussians**

\[
E = - \sum_{i=1}^{M} \log \sum_{k=1,z} q_k p(\tilde{x}_i | \tilde{\theta}_k)
\]

For simplicity, the input layers are shown to have \(N_u\) units each; however, the actual input dimensions are \(20N_u\) for \(\tilde{a}_i\) (for the unary encoding of each amino acid) and \(4N_w\) for \(\tilde{z}_i\) (for the \((\cos \phi, \sin \phi, \cos \psi, \sin \psi)\) encoding of each residue). In the experiments performed thus far, \(N_w = 13\). On each iteration, on each input example and for each class, the class-conditioned probability of the input is combined with the mixing proportion and normalized to produce the **responsibility** values (not shown). Given the responsibilities, the re-estimations of the free parameters (means, variances, and mixing proportions) are performed, based on setting the derivatives of \(E\) equal to zero.

**3.4.3 Learning Rules and Prediction Rules**

Consider the 2-class case again. When a mixture network is activated on a particular input \(\tilde{x}_i\), the output units of the network compute \(p(\tilde{x}_i | \tilde{\theta}_1)\) and \(p(\tilde{x}_i | \tilde{\theta}_2)\) Each of these values can
be multiplied by the corresponding prior probability $\pi_1$ or $\pi_2$ and normalized. such that two responsibility terms are produced:

$$r_{ij} = \frac{\pi_j p(\bar{X}_i | \bar{\vartheta}_j)}{\sum_{k=1,2} \pi_j p(\bar{X}_i | \bar{\vartheta}_k)}$$

The reader will recognize that $r_{ij}$ is just the normalized $p(C_j | \bar{X}_i)$.

It is also easy to break down these quantities into their $A$ and $Z$ components, for example:

$$r_{ij}^A = \frac{\pi_j^A p(\bar{A}_i | \bar{\vartheta}_j^A)}{\sum_{k=1,2} \pi_j^A p(\bar{A}_i | \bar{\vartheta}_k^A)}$$

and similarly for $r_{ij}^Z$.

**Prediction of structural class given sequence input:**

Once the mixture network is trained, one can use it to predict the secondary structure, i.e., the structural class, for a new data item, such as a sequence fragment $\bar{a}_{new}$ from a protein sequence with unknown structure. One simply "freezes" the means and variances for the $A$-half of the network, processes the network on the input $\bar{a}_{new}$, and reads the $A$-parts of the output.

$r_{new,1}^A, r_{new,2}^A$

This example also suggests the proper way to measure the predictability of the secondary structure classes produced from the training. The measure used in our analysis is simply the Matthews correlation $M(r_{ij}^A, r_{ij}^Z)$ of the class predictions made by the $A$- and $Z$-halves of the network, over all the training examples $\bar{X}_i = (\bar{a}_i, \bar{z}_i)$.

**Training the parameters of the model:**

For maximum likelihood fitting of finite mixture models over certain well-behaved distributions, there exists a fast minimization technique with guaranteed convergence properties. The *Expectation-Maximization (EM)* (Dempster, Laird and Rubin, 1976) method is an alternation method, one of a general category of algorithms that can be used to estimate latent class models or other models with latent or unobserved variables. The optimization problem is to maximize $p(X | \bar{\theta})$ over $\bar{\theta}$: rather than maximize it directly, we represent our latent variable(s) $Y$ and break down the task on each iteration $t$ into two steps:

**E Step** Compute the conditional distribution $P(Y | X, \bar{\vartheta}_{t-1})$

**M Step** Set $\bar{\vartheta}_t$ to the $\bar{\theta}$ that maximizes $E_{Y,X,\bar{\vartheta}_{t-1}}[\log p(Y | X | \bar{\theta})]$. 
In the mixture of gaussians case, the missing \( Y \) is actually the vector of \( r_j \) responsibility terms, that is, the \( p(C_j|X) \) terms. The parameters \( \bar{\theta} \) are the class means, variances, and mixing proportions. The maximization in the \( M \) step can be solved directly by setting

\[
\frac{\partial E(\bar{r})}{\partial \theta} = 0.
\]

where \( E \) is just the objective function and \( E(\bar{r}) \) means \( E \) given particular responsibility values \( \bar{r} \) computed over each of the \( \bar{x} \in X \).

After much algebraic manipulation, the following simple update rules are derived. For each iteration, and for each dimension \( h \) of \( \bar{x}_i \in X \) and each class \( C_j \), we have:

\[
\mu_{jh} = \frac{x_{ih} r_{ij}}{\sum_g r_{gj}}
\]

\[
v_{jh} = \frac{x_{ih}^2 r_{ij}}{\sum_g r_{gj}}
\]

\[
\pi_j = \frac{\sum_g r_{gj}}{M}
\]

where \( M \) is the number of examples.

### 3.4.4 Experiments and Results

A set of 100 simulations were performed with the objective function and learning and prediction rules of Method 2. The first 50 simulation runs employed exactly the same data as used in the Method 1 (and Method 3) experiments. The second set of 50 runs used a modification of the \( A \) part of the data. The change was made because the runs with the original \( X = (A, Z) \) data produced \( A, Z \) class correlation values (as measured by the Matthews measure) too close to zero (mean = 0.074, max = 0.12 on test set). Such predictability values make the corresponding structure motifs essentially useless as secondary structure motifs. It was clear from the beginning that the mixture of gaussians, with independent dimensions, is a very poor representation of sequential data. It was hoped, after the first 50 runs, that a re-representation of the sequence data in terms of only the most informative 52 of the 260 dimensions would help
matters by reducing noise, and it did. A principal component analysis (Mardia, Kent and Bibby, 1979) was performed on the $A$ training and testing data in order to find the 52 dimensions of maximal variance: these eigenvectors of the original covariance matrix corresponding to these 52 dimensions were used as a basis over which to re-represent all the $a_i \in A$. The best results from the second 50 runs are shown in the tables below.

The network architecture, for the runs using the modified data, was $(52+52 \rightarrow 2)$. Each of the runs was begun from a random initial starting point in parameter space, with initial means chosen from gaussian distributions centered on 0.5 and with variances 0.3 or 0.4.

The simulation runs, and their corresponding final classifications, seemed to fall into two categories. Roughly one-third of the runs resulted in trivial classifications — all of the examples being placed into one of the two classes, resulting in a very poor value for the objective function. The other two thirds of the runs produced nearly identical objective function values, and, as will be seen in Section 3.6, nearly identical classifications.

Two of the best of the resulting classifications — called X2Aclass and X2Bclass — in terms of predictability on both training and testing sets, are featured in Table 3.5.

One can note that the predictability values for the two new classes are significantly lower than those attained using Method 1. The numbers are very close to those for the least predictable (using dichotomous classification) of the traditional classes, $\beta$-sheet.

One of the possible advantages of a method like Method 2 is that it might trade off some between-spaces predictability for an improvement in within-space modelling of $p(A, Z)$, and possibly also of $p(A)$ and/or $p(Z)$ individually. A classification of the $Z$ data with a low MDL-cost is one of our goals, because such a classification would seem to yield secondary structure encodings of sequence data that might prove very useful in tertiary structure prediction. A comparison of our methods in terms of this MDL cost is presented in Section 3.6 below.

### 3.4.5 Possible Extensions

Thus far, we have taken advantage of Shannon's result that provides a convenient lower bound on the MDL cost of an encoding of data. This log likelihood bound is an approximation that enables the use of efficient and well-understood optimization techniques. They key idea in this version of Shannon's theorem is that the total encoding cost — the cost of the model parameters, plus the cost of the code bits, plus the cost of reconstructing the data from the code — has a lower bound given by the log likelihood of the data given the model, and that this bound
Table 3.5: Mathews correlation values ($M_k$) between sequence and structure classifications, as measured by responsibility terms $r^A_{ik}$ and $r^S_{ik}$, respectively, from the sequence and structure sub-networks: on training and prediction sets of examples. The sequence data $A'$ was obtained by performing a $260 \to 52$ PCA dimensionality reduction on the original data $A$. Results for two X2class runs are compared with a result from Method 1 as well as with values for neural networks trained for dichotomous prediction of traditional classes $\alpha$-helix, $\beta$-sheet, and coil.

<table>
<thead>
<tr>
<th>Classification</th>
<th>$M_k$ on Train Set</th>
<th>$M_k$ on Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2Aclass</td>
<td>0.30</td>
<td>0.25</td>
</tr>
<tr>
<td>X2Bclass</td>
<td>0.31</td>
<td>0.24</td>
</tr>
<tr>
<td>X1Aclass</td>
<td>-0.51</td>
<td>-0.43</td>
</tr>
<tr>
<td>$\alpha$-helix</td>
<td>0.37</td>
<td>0.33</td>
</tr>
<tr>
<td>$\beta$-sheet</td>
<td>0.31</td>
<td>0.26</td>
</tr>
<tr>
<td>Coil</td>
<td>0.41</td>
<td>0.39</td>
</tr>
</tbody>
</table>

represents the optimal tradeoff between these different component costs.

But suppose that for a particular problem there are reasons to want a particular point on the tradeoff between component costs, a particular degree of reconstruction error, say, regardless of the costs of model and code. In this case, it will not suffice to optimize the log likelihood and let the algorithm and data dictate the optimum: in such a case, it may be advantageous to explicitly separate, represent, and weight the different component costs, and then optimize this weighted objective function.

Again ignoring the cost of model parameters, we can usefully divide the total cost of an encoding into separate complexity and distortion terms. The complexity, $C$, corresponds to the cost of encoding the data in terms of the model. In the case of gaussian models as used above, a reasonable complexity formulation is $C_{ik} = C(\bar{x}_i, C_k) = -\log \pi_k$ expected per-example code cost, for each datapoint $\bar{x}_i$ and its class (cluster) $C_k$, where $\pi_k$ is the probability of that class as observed over the data set. This definition reflects the intuition provided by the MDL communications paradigm — that each data point is communicated by sending its class label as its encoding. The distortion cost $D$ is, for our gaussian classes, $D_{ik} = D(\bar{x}_i, C_k) = [\sum_{j}(x_j - \mu_{kj})^2]^{1/2}$. With these definitions, one can define and optimize an objective function of the form

$$E = \sum_i \sum_k p_{ik} \lambda_C C_{ik} + \lambda_D D_{ik}$$

where $\lambda_C, \lambda_D \in \mathbb{R}$, $\lambda_C + \lambda_D = 1$, and where $p_{ik} = \text{Prob}(\bar{x}_i \in C_k)$.

For our joint ($A, Z$) modelling, the cost terms may be further decomposed. First, the dis-
tortion costs may be divided into $D^A$ and $D^Z$ components, which may be differentially weighted. Second, the complexity costs may now be split into within-$A$, within-$Z$, and between-spaces complexity costs, the last of which relates directly to predictability:

$$- \sum_h \lambda_1 \log \pi^A_h - \sum_k \lambda_2 \log \pi^Z_k - \sum_h \sum_k \lambda_3 \log \pi_{hk}.$$

The setting of the $\lambda$ coefficients and subsequent optimization of the weighted objective functions allow one to aim for particular kinds of representations. It allows one potentially to trade some quality useful in later levels of prediction for higher primary-to-secondary predictability, for example. We note here that a very general complexity versus distortion tradeoff in clustering can be analyzed and exploited in contexts other than the mixture of gaussians model. Buhmann and Kuhnel (Buhmann and Kuhnel, 1993) present a general treatment and review several different complexity and distortion measures suitable for different domains and purposes. Also, many of the same issues arise in the information-theoretic analyses of data compression and of communication channels: see, for example, the treatment of “rate distortion theory” in the now-classic information theory reference by Cover and Thomas (Cover and Thomas, 1991).

The major problem with Method 2 seems to be the misapplication of the vector-space and gaussian representation of the amino acid sequence data in $A$. Weighting the component costs differently cannot save the method, as is, from this flaw. A way must be found to overcome this problem of $A$’s representation while still optimizing a function that rewards low $Z$-distortion and good $A \rightarrow Z$ predictability. Method 3 represents an attempt to do just that.

### 3.5 Method 3: Conditional Mixture Models — Mixture of Competing Experts

#### 3.5.1 Intuition and Background

Method 1 and Method 2 both develop secondary structure classes by finding class structure in $A$ and $Z$ such that the classifications “agree” (subject to other constraints). Therefore much of the computational effort goes into some kind of clustering within the space $A$ of subsequences. even though the kind of motif we seek — secondary structures — is defined only with respect to clustering in $Z$. Moreover, it has been observed by several researchers that clustering in sequence space is generally harder than clustering within structure space. Therefore, one wonders.
shouldn’t one try to form clusters in $Z$ that are somehow conditional upon the items in $A$, but without having to work to find clusters per se in $A$? This section explores one way of doing exactly that.

Method 3 is based upon an extension to the gaussian mixture modelling approach of Method 2. In the new method, rather than simply estimating $Z$-class parameters from the $Z$ data being modelled, we allow these parameters to be a function of the data in $A$. Different parameter values become associated with different $\tilde{a}_i$ inputs. Different parameter values cause different output of the system, that is, a different classification decision for a structure fragment $\tilde{z} \in Z$.

The general learning model employed here is known as the “adaptive mixture of experts”, or “mixture of competing experts”, as defined and explored by several researchers (Nowlan, 1991; Jacobs, Jordan and Barto, 1990; Jacobs et al., 1991). In this machine learning paradigm, one employs a set of networks, in which one network is the “Gating net” used for directing and interpreting the activities of a set of “Expert nets”.

One way to view the mixture of experts method is to see first of all that basic mixture modelling is a particular kind of (soft) competitive learning that performs probability density estimation, estimating $p(X|M)$ for some dataset $X$ and model $M$. In our problem of protein secondary-structure motif discovery, this becomes an estimation of the joint probability density $p(A, Z|M)$. In contrast, the basis of Method 3 is the estimation of the conditional probability density of the structure data given the sequence data and the model: that is, $p(Z|A,M)$. Like that of Method 2, the theoretical foundation for Method 3 may be understood in terms of the MDL communication paradigm. In this case, instead of the Sender encoding and sending both $A$ and $Z$ to the Receiver for the latter to reconstruct and read, the Sender and Receiver both have $A$, and the Sender has to send $Z$ to the Receiver. The question is, how small can we make the total cost (complexity and distortion) of encoding and sending $Z$ if we are allowed to use whatever knowledge of $A$ we have? And the goal of an objective function based on this conditional probability formulation is to find the minimum total cost encoding of $Z$ given $A$.

It is helpful to consider the communication scenario in further detail. How exactly might the possession of $A$ help one to encode and decode $Z$? Suppose the Sender trains a pair of networks: a left-hand network that for each $\tilde{a}_i$ outputs an $m$-tuple $\pi_{1i}, \pi_{2i}, \ldots, \pi_{mi}$; and a right-hand network that performs a mixture of gaussians clustering in $Z$, using $m$ gaussian clusters. The Sender sends the weights (and mean and variance parameters) of these networks — at a
small one-time cost — to the Receiver. Now, for each $\tilde{a}_i \in A$, the Receiver (who has rebuilt the trained networks) uses the left-hand net to obtain the $\pi_{1i}, \pi_{2i}, \ldots, \pi_{mi}$ and chooses the highest $\pi_{ki}$. This corresponds to the $k$th class in $Z$. The Receiver has the gaussian parameters, so knows the class centroid and standard deviation from the centroid for this $k$th class, and so knows an approximation to $\tilde{z}_i$. He has used $\tilde{a}_i$, and the association between the left-hand and right-hand networks, to construct $\tilde{z}_i$, within some error. The Sender must also send him the reconstruction bits to make up for the error. This scenario is a simplification — for example, it is not optimal to turn the $\pi_{1i}, \pi_{2i}, \ldots, \pi_{mi}$ vector into a single hard classification decision — but it conveys the important ideas.

Another way to understand the mixture of experts model is in terms of task decomposition. In this view, a trained system provides a two-stage solution to the problem of outputting an appropriate response $y$ to an input $x$. In the first stage, the system (the left-hand network, say) reads the input and delegates the task to the most appropriate of the Expert subnetworks (in the right-hand network, in terms of our earlier formulation). In the second stage, the one or more chosen experts process the input and produce the output. Just as gaussian mixture modelling is a “soft” form of competitive learning, one can use conditional gaussian mixture modelling as a soft form of competitive associative learning. Hence the task delegation is more often really an output interpolation — the output of the whole system is typically a weighted average of the outputs of each of the experts.

The task decomposition use of mixtures of experts may employ complex Expert networks — each Expert might itself implement a mixture model, or might comprise a multi-layered feed-forward network. In our formulation, for the protein structure classification domain, each Expert net represents just one gaussian model, i.e., one class, one gaussian unit rather than a whole network. (See Figure 3.6.)

The performance task of the mixture of experts network has been described, but what about the learning task? The goal of learning, as implemented through optimization of the objective function defined below, is two-fold (reformulating in terms of our $A, Z$ instead of $X, Y$):

1. The parameters of the Expert networks should be adjusted so that the responses, over all of the $\tilde{z}_i$, of the $m$ expert-net outputs (gaussian mixture units) of a network comprise an acceptable MDL $m$-class modelling of $Z$.

2. The parameters (weights) of the Gating network should be adjusted so that the $m$ outputs
of the Gating network, over all of the \( \tilde{a}_i \), provide accurate assessments, or predictions, of the relative expertise of the experts. For example, the “winning” Gating net output should correspond to the “winning” Expert net output.

The two goals outlined above suggest the motivation for the use of Method 3: It may be possible to produce a secondary structure classification that is both informative with respect to local tertiary structure and predictable from local amino acid sequence: that is, a classification close to both within-Z and between-spaces optimality.

### 3.5.2 The Objective Function

The objective function to be minimized is

\[
E = - \sum_{i=1}^{M} \log \sum_{k=1,2} \pi_k(\tilde{a}_i)p(\tilde{z}_i|\tilde{\theta}_k)
\]

where the gaussian probability densities are computed exactly as in Method 2. Dimensions of the parameter vectors are assumed independent given the class \( k \): therefore

\[
p(\tilde{z}_i|\tilde{\theta}_k) = \frac{\exp\left(-\frac{1}{2} \sum_j^{N_w} \frac{\left|\tilde{z}_i - \mu_{z,j}^{k}\right|^2}{\sigma_{z,j}^{k}}\right)}{(2\pi)^{N_w/2}(\prod_j^{N_w} \sigma_{z,j}^{k})^{1/2}}.
\]

The \( \pi_k(\tilde{a}_i) \) values, taking the place of the mixing proportion \( \pi_k \) used in Method 2, are produced by the Gating network output units. In our experiments the Gating network is a feedforward network with three layers: input, hidden, and output. Each hidden layer unit computes a sigmoidal function of the weighted sum of its inputs. Each output unit computes a softmax function (Bridle, 1990) of the weighted sum of its inputs. \( \text{softmax}_j(x_j) = \frac{\exp(x_j)}{\sum_{h=1}^{k} \exp(x_h)} \), where \( x_j \) is the total input to the \( j \)th unit and \( x_h \) is the total input to the \( h \)th unit.

Figure 3.6 illustrates the use of the objective function based on conditional mixture of experts approach to density estimation.

### 3.5.3 Learning Rules and Prediction Rules

**Prediction of structural class given sequence input:**

Method 3, like Method 1, uses two different networks for the \( A \) and \( Z \) inputs: devising a way to predict a \( \tilde{z} \) given an \( \tilde{a} \) is therefore fairly straightforward. Because we again use the
Conditional $A \rightarrow Z$ Density Modelling with Adaptive Mixture of Experts

**Figure 3.6:** Illustration of the neural network implementation of conditional mixture modelling used in Method 3. The left-hand network (Gating Net) is a feed-forward network with sigmoidal activations in the hidden layer and softmax activations in the output layer. It receives input from $A$. The right-hand network (Experts Net) implements a mixture of gaussians model of latent class structure in $Z$. For simplicity, the input layers are shown to have $N_w$ units each: however, the actual input dimensions are $20N_u$ for $\tilde{a}_i$ (for the unary encoding of each amino acid) and $4N_w$ for $\tilde{z}_i$ (for the $(\cos \phi, \sin \phi, \cos \psi, \sin \psi)$ encoding of each residue). In the experiments performed thus far, $N_w = 13$. On each iteration, on the $\tilde{z}_i$ part of each input example, and for each class $C^Z_k$, the class-conditioned probability $p(\tilde{z}_i|\theta_k)$ produced by the right-hand net is combined with the $\pi_k(\tilde{a}_i)$ produced by the left-hand net and normalized to produce the responsibility values (not shown). The free parameters in $\theta_k$ (means and variances) and the weights of the left-hand network are updated by conjugate-gradient optimization based on the $\frac{\partial E}{\partial \theta}$ terms. The trained two-network configuration may be employed by using the Gating net outputs to predict the classification decisions made by the Experts Net.

Matthews correlation as the final arbiter of agreement, the network outputs must be turned into hard classification decisions. For the Gating net, this means taking $c_a = \max_k \pi_k(\tilde{a})$ as the $A$-prediction. One way to extract the "answer" given by the Experts net is to use $c_z = \max_j r_j(\tilde{z})$. If a new sequence fragment $\tilde{a}_{new}$ from a sequence with unknown structure is presented to a system trained by Method 3, then the secondary structure prediction made by the system is $c_a$. Predictability of a classification produced by a trained system can therefore be measured by $M_k(c_a, c_z)$ computed over all cases. As before the measure of how well the secondary structure prediction system generalizes to new data is taken by computing the $M_k(c_a, c_z)$ over a testing set, separate from the training set.
Training the parameters of the model:

The mixture of experts system is a combination of a relatively standard multi-layer feed-forward network and a gaussian mixture model with adaptive means and variances: the update rules for connection weights and other parameters reflect this.

As before, the learning rules are derived from the gradient of the objective function. One has therefore to consider the following quantities (for each training example $\tilde{x}_i = (\tilde{a}_i, \tilde{z}_i)$ and each class $C_k$ and each dimension $j$):

$$\frac{\partial E}{\partial \mu_{kj}} = \frac{r_{ik}(z_{ij} - \mu_{kj})}{v_{kj}}$$

$$\frac{\partial E}{\partial v_{kj}} = \frac{1}{2} \frac{r_{ik}(v_{kj} - (z_{ij} - \mu_{kj})^2)}{v_{kj}^2}$$

$$\frac{\partial E}{\partial w_{kh}} = \frac{\partial E}{\partial \pi_k(\tilde{a})} \frac{\partial \pi_k(\tilde{a})}{\partial w_{kh}}$$

$$= \frac{-r_{ik}}{\pi_k(\tilde{x}_i)} \frac{\partial \text{softmax}_k(\sum w_{kr}a_r)}{\partial w_{kh}}$$

(3.27)

(3.28)

3.5.4 Experiments and Results

A set of 80 simulation runs were performed with the mixture of experts networks on the original $X = A \times Z$ data described in the section on Method 1. In the first set of 40 runs, the $A$ network (gate network) employed 3 hidden units, and in the last 40 runs, 4 hidden units were used. Therefore the network architectures were $(260 \rightarrow (3or4) \rightarrow 2)$ and $(52 \rightarrow 2)$ for the Gating and Expert nets, respectively. Each run was begun from a random initial state, with weights in the Gating net initialized by a gaussian distribution with mean 0 and variance 0.2, 0.3, or 0.4, and the Expert network gaussians were initialized with means centered on 0.5 and variances 0.3. A weight-cost term of 0.002 was used in the Gating network, as a regularization aid. These hidden layer sizes and initialization values were chosen heuristically by the investigators and subjected to limited “trial and error” comparison. Both heuristic suggestions and statistical methodologies for choosing neural network architectures can be found in the literature (Baum and Haussler, 1988; Le Cun, 1989; Geman, Bienenstock and Doursat, 1992).
Conjugate gradient optimization was used to train the gate-and-experts configuration, and in each run the training was continued until convergence (or until failure of the line search procedure to accurately find a step in a descent direction). Such a training regime might be expected to over-train the networks — permitting good performance on the training set and poor generalization to the test set. However, this did not appear to be a major problem because, first, the training may have benefited from the natural regularization arising from the large data set and the use of weight cost penalties; and, second, even if over-training affected the continuous values in the gate net output and Expert net responsibility terms, our measure of predictability is based on rounded, discrete values. That is, the difference between training a parameter to 0.9999 instead of 0.8733 is not very important when any value greater than 0.5 is rounded to 1 anyway.

It must be noted that the dynamics of learning in these networks is quite complex and interesting, and training them well seems to require a nontrivial degree of interaction and direction. For example, if both the Gating and expert nets are allowed to adapt unhindered and in parallel, particular kinds of poor locally optimal states can result. Small deviations from equality in the initial Gating net outputs can cause one of the Expert modules to be "locked out" of further competition and adaptation. That is, if one of the experts is predicted by the gate to "lose the first round", it can be forced to continue "losing" every round thereafter. This problem can be avoided by "freezing" the weights of the Gating net for a few initial iterations, thereby allowing the experts net to come to a more balanced and accurate initial model of the Z data. Taking the reverse action — freezing the experts while adapting the gate net — also seems to be required at one or more points in the training, in order to force the gate outputs to better model their respective Expert responsibility terms. In our experiments, some apparently successful strategies were found by trial and error, but these strategies are not believed to be optimal, and further exploration is certainly indicated.

As in the Method 2 experiments, a significant portion (one fourth) of the Method 3 simulation runs became stuck in bad local minima corresponding to a one-class trivial classification. The other three fourths of the runs produced nearly identical objective function values, and, as will be seen in Section 3.6, nearly identical classifications (and very similar to those produced by Method 2 as well).

The predictability results from two of the best runs are shown in Table 3.6. It can be seen that the predictability index on sequence fragments not found in the training set are comparable
to those for traditional prediction of the traditional secondary structure class \(\alpha\)-helix. The values are below the best results obtained with Method 1.

Table 3.6: Mathews correlation values \((M_k)\) between sequence and structure classifications, as measured by Gating net outputs and Expert net responsibilities, respectively: on training and prediction sets of examples. Results for two X3class runs are compared with a result from Method 1 as well as with values for neural networks trained for dichotomous prediction of traditional classes \(\alpha\)-helix, \(\beta\)-sheet, and coil. The novel classes discovered with Method 3 are evidently not as predictable from sequence as X1AClass derived from the first method described in this chapter, but they are as predictable as the standard \(\alpha\)-helix class.

<table>
<thead>
<tr>
<th></th>
<th>(M_k) on Train Set</th>
<th>(M_k) on Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>X3Aclass</td>
<td>0.44</td>
<td>0.34</td>
</tr>
<tr>
<td>X3Bclass</td>
<td>0.39</td>
<td>0.33</td>
</tr>
<tr>
<td>X1AClass</td>
<td>-0.51</td>
<td>-0.43</td>
</tr>
<tr>
<td>(\alpha)-helix</td>
<td>0.37</td>
<td>0.33</td>
</tr>
<tr>
<td>(\beta)-sheet</td>
<td>0.31</td>
<td>0.26</td>
</tr>
<tr>
<td>Coil</td>
<td>0.41</td>
<td>0.39</td>
</tr>
</tbody>
</table>

It was hoped that Method 3 might produce classifications that, if somewhat less predictable than those obtained with other methods, represented lower MDL costs. The conditional log likelihood values corresponding to the objective function definition are not very helpful in comparing Method 3 to Methods 1 and 2 in this respect. This is because the \(\pi_k(\tilde{a}_i)\) factor in the \(\pi_k(\tilde{a}_i)p_k(\tilde{z}_i)\) terms represents the degree to which the Gating net succeeded in modelling the experts net. If this between-networks correspondence was poor, it should be and is measured by the predictability index: it should not be permitted to skew the assessment of how well the experts net modelled the \(Z\) data. Therefore, one can formulate an alternative MDL measure for the Expert nets and the \(Z\) classification, the log likelihood calculation with an empirical mixing proportion \(\pi_k\) used instead of the one conditional upon the \(A\) data. This heuristic and approximate MDL measure is used in comparing Method 3 with the other methods in Section 3.6 below.

3.5.5 Possible Extensions

The classes discovered by Method 3 thus far are reasonably predictable, and possess other interesting qualities as noted below in Section 3.6. However, these classes are not more predictable than the best of the traditional (dichotomous) classes, nor as predictable as those produced with Method 1. It is possible to employ a strategy described in the Extensions to Method 2, a strategy of separating and weighting the different component costs in the objective function.
One could do this with an aim at more predictable classes from competing experts networks. More generally, further thought and experimentation should be carried out in the search for other objective functions that share with Method 3 the idea of using different representations for the A and Z data while bringing these two associated datasets under the same probabilistic framework.

3.6 Comparison of the Three Methods

Three different but related methods have been investigated for their ability to produce protein secondary structure classes that are predictable from amino acid sequence and that embody a particular optimal encoding criteria that, we believe, may contribute to their usefulness in further stages of tertiary structure prediction. Both of these goals for the derived classifications derive from a general view of secondary structure as an intermediate representation language between primary and tertiary structure, and as a necessary middle stage in full protein structure prediction. Simply put, a good secondary structure language \( \{y_k\}_{k=1}^m \) should be readily predictable from sequence data A, and each \( y_k \) should carry the maximal amount of structural information (from Z) per unit of code-length.

The results on predictability were presented in the Results section for each of the three methods. The first method, designed with predictability (and only predictability) in mind, has thus far proven superior in this respect to the other methods; moreover, it has produced novel secondary structure classes that are more predictable than the standard secondary structure classes. (However, it was shown how weighted objective functions could potentially be added to Methods 2 and 3 to produce more predictable classes.)

Two important analyses remain to be presented on the classifications produced by the three methods. First is the measurement of similarity between the best classifications produced from each of the three methods, as well as the similarity between our novel classifications and the traditional classes. The latter has been done for the most predictable classes, discovered with Method 1; further insights can be gained by doing the same for some of the other classifications. Second is an attempt to measure and compare the MDL costs of the classifications.

For the non-trivial classifications produced by our methods (that is, ignoring the classifications in which all examples were placed into a single class), a somewhat surprising result was found: Method 2 and Method 3, across many runs starting from different random initial states, consistently discover the \( \alpha \)-helix motif.
As Table 3.7 illustrates, for two of the classifications discussed in the Results sections for Methods 2 and 3, the best classifications in terms of predictability and in terms of our objective functions, agree with the traditional α-helix class with a Matthews value \(|M| > 0.80\). The correspondence is perhaps even better than the Matthews value indicates. If one views the testing set examples as an ordered list — respecting the residue orders of the original sequences — one sees “runs” of particular structures, as expected: that is, for example, a sequence of 10 helix residues might occur, followed by a sequence of 15 coil residues, another 12 helix residues, and so on. These “runs” of real helix and of our discovered helix-like class match up exactly: the classifications typically differ only on the edges of such runs, on the first two or last one residue in a run. Not unexpectedly, these two novel classes, each of which agrees highly with α-helix, agree highly with each other (after accounting for the negative sign of the correlation).

Moreover, all of the non-trivial simulation results for each of Method 1 and Method 2 were similar to each other, with \(|M(\text{result}_i, \text{result}_j)| > 0.68\) for all \(i, j\).

Table 3.7: Matthews correlation values, for class X2Aclass produced by Z half of joint density modelling network (Method 2) and for X3Aclass produced by structure portion (experts net) of conditional density modelling network (Method 3), as measured against each other and against traditional classes α-helix, β-sheet, and coil.

<table>
<thead>
<tr>
<th></th>
<th>X3Aclass</th>
<th>α-helix</th>
<th>β-sheet</th>
<th>coil</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2Aclass</td>
<td>-0.91</td>
<td>-0.82</td>
<td>0.28</td>
<td>0.51</td>
</tr>
<tr>
<td>X3Aclass</td>
<td>0.81</td>
<td>-0.27</td>
<td>-0.51</td>
<td></td>
</tr>
</tbody>
</table>

Computation of such correlations also provides further evidence of the inefficacy of using the Method 2 representation of the \(A\) sequence data. The high correlation of the Method 3 classes with helix was observed for the \(Z\) part of the joint density network: if the same analysis is made for the output of the \(A\) component of the network, the results are less interesting, as can be seen in Table 3.8. The numbers suggest that the gaussian representation of the \(a \in A\) as used in Method 2 produced mostly noise which the \(Z\)-component was forced to overcome.

Table 3.8: Matthews correlation values, for class X2Aclass produced by \(A\) half of joint density modelling network (Method 2) as measured against traditional classes α-helix, β-sheet, and coil.

<table>
<thead>
<tr>
<th></th>
<th>α-helix</th>
<th>β-sheet</th>
<th>coil</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2Aclass (sequence)</td>
<td>-0.23</td>
<td>0.03</td>
<td>0.17</td>
</tr>
</tbody>
</table>

How do the classes produced by Methods 2 and 3 relate to the most predictable classes produced with Method 1? Class X2Aclass from Method 2 has a correlation of \(M = -0.64\) with class X1Aclass from Method 1. Typical classes from Method 3 are slightly less similar to
the best produced by Method 1. For example, X3Aclass and X1Aclass correlate at a level of $M = 0.61$. The reader will recall that the novel classes from Method 1 are observed to have similarity to $\alpha$-helix as well, though do not match it as closely as the classes from Methods 2 and 3. The X1Aclass differences from helix are significant, because that class was found to be more predictable than the standard helix class.

Given that Methods 2 and 3 tend to find the same classes (slight modifications of the $\alpha$-helix class), it is not surprising that the MDL costs computed on the Z data for the classes produced by each of the two methods are nearly identical. In the simulation runs that produced non-trivial classifications, the mean values for final per-example log likelihood were: Method 2, 53.5; Method 3, 51.9. We emphasize the very rough and tentative nature of these estimates, given the reliance on several useful but precarious assumptions about our learning architectures and protein representations. Much more research is needed.

It is interesting that two self-supervised learning methods, receiving no explicit teaching signal or information about standard structure motifs, "trying" only to optimize a data-encoding property, and starting from random points in classification space, would converge repeatably to discovery of the alpha-helix motif. This particular result says more, perhaps, about the "naturalness" of the helix motif and of the original insights of Pauling and colleagues than it says about the strengths or weaknesses of our methods.

It is also interesting that in the experiments with the method designed primarily for the discovery of highly-predictable classifications, the best resulting motifs are more predictable than any of the three standard secondary structure classes: moreover, to the extent that the novel, predictable classes resemble any of the standard classes, they most closely resemble $\alpha$-helix.

Further work should investigate in some depth the degree to which such results depend on any of the initial assumptions our investigations have made. For example, we used a motif length (window size) of 13 residues, a number found empirically to be optimal for prediction of the standard classes (Qian and Sejnowski, 1988). What sorts of different motifs might be discovered if smaller or larger lengths of input fragments are used? Likewise, the use of $m > 2$ classes might force the networks to discover significantly different classifications. The next section, the last in this chapter, discusses how input representations and other methodological assumptions affect the qualities of derived structural motifs.

Another small but potentially important limitation of our chosen input representations is our separate encoding of the $\Phi$ and $\Psi$ angles. In practice, these parameters are correlated, and a
better representation could exploit this fact and thereby lower the number of degrees of freedom in the learning. For example, a von Mises circular distribution can be used instead of simple gaussians (Dowe et al., 1996). Our representations used twice as many parameters as needed, if viewed from this perspective. Of course, there are almost certainly correlations between nearby positions within subsequences of length 13 (or whatever), too, and a truly principled Bayesian methodology would have to handle this as well.

We should also note that various kinds of post-processing can be used after local secondary structure prediction methods, including after our own. Such processing can bring to bear global constraints and patterns in order to “clean up” the comparatively noisy local predictions (Rost et al., 1995).

3.6.1 The Use of Artificial Neural Networks

We emphasize here that the focus of this chapter is not on artificial neural networks or their capabilities per se. The neural network serves as a convenient computational formalism and architecture for many statistical modelling and estimation methods, and we believe that neural networks are best employed and understood within a larger statistical framework (Elder and Pregibon, 1996; Hastie and Tibshirani, 1990; MacKay, 1992).

3.7 Comparison with Other Local Structure Motifs

In order to assess the algorithms and classifications presented in this chapter, it is necessary to consider them within the larger context of protein structural motifs.

A reading of the relevant research literature in protein analysis suggests six broad criteria by which to measure protein structure motifs, and, by extension, the methods used to define and discover them.

Protein Local Structure Motif Criteria:

1. Predictability is the degree to which one level or facet of protein structure or function may be predicted from knowledge of another. For the local structure motifs we have designated as “secondary structure”, predictability is the ability to accurately predict secondary structure classes from amino acid sequence. We have chosen to measure this kind of predictability in terms of the Matthews correlation between “class guessed by looking at sequence” and “actual class defined by local tertiary structure information”.
2. *Predictive Utility* is a more nebulous concept. If one takes the view of secondary structure as an intermediate-level encoding, between primary structure (sequence) and tertiary structure, as expounded in Chapter 2, Section 2.9, then predictive utility ought to be some measure of the gain in accuracy in predicting tertiary structure with a particular encoding, as compared with prediction using other possible encodings. We have identified an MDL-derived $p(Z|M)$ cost as one reasonable component of predictive utility, because an MDL-optimal encoding of local tertiary structure fragments tends to preserve more structural information per unit of code-length. A second aspect of predictive utility might be the ability to represent the confidence level or degree of uncertainty in secondary structure prediction. In our framework, such information might be derived from the degree of distortion in the original $Z$-cluster corresponding a given class label, or from the $A \rightarrow Z$ predictability of the classes found by the motif-discovery procedure. Other factors in predictive utility are discussed in subsequent subsections.

3. *Naturalness*, or the equally unwieldy word "intrinsicism" means the degree to which a motif captures some essential biochemical or evolutionary properties, or some essential class structure in the space of protein sequence or structure fragments under consideration. Some clustering methods, for example, are infamous for finding ersatz clusters in uniformly distributed data. Other clustering methods produce results very dependent upon their starting point. Such results are to be avoided.

4. *Ease of discovery* refers to the computational complexity and data complexity of the methods required to discover the motif.

5. *Systematicity* is the degree to which a motif discovery method is derived from explicitly-stated principles and the degree to which the method can repeatably be applied to diverse data and produce consistent results.

6. *Intelligibility* refers to the ease with which researchers and practitioners of protein science can understand a given structure motif and can incorporate its information into their own work. Many factors affect intelligibility. For example, a discovered structure class that contained $\frac{1}{3}$ traditional $\alpha$-helix, $\frac{1}{3}$ traditional $\beta$-sheet and $\frac{1}{3}$ "coil" is harder to explain than one which correlates almost perfectly with $\alpha$-helix. Also, for example, a motif expressed in first-order logic with terms for well-known biochemical aspects such as amino acid names and dihedral bond angles is easier to understand than a motif represented only in a set
of several hundred neural network connection weight values. Further aspects of motif intelligibility are discussed below.

Many aspects of motif discovery methods — including the algorithms, the chosen dataset(s), the representation of input sequences and structures, the number of motifs/classes sought — profoundly affect the degree to which the resulting motifs satisfy the above criteria. Some of the most important of these aspects are discussed in the subsections below, with reference to our methods as well as to other methods prominent in the literature.

It is useful to adopt Conklin's (1995) definitions of motif-types in discussing these issues. In terms of his designations, the work presented in this chapter aims at the discovery of structure-sequence motifs, that is, local tertiary structure motifs (classes) that are tagged with representations of the amino acid sequence patterns associated with the structure motif. Sequence-structure motifs are sequence motifs annotated with information about the secondary, super-secondary, or tertiary structures to which they are believed to correspond. There are also of course sequence motifs and structure motifs.

Sequence motifs are typically derived by first aligning a set of sequences and then abstracting the amino acid residues found to occur at each position. (For example, if only instances of the amino acids $A = \text{alanine}$ and $G = \text{glycine}$ occur at a position, then the abstraction might be represented as "$A"$ or "$G"$ or "$\text{SMALL SIDE-CHAIN}$".) Sequence motif discovery procedures often take a primarily symbolic approach. Structure motifs, on the other hand, are most often found using numerical clustering techniques. The abstraction in such approaches occurs as a natural result of placing individual exemplars into clusters or classes based on some similarity measure defined over the numerical features.

Our Methods 1 and 2 are seen to produce sequence-structure or structure-sequence motifs, depending on one's viewpoint: Method 3 is aimed squarely at the discovery of structure-sequence motifs. This accords with our major goal of broadening the concept of secondary structure and then finding novel secondary structure classes. Structure-sequence motifs have come into the fore in recent years, as interest in sequence-threading and the "inverse folding problem" have grown. Perhaps, too, others share our hypothesis that better alternatives to standard secondary structure classes may be waiting to be found.

Several prominent projects in the discovery of structure-sequence motifs are discussed below.
3.7.1 Use of Both Sequence and Structure Data

The discovery of sequence motifs has been, along with its sister goal of multiple sequence alignment, the mainstay of computational molecular biology from the beginning. A central hypothesis has been that similarity of sequence implies similarity of structure and function, and evolutionary conservation of sequence implies conservation of structure and function. Thus it is in a sense difficult to find any work on sequence motifs that is not also really work on sequence-structure motifs. Much of the vast literature on consensus sequences, sequence Profiles, and Hidden Markov Modelling of protein families describes the discovery of motifs over sets of sequences that are already known to correspond to one particular overall tertiary structure. But where researchers have sought to define family-independent subsequence motifs supposed to carry important structural information, they have not necessarily succeeded. In a study performed by the Rooman-Wodak group (Rooman and Wodak, 1991), 11 out of a set of 12 sequence-structure motifs claimed to be predictive of secondary structure were found not to be.

This brings us to structure-sequence motifs. A few researchers have taken their approach to discovery of structure motifs and have added sequence annotation, or computed sequence-structure correlation, after the fact.

Unger et al. (1989) used a k-nearest-neighbours method to find clusters in the space of protein backbone fragments of length 6 residues. They tabulated the frequencies of amino acid types at every position, producing a sequence motif for each of their $m \approx 100$ structure motifs.

Rooman et al. (1990b) produced a physico-chemical properties motif for each of the 4 – 10 structural classes they discovered in different runs of a hierarchical clustering of fragments of length 6…10.

Zhang and Waltz (1993) used auto-encoder neural networks, implementing a variant of k-means clustering (Boulard and Kamp, 1987; Buhmann and Kuhnel, 1993) on structure fragments of size 7, and then tested the $\chi^2$ significance of the association of their 23 local structure classes with particular amino acid combinations.

There are two major limitations to these methods. First, of course, the sequence information is incorporated into the structure motifs after the latter have been defined. This approach is not designed to, and is not likely to, produce structure classes predictable from amino acid sequence.

Second, as noted by Conklin (1995), in the methods outlined above, each structure motif may be associated with only a single sequence motif; there is no provision made for associating a
structure motif with a *disjunction* of different sequence motifs (except in the narrow disjunction implicit in the abstraction of several very similar sequences into a more abstract motif).

Conklin's method and our own methods get around both of these limitations, taking strikingly different routes. Conklin represents both sequence and structure objects in the symbolic format of a *spatial description logic*, a restricted first-order logic used to describe and manipulate *concepts*. Motif discovery in his system occurs through similarity-based clustering (structured concept formation) of combined sequence-structure representations. The sequence-structure predictability is built into the discovery process, and is measured *a posteriori* with $\chi^2$ tests.

The classes developed by our methods are designed to be predictable, as predictability is one of the key criteria built into the objective functions and measured after the motifs are discovered. The ability to associate more than one sequence motif with a structure motif derives from the fact that in Method 1 and Method 2 one may define $m_A$ $A$-classes and $m_Z$ $Z$-classes for $m_A > m_Z$. In Method 3, no classes *per se* are imposed upon the $A$-data, but it should be possible to perform a clustering and feature abstraction for all the $a_i$ corresponding to $z_i \in C^Z_k$ *a posteriori* for each class $C^Z_k$.

### 3.7.2 Number of Classes, Number of Motifs

The number of different motifs, or classes, sought in a discovery procedure has important impact on both the information-theoretic aspects of predictive utility of the resulting motifs and the general intelligibility and usefulness of the motifs to molecular biologists.

First, in terms of the *distortion versus complexity* tradeoff in latent class density modelling, it is clear that more classes generally implies lower distortion and higher complexity costs. That is, the larger the number of classes and hence class centroids (exemplars, control points), the closer a given point will be to the centroid of its own class, *ceteris paribus*. But the larger the number of classes is, the more bits it takes to encode each data point in terms of its class-label encoding.

Though the MDL/communications paradigm is a somewhat artificial theoretical tool, it does reveal important practical aspects of data models. As the number of classes in a model of structure fragment data increases, a very real tradeoff becomes apparent. Each motif becomes more specific, in that it carries more detailed local structural information about a smaller set of fragments. This might make subsequent Stage 2 tertiary structure prediction easier, because structure-packing considerations are made more precise. On the other hand, there is a loss
of abstraction, a greater number of parameters to optimize in the motif discovery algorithm, a potentially greater difficulty in finding statistically significant estimates of frequencies and probabilities of motifs and features.

A growing consensus in computational molecular biology favors classes less coarse than the standard 2–5 secondary structure classes. Conklin, in his survey (1995) cites three reasons:

1. Conklin claims that there exist wide discrepancies between different methods of assigning secondary structure designations from crystallographically determined structures. This point is debatable. It appears to other observers that the Kabsch and Sander standard is both well-founded and widely accepted. However, to the extent that discrepancies do exist, one must take care that a prediction system is not just modelling the idiosyncrasies of particular structure definition rules.

2. A great number of fragment patterns tossed into the large default class “random coil” are neither random nor undefinable. Add to this the fact that different kinds of helices, and different kinds of \( \beta \)-strand configurations, can be observed, and there is a case to be made for additional subclasses of the three major classes.

3. Secondary structure packing analysis is a non-trivial task, and more accurate descriptions of local backbone structure — as ought to result from motif discovery with larger numbers of classes — can make the task much easier.

We have focused thus far on dichotomous (two-class) prediction, for simplicity of experiments and of reporting. However, there is no theoretical reason why the methods presented herein cannot be used for the discovery of greater numbers of structure classes, greater numbers of sequence classes, or both. Bayesian/MDL criteria, upon which our methods 2 and 3 are based, can also be used to help determine the optimal number of classes (Hunter and States, 1992; Dowe et al., 1996; Cheeseman and Self, 1987).

### 3.7.3 Locality: Size of Input Fragments

The size of sequence and structure fragments input to motif discovery systems is another issue closely related to the question of abstraction versus specificity. Smaller fragments imply smaller, more localized motifs. Smaller motifs mean that a greater number of them are needed to represent an entire sequence or structure, and hence a greater number of parameters are used in
latter stages of a modelling or prediction task. On the other hand, smaller motifs also correspond to more frequently-occurring patterns. and therefore problems in probability estimation are minimized.

One must also carefully consider domain-specific and goal-specific criteria when choosing fragment size: Over what lengths of sequence and of structural backbone chain do the phenomena of interest manifest themselves? For example, individual β-strands can be captured with fragments of size 6 to 12, typically but what about the turns between strands? How much information about the strand is conveyed by the nearby turns, and vice versa? How much information do different strands carry about each other? How much non-local information is necessary to determine a sequence fragment's propensity to "become" a helix or a strand or a stretch of "coil", for example?

The information-theoretic and the biophysical issues here are deep and complex. An empirical, trial-and-error approach might be reasonable in attacking this problem. A fragment size of \( N_w = 13 \) was chosen for our experiments, for both sequence and structure inputs, because that window-size was found to be effective in previous work on secondary structure prediction (Qian and Sejnowski, 1988; Kneller, Cohen and Langridge, 1990). In such earlier studies it was found that smaller windows failed to provide sufficient local contextual information for prediction of the secondary structure of the central residue in the window; for windows of length larger than 13, the marginal gains in extra contextual information were swamped by noise.

Most of the reported projects in structure-sequence motif discovery looked for motifs of size 6 to 8 (Zhang and Waltz, 1993; Hunter and States, 1992; Unger et al., 1989; Conklin, 1995). Future work with our methods ought to investigate such smaller fragments and motifs as well as seeking greater numbers of motifs. If nothing else, the use of smaller fragments will enable easier comparison of our methods with others.

### 3.7.4 Representation, Shape, and Parameterization of Motifs/Classes

Finally, perhaps the most important initial choice to be made in designing a motif-discovery method is the decision as to what kind of representations to use for motifs. The differences between some of the options are huge, as large as the traditional gulf between the "symbolic/logical" and "numeric/statistical" camps in artificial intelligence research. The differences are large and the stakes can be high, both in terms of the amount of interesting information captured by the resulting motifs and in terms of the ability for us to understand and communicate
the information.

For structure classification, numerical clustering methods dominate the field (Zhang and Waltz, 1993; Hunter and States, 1992; Rooman, Rodriguez and Wodak, 1990a). There are good reasons for this. First, structures are geometric and physical objects, and the representation of such objects in terms of vectors, angles, and chemical properties is an old and strong tradition in the physical and computational sciences. Second, the use of numeric features and statistical clustering techniques is very amenable to the use of well-defined objective functions, thus enabling a generally principled approach and the use of well-understood optimization procedures.

In sequence classification, on the other hand, a symbolic representation might be more natural. Amino acids are easily represented by discrete symbols and codes. Perhaps more important is the fact that sequences are not vectors. Naive vector representations of sequences, as used in early neural network approaches, pose the risk of losing important inter-residue relationships. These potential losses may be minimized by using the vectorial approximation only over very short distances and by using statistical models that are good at capturing higher-order correlations between residues. (Feed-forward, multi-layer networks with nonlinear units are good; gaussian mixture models with independent dimensions are bad.) Another option is to embed the sequences, and an appropriate metric over them, in a suitable non-Euclidean Minkowski space, using the kind of transformations proposed by Goldfarb (Goldfarb, 1985).

If the goal is intelligibility of derived motifs, there is no contest — logical representations are preferred. In fact, the reader may wonder whether an important problem has been glossed over in our facile equation of "classes" and "motifs", because clearly it is difficult to look at a set of hundreds of connection weights or means and variances and see anything resembling a motif. However, once a set of classes has been discovered, there is no major obstacle to finding more recognizable and descriptive motifs after the fact. The set of sequences, for example, corresponding to a particular structural class $C_k^2$, can be aligned, clustered, and so on, using standard methods, and consensus sequences can be produced.

Another virtue of both Conklin's method and our own is that sequence and structure motifs are represented the same way. Unlike the other structure-sequence motifs mentioned in this chapter and surveyed by Conklin (Conklin, 1995), our structure-sequence motifs do not inherit a "confused dual semantics" (Conklin, 1995). This feature of our methods would presumably enable them to be integrated more easily into larger, multi-level, multi-view protein analysis systems wherein many different kinds of features are used to predict other features.
3.7.5 Intrinsic versus Extrinsic Clustering Criteria

Implicit in some of the above discussion is a concept of intrinsic versus extrinsic criteria for clusters and motif discovery. In the multi-stage process and multiple levels of description that characterize PSP, as in machine vision and speech recognition, there is a tension between the versions of an intermediate representation language suggested by optimizing local, “current-level” criteria (What are the best clusters in \( \Phi \Psi \) space?) and those suggested by optimizing “next-level-up” criteria (Which clusterings produce classes that work well as primitive symbols in a tertiary structure encoding?) This is a fundamental issue not yet addressed in the computational molecular biology domain, nor, to our knowledge, sufficiently addressed in other domains (Bourlard and Wellekens, 1986a; Bourlard and Wellekens, 1986b) (though Baxter’s work (Baxter, 1994) looks promising). A comprehensive multi-level Bayesian analysis that tackles this issue, perhaps integrating methods for entropy reduction in spatial distributions with those for sequential distributions (Caianello, 1990; Storer, 1988), would be a welcome addition to the protein structure analysis toolkit.
Chapter 4

Modelling Aligned Sequences, and Discovering Higher-Order Features in Discrete Datasets

We focus in this chapter on another motif discovery problem whose solution can be applied to the task of protein structure prediction.

Motif discovery reduces to the detection, selection, and weighting of features to be used in the motif; and the problem at hand is the detection of a particular kind of higher-order feature that can be found in sets of aligned protein sequences. These features are \( k \)-tuples of correlated, possibly non-contiguous (sequence-distant) residue positions across structurally and evolutionarily related protein sequences. We justify the study of these particular features on the basis of their usefulness in: (1) the discovery and elucidation of important structural and functional constraints on protein sequences, and (2) the creation of classification and matching systems more sensitive than currently popular methods that do not properly capture long-range interactions, such as Profile analysis (Gribskov et al., 1988; Gribskov, McLachlan and Eisenberg, 1987a) and Hidden Markov Model analysis (Krogh et al., 1994).

This motif discovery problem is also of wider interest because it may be seen as an instance of a more general class of problems in modelling and drawing inferences from large databases of discrete records (Haussler et al., 1992; Miller, 1993; Paturi, Rajasekaran and Reif, 1995; Willett, 1987). We identify three basic kinds of probabilistic modelling — full distribution modelling, hypothesis testing, and data mining — and suggest that the latter deserves more attention in
the domain of protein sequence and structure analysis, among many others.

The novel data-mining method of correlation detection presented herein takes advantage of the discreteness of protein sequence representations, a simple sampling and binning trick, and an appropriate set of probability distributions and hypothesis tests, in order to circumvent the combinatorial explosion faced by more standard, direct means of apprehending very high order joint frequency distributions. Our approach is best viewed as an initial filter for finding highly-correlated variables, with some quantifiable degree of confidence. It can be used in conjunction with any of several more comprehensive modelling methods (Miller, 1993; Katz, 1987; Ku and Kullback, 1969; Lewis, 1959; MacKay, 1994; MacKay, 1994; Bahl, Jelinek and Mercer, 1983) — our method to discover the plausible correlations, and the other method to further verify them build them into a model of the dataset.

The organization of this chapter is as follows. We first introduce the problem of detecting higher-order features in aligned sequence data, and discuss its importance in the modelling and prediction of protein structures. The connection to other database modelling applications is discussed. We review previous approaches to the problem, with emphasis on methods used in the macromolecular sequence and structure domain. These approaches include consensus sequences and profiles, Hidden Markov Models (HMMs), other latent variable models with Bayesian training algorithms, and the explicit calculation of pairwise column-correlation statistics. Then our novel method is described, along with analyses of the time and space complexity of the algorithm and of the probability of various kinds of error in detecting interesting features. Test results and analyses of the method on some artificial and actual protein datasets are presented.

4.1 Motivation: Correlations between Residue Columns

Given a set of aligned sequences, such as shown below, representing a protein family or superfamilly, one can begin to characterize the family by finding and selecting its representative features.

4.1.1 First-Order Features

The first (and often only) step in a typical feature-selection process is the analysis of features corresponding to individual residue numbers (a.k.a. "positions" or "sites"); these are primitive
Table 4.1: Seven sequences of the Crambin (1crn) protein family, aligned. The data comes from the HSSP database. We consider such aligned sets of sequences as $M \times N$ tables of attribute values. Each column $c_j$ corresponds to an amino acid residue and may be considered as a set of outputs of a discrete random source, or as i.i.d. samples from a discrete random variable $v_j$. The values, or states, $a_i$ are particular amino acid symbols: and in this case $M = 7$ and $N = 46$. Typically, both $M$ and $N$ are an order of magnitude larger, though $N$ can be as large as 1000 for proteins and 20,000 for RNA.

\begin{table}[h]
\centering
\begin{tabular}{cccccccccccc}
1 & 2 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 46 \\
$S_1$: & TTCCPSI V ARSNFNVCRLPGETPAI CATYTGC I HPGATCPGDYA N \\
$S_2$: & - SCCRNTW A RNCYNVCRLPGT I E I C A K K CD CKI S G T C P S D Y P K \\
$S_3$: & - SCCPTTA ARNQYN I CRLPGTR PRPVCAALS G CKI S G T C P S D Y P G Y - - \\
$S_4$: & - SCCPNTT GRN I YNA CRLTGA PRPTCAKLS G CKI S G T C P S D Y P K \\
$S_5$: & - SCCPS T T ARNI YN C YN X C W A G G S R P V C A L S G CKI S G T C B S G W B H \\
$S_6$: & - SCCPNTT GRN I YNCTR L GG GS R E C A L S G CKI S A S T C P S D Y P K \\
$S_7$: & - SCCKNTT GRCYNACR F AGG S R P V C AT A C G CKI S G P T C P R D Y P K \\
\end{tabular}
\end{table}

or first-order features (FOFs). Features relating two or more positions are higher-order (HOFs). Typically, first-order and higher-order features in protein sequence analysis derive from simple calculations of the amino acid frequencies for particular positions. Estimates of underlying probability distributions from some hypothesized population of possible sequences (the entire protein family) are inferred from empirical frequencies over the sample (the known members of the family). These estimates may be based entirely and directly on frequency counts, or may combine frequency counts with other factors, such as prior probabilities and regularizing terms, as in a proper Bayesian approach (Wolpert and Wolf. 1993): extensive "bootstrap" and similar re-sampling computations may also assist (Efron. 1983; Efron and Tibshirani. 1991).

Following this basic approach, one can further compute the variability of a position using the information-theoretic definition of entropy (Shannon and Weaver. 1964; Shannon. 1949; Kolmogorov. 1969; Gatlin. 1972). For a discrete variable $c_j$ we define

$$H(c_j) = - \sum_{i=1}^{|A|} p(a_i@c_j) \log p(a_i@c_j)$$

where $p(a_i@c_j)$ is the probability of the amino acid $a_i$ appearing at position $c_j$.

The entropy $H(c_j)$ ranges from 0 to $\log |A|$, where $|A|$ is the size of the alphabet $A$.

\footnote{Our terminology. The usage of "first order" and "higher order" differs widely across different statistical and mathematical contexts.}
(which in this case is the set of symbols for the 20 amino acids). Thus a completely conserved position $c_j$, having no variability, has entropy $H(c_j) = 0$. The use of entropy-like measures to detect perfectly- and relatively-conserved positions is crucial to the multiple alignment task itself (Altschul and Lipman, 1990; Barton and Sternberg, 1987), and these first-order features often elucidate important evolutionary relationships and structural and functional constraints, as when one speaks of the "invariant tryptophan" or the "conserved cysteines for the disulphide bridges" in the Immunoglobulin constant domain (Kabat, 1985). Such features and their frequency estimation also form the basis for several of the most widely-used template generation and template matching methods of sequence classification, notably including weight-matrix methods (Staden, 1984; Staden, 1988; Staden, 1989), blocks methods (Henikoff, Wallace and Brown, 1990), and the Profile methods of Gribskov et. al. (1988; 1987a).

4.1.2 Higher-Order Features

Although the single-position features are of great use in understanding protein sequence families and motifs, it is clear that higher-order features — representing associations among $k \geq 2$ residue positions — can capture important constraints and relationships missed by first-order analysis.

Consider a set of aligned sequences of symbols (not real amino acid sequences!)$^2$:

```
col1  col2  col3  col4  col5  col6
A     B     C     D     E     F
W     U     C     V     E     G
Z     L     C     M     W     M
V     U     C     V     A     G
A     B     C     D     Z     Z
W     L     C     M     E     Z
```

Example 4.1

At first glance, one notes that position 3 is very conserved; and, reading down from the top, one notes further that positions 2 and 4 seem highly variable, like positions 1, 5, 6. However, whereas positions 2 and 4 display several mutations, they "mutate in lockstep" — their mutations covary perfectly. Given a statistically significant (manifested in many more sequences) example

$^2$We use this same toy example throughout this chapter. The numbers of both rows and columns have been kept unrealistically small for the purposes of exposition in words and figures.
like this, one could extract a second-order feature, perhaps in the form of a "rule" relating positions 2 and 4, like "at (2,4) in protein family \( F \), B goes with D, U goes with V, and L goes with M". If statistically significant, such a relationship would strongly suggest a structural or functional constraint imposed on the two positions, whereby in the course of molecular evolution a mutation at one site must always have been compensated by an appropriate mutation in the other site. Of course, such covariances are not often so perfectly simple — typically, an amino acid may be paired with any of several other amino acids, with each pairing in the rule having an associated probability.

The use of such detectable correlations in structure prediction and classification, and the empirical evidence supporting such use, is discussed in Sections 4.3.1 and 4.3.2 below.

### 4.1.3 The Problem

Having motivated the problem, we now state it formally and in a more general context:

Assume that we are given a database of \( M \) objects \( \mathbf{s}_i \) ("s" for sequence), each of which is characterized by particular values \( a_{ij} \in \mathcal{A}_j \) for each of \( N \) discrete-valued variables \( c_j \) ("c" for column). A particular value for a particular variable is an attribute and denoted \( a_j(c_j) \).

We further assume that there is some "true" underlying probability distribution \( q() \) which, for all orders \( k = 1, 2, ..., N \) specifies the probabilities for each possible \( k \)-tuple of attributes. For example, for \( k = 1 \), we have \( q(c_j) : \mathcal{A}_j \rightarrow [0, 1] \). Inherent in the particular problems we pose is the problem of estimating or approximating the distribution \( q() \), or at least parts of it.

**Problem 1**: Detecting Inter-Column Correlations

Given a set of real numbers \( \rho_k \) for \( k = 1, 2, ..., N \), return a list of all \( k \)-tuples of column numbers \( (j_1, j_2, ..., j_k) \) such that

\[
D(q(c_1^{j_1}, c_2^{j_2}, ..., c_k^{j_k}) || \prod_i q(c_i^{j_i})) > \rho_k.
\]

Here \( D() \) is the Kullback divergence measure, though it is just one of several correlation functions that one may use in this context.

**Problem 2**: Detecting Inter-Attribute Correlations

(“Suspicious Coincidences”)

Given a set of real numbers \( \theta_k \) for \( k = 1, 2, ..., N \), return a list of all \( k \)-ary joint attributes
\((a_1^0 @ c_1^0, a_2^0 @ c_2^0, \ldots, a_k^0 @ c_k^0)\) such that

\[
\frac{q(a_1^0 @ c_1^0, a_2^0 @ c_2^0, \ldots, a_k^0 @ c_k^0)}{\prod_i q(a_i @ c_i)} > \theta_k.
\]

The inequality featured above is only one of many possible formulations of a test for deviation from independence between attributes. More generally, the \(\theta_k\) may be functions (of the particular symbols or marginal probabilities) rather than constants. Our own approach to the problem involves the counting of a particular kind of event that indirectly estimates the the ratios between higher-order joint probabilities and products of lower-order marginals. Hence a more general formulation, and one that encompasses our own approach, is as follows:

**Problem 2b:**

Given a real number \(\theta \in [0, 1]\), return a list of all \(k\)-ary joint attributes \((a_1^0 @ c_1^0, a_2^0 @ c_2^0, \ldots, a_k^0 @ c_k^0)\) such that

\[
P(\text{Observed}(a_1^0 @ c_1^0, a_2^0 @ c_2^0, \ldots, a_k^0 @ c_k^0) | \text{Independent}(c_1^0, c_2^0, \ldots, c_k^0), \text{Model}) < \theta.
\]

for some observed behaviour of \((a_1^0 @ c_1^0, a_2^0 @ c_2^0, \ldots, a_k^0 @ c_k^0)\) and some model which underlies one's estimation or hypothesis testing method.

For the remainder of this chapter, we simplify the analysis and exposition by assuming that all the variables have the same set of possible values, represented as symbols \(a_i \in A\) for some finite alphabet \(A\) of size \(|A|\). (Thus, if the objects are protein sequences \(A\) is the set of 20 naturally occurring amino acids. If the objects are DNA or RNA sequences, then \(A = \{A, C, G, T\}\) or \(A = \{A, C, G, U\}\) respectively. If the domain is speech recognition and the objects are suitably pre-processed utterances, the columns may correspond to time steps and the alphabet to phonemes or other primitive quantized features.)

Note that in our problem formulation, no limiting assumptions are made about how many such cliques of correlated variables there are, how wide they may be (the maximal size of \(k\)), nor on the absolute or relative degrees of correlation. Nor do we make other assumptions about the statistical structure of the distribution \(q()\). In practice, the intrinsic complexity of the problem forces algorithm designers either to place \(a\) priori limits on key input parameters or to tolerate very poor efficiency or estimation results for particular ranges of input parameters or particular data distributions.
4.2 Connection with Other Database Modelling Problems

The problem of finding correlated positions in protein sequence families is closely related to other problems in the probabilistic modelling of datasets. It is useful to look at how novel methods for this computational biology problem might apply to other domains. The database treatment also serves to highlight the key issues in combinatorics and estimation faced by all methods.

If we consider the protein sequences as records, and each column as a field, which takes on values from $A$, then we have the classic database model and can compare and examine methods from different domains. The general problem is to model, in part or in entirety, the underlying probability distribution $q(x_1, x_2, \ldots, x_N)$ for the $N$ fields considered as discrete random variables $x_i$, from which the particular current database records are assumed to have been drawn.

Miller (Miller. 1993) illustrates the general problem with an intuitive example: Consider a database for an automobile insurance company. Each record describes one policy-holder, who is represented by particular values for fields such as Previously-Insured?, Age-Over-25?, Gender, Has-Moving-Violations?, .... and Major-Claim?. (We assume, for simplicity of notation in subsequent derivations, that all of the fields are binary, and we therefore refer to them again as attributes.) There are many possible queries and other operations that one might want to perform on such a database. Each desired operation requires estimating parts of the underlying probability distribution.

4.2.1 Three Goals of Probability Estimation

It is useful for the presentation and discussion of the various probability density estimation methods to delineate three different goals of such estimation, each corresponding to a large body of research:

1. Estimation of the fully-specified, fully higher-order joint probability distribution: Estimate a probability density $q$ that specifies

   $q(\alpha) = q(a_1^\alpha @ c_1^\alpha, a_2^\alpha @ c_2^\alpha, \ldots, a_k^\alpha @ c_k^\alpha)$

   for all $k$-tuples of attributes and possible values.

2. Hypothesis testing, for particular hypotheses concerning particular amino acids and particular residue positions: For example, are the data consistent with the hypothesis that columns $c_1^\alpha, c_2^\alpha, \ldots, c_k^\alpha$ are independent?
3. Feature detection, or “data mining”: Detect the most suspicious coincidences, for example, joint amino acid symbol occurrences that are more probable than would be predicted from lower-order marginals. Related to this, find the most highly correlated \( k \)-tuples of columns. These different problem types are discussed in more detail below.

4.2.2 Estimation/Approximation of the Complete Underlying Probability Distribution

Continuing with Miller’s formulation of the insurance database example, we consider what kind of queries might be made of such a database. For example, given a new would-be policy-holder, how can the database be used to estimate the probability of a major claim being made by the applicant? Solving this in general corresponds to estimating the distribution

\[
q(x_N | u_1, u_2, \ldots, u_{N-1}),
\]

where \( u_i \in \{0, 1\} \) is the particular value for the \( i \)th attribute in the record of the new applicant. That is, either \( u_i = x_i = 1 \) or \( u_i = \tilde{x}_i = 0 \).

Using Bayes’ Rule, and the fact that the attributes are binary, one obtains

\[
q(x_N | u_1, u_2, \ldots, u_{N-1}) = \frac{q(u_1, u_2, \ldots, u_{N-1}, x_N)q(x_N)}{q(u_1, u_2, \ldots, u_{N-1})q(x_N) + q(u_1, u_2, \ldots, u_{N-1}, \tilde{x}_N)q(\tilde{x}_N)}
\]

where the \( x_N, \tilde{x}_N \) notation means the 1 and 0 values, respectively, for the \( N \)th attribute.

The question of how best to estimate these higher-order probabilities exemplifies the difficulties and tradeoffs in probabilistic modelling and machine learning.

At one extreme, one can in this example make the strong assumption of conditional independence, and set

\[
q(x_N | u_1, u_2, \ldots, u_{N-1}) = 
\frac{q(u_1 | x_N)q(u_2 | x_N) \ldots q(u_{N-1} | x_N)q(x_N)}{q(u_1 | x_N)q(u_2 | x_N) \ldots q(u_{N-1} | x_N)q(x_N) + q(u_1 | \tilde{x}_N)q(u_2 | \tilde{x}_N) \ldots q(u_{N-1} | \tilde{x}_N)q(\tilde{x}_N)}.
\]

Alternatively, if the database is large and contains many identical records, then one can estimate \( q(x_N | u_1, u_2, \ldots, u_{N-1}) \) simply by counting the number of records that share the attribute values \( u_1, u_2, \ldots, u_{N-1} \) and which show a major claim \( x_N \).

The first approach will produce large modelling error, or bias, if the conditional independence assumption provides a poor approximation to the higher-order probabilities.
The second approach would typically introduce large sampling error, or variance, which measures the difference between the underlying $N - 1$th-order probabilities and the estimates made from small frequency counts.

4.2.2.1 Modelling Reduces to Maximum Entropy Assumption and "Suspicious Coincidences"

The recognized need for estimation techniques that find some near-optimal trade-off between bias and variance (Geman, Bienenstock and Doursat, 1991) has been addressed by a long series of methods that employ a combination of lower- and higher-order terms with some kind of maximum entropy assumption (Lewis, 1959: Ku and Kullback, 1969: Miller, 1993). The basic idea behind all such methods is to assume that all higher-order probabilities $q(u_{i1}, u_{i2}, \ldots, u_{ik})$ are predicted exactly by the product of the probabilities of some subset of their lower-order marginal probabilities, unless there is explicit and strong evidence otherwise. Using this variant of a maximum entropy assumption (Lewis, 1959: Rissanen, 1988), one need only to represent explicitly those higher-order terms for which $q(u_{i1}, u_{i2}, \ldots, u_{ik}) > \prod_j q(u_{ij})$ by some amount to be chosen by the modeler. Such terms were called "suspicious coincidences" by pattern recognition pioneer Barlow (Barlow, 1972; Barlow, 1989), and they play a crucial role in much of this century's work on perception, learning, and pattern recognition (Watanabe, 1985) (though this role was not always recognized clearly by researchers and practitioners in these fields).

This important idea can be made clearer by following further with Miller's presentation (Miller, 1993). Miller presents strong arguments for the use of Gibbs models in modelling a database. The Gibbs models are built from a special set of parameters, called potentials, one for each pattern of attribute values $(u_{i1}, u_{i2}, \ldots, u_{ik})$ in the database. Probabilities of particular patterns may be derived from their potentials, and vice versa, using a special Gibbs' Relation equation derived from statistical mechanics. The basic form of the potential function $J(\omega)$ represents the relationship between the actual probability of a higher-order pattern and the probability predicted by its lower-order marginal components:

$$J(\omega) = Z \log q(\omega_\Gamma) - \sum_{\gamma \subset \Gamma, |\gamma| = |\Gamma| - 1} (|\Gamma| - 1)^{-1} Z \log q(\omega_\gamma), \quad (4.1)$$

where $\omega$ is a pattern of attribute values, $\Gamma, \gamma$ are sets of attributes, and $Z$ may be thought of as a normalizing constant, analogous to the standard partition function in statistical mechanics.
For example, if the set of attributes is \( \{x_1, x_2, x_3\} \) and \( \omega = (u_1, u_2, u_3) \), then, using the abbreviation \( q(\omega_{\{x_1\}}) = q(x_1) \), and so on, we have (Miller, 1993):

\[
J_{\{x_1\}}(\omega) = Z \log q(x_1) \\
J_{\{x_1, x_2\}}(\omega) = Z \log \frac{q(x_1, x_2)}{q(x_1)q(x_2)} \\
\ldots \\
\ldots \\
\ldots \\
J(\omega) = J_{\{x_1, x_2, x_3\}}(\omega) = Z \log \frac{q(x_1, x_2, x_3)}{[q(x_1, x_2)q(x_2, x_3), q(x_1, x_3)]^{1/2}}
\]

Typically, most of the higher-order potentials tend to be near zero, because the numerators within the logarithm are approximated well by the denominators. Thus, in cases wherein components of a higher-order pattern are unknown, or cannot be estimated, or when the pattern is simply assumed to be "unsurprising", \( J(\omega) \) can be set to zero. Because the number of potentials grows exponentially in the number \( N_A \) of attributes, it will not in general be feasible to calculate and store every potential. Instead, only the largest magnitude potentials are used in modelling the database. The mathematics of Gibbs models manages the relative contributions of all the potentials so that the probability estimate of a higher order pattern works out to be an optimal maximum entropy estimate, given the choice of which potentials are calculated and stored (Miller, 1993; Geman and D., 1984). The onus is on the modeler to manage the bias/variance trade-off by deciding which potentials to store. Therefore there is a need for very fast methods to find the suspicious coincidences which correspond to the higher-magnitude potentials.

4.2.3 Hypothesis Testing

The general problem of hypothesis testing can be stated as:

Given a dataset \( \mathcal{D} = \{\tilde{s}_1 = \tilde{u}_1, \tilde{s}_2 = \tilde{u}_2, \ldots, \tilde{s}_M = \tilde{u}_M\} \) and a null hypothesis \( H \), decide whether the estimated probability \( P(\mathcal{D}|H) \) of the data given the hypothesis is sufficiently low that we should reject the hypothesis. Typically, the hypothesis corresponds to a particular probability distribution \( q(c_1, c_2, \ldots, c_N) \), over a set of variables \( \{c_1, c_2, \ldots, c_N\} \), that underlies or generates the data, and thus one makes inferences based upon an estimation of \( P(\mathcal{D}|q) \). Sometimes, the hypothesis test concerns a choice between two or more particular distributions,
$p_1(c_1, c_2, \ldots, c_N), p_2(c_1, c_2, \ldots, c_N), \ldots, p_z(c_1, c_2, \ldots, c_N)$. Often, one eschews the analysis of the shapes of the entire distributions in favor of simple comparisons between their respective means, variances, or other moments and parameters (Silvey, 1975).

An important kind of hypothesis test that is very relevant to the analysis of macromolecular sequence families is contingency table testing for association between two (or more) discrete variables. Given two discrete random variables $c_1$ and $c_2$, and a set of data $\tilde{u}_i = (u_{i1}, u_{i2})$, where $u_{i1} \in A$, $u_{i2} \in B$ (often we have $A = B$), one first forms a contingency table $T_{jk}$ of size $|A| \times |B|$, where $t_{jk}$ is the frequency of occurrence of the joint symbol pair $(a_j, b_k)$. $a_j \in A$, $b_k \in B$. In this hypothesis testing framework, the null hypotheses is typically that the variables $c_1, c_2$ are independent.

There is an extensive literature on this problem. Currently popular approaches are based on: (1) estimating a model, in terms of a parameterized probability distribution $p'(\cdot)$, from the data and then (2) performing a goodness-of-fit test on $p'$ versus $p_{\text{ind}}$ (where $p_{\text{ind}}$ is the maximum entropy distribution formed from the first-order marginals). The first step, model estimation, typically involves an iterative maximum likelihood procedure. This second step, the hypothesis test, often uses a classic statistic like Pearson's $\chi^2$ or the deviance (a likelihood ratio test, similar to Kullback divergence). These two standards have in recent years been subsumed under the more general class of power-divergence statistics. Such methods bring sophisticated mathematical machinery and considerable amounts of computation to bear on the task of deciding the independence or dependence of very small sets of pre-selected random variables. They do not offer much assistance in finding cliques of probably-correlated variables amongst large numbers of random variables. These methods are good for answering carefully-posed questions such as, “Is there good evidence that residues 55 and 84 covaried throughout the evolution of the Phospholipase family?”, but not for deciding which questions to pose.

### 4.2.4 Data Mining

If hypothesis testing concerns the verification of interesting features, then data mining is the theory and practice of discovery of such features. Before one can test hypotheses, one needs to be able to generate them. This hypothesis generation is increasing in importance as scientific, governmental, and commercial databases grow ever larger and more interconnected. Methods that can find unexpected associations between variables or attributes and find “natural” clusters of records will be at a premium in the coming years. For example, an investor, given a database
of hourly quotes for stock, bond, currency and commodity prices. would presumably be very interested in knowing whether there are any \( k \)-tuples of financial instruments whose prices tend to covary over time (especially if there were a small, constant time-lag between members of the set). The database may be so large, and the time requirements so stringent, that neither full-scale probabilistic modelling nor systematic series of hypothesis tests are feasible. The investor needs fast, heuristic methods that produce small sets of interesting candidates for further analysis.

Clearly, the data mining scenario is very relevant to the particular protein sequence family modelling problem that we have described.

### 4.2.5 Computational Complexity

In all but the most trivial situations, one discovers very quickly that an exact determination of all probability densities over all attributes and values is impossible. The impossibility has one main cause — there are too many joint attribute-value density terms — and this manifests itself in two main problems: (1) computing and storing frequency counts over all terms, over the database, requires too much computation and memory; (2) there is usually an insufficient number of database records to support probability estimates based on those frequency counts.

Let us consider some details. For \( M \) records, \( N \) variables, and \( |\mathcal{A}| \) possible values, there are

\[
\binom{N}{k} = \frac{N^V}{(N-k)k!} \quad \text{k-tuples of columns. Adding the number of k-tuples for each } k = 1, 2, \ldots, N \text{ results in } 2^N - 1 \text{ such tuples of all sizes.}
\]

One natural way to think about this complexity is in terms of the power set \( 2^Q \) of the set \( Q \) of column variables. \( 2^Q \) is a lattice under the operation \( \subseteq \), a "tower" corresponding to a graph whose nodes are subsets of \( Q \). From this viewpoint, two nodes representing subsets \( \sigma_1 \) and \( \sigma_2 \) are connected iff either \( \sigma_1 \subseteq \sigma_2 \) or \( \sigma_2 \subseteq \sigma_1 \). We say that \( \sigma_2 \)'s node is above \( \sigma_1 \)'s if \( \sigma_1 \subseteq \sigma_2 \). This gives a natural meaning to the term "higher-order", as appearing higher up the tower. We call the bottom, the null set node, the 0th tier; the single column terms form the first tier. and so on. (See the figure on Page 115.)

Continuing with the tower analogy, we note that each "floor" of this edifice contains

\[
\binom{N}{k} \quad \text{"suites"}, \text{ and each suite contains } |\mathcal{A}|^k \text{ "rooms". In other words, the kth level of the lattice corresponds to } \binom{N}{k} \text{ different } k\text{-tuples of column variables, and associated with each such } k\text{-tuple is an } (|\mathcal{A}| \text{ by } |\mathcal{A}| \ldots \text{ by } |\mathcal{A}|) \text{ contingency table, each cell of which must store} \]
the counted frequency of a particular joint symbol \((a_{i_1}, a_{i_2}, \ldots, a_{i_k})\) were one to use a classical contingency table test for the correlation between those particular \(k\) columns.

For any \(k \in \{1, 2, \ldots, N\}\), for any particular \(k\) \textit{tuple} of columns \((c_{j_1}, c_{j_2}, \ldots, c_{j_k})\), there are \(|A|^k\) possible joint values. For any \(k \in \{1, 2, \ldots, N\}\), for any particular \(k\) \textit{tuple} of columns \((c_{j_1}, c_{j_2}, \ldots, c_{j_k})\), the estimation of Kullback divergence or other correlation function using the
Figure 4.2: Shown above are the $n$-grams, for all sizes $n = 1, 2, \ldots, 6$, from the toy dataset. One way around the combinatorial barrier to comprehensive probabilistic modelling is to estimate the joint probabilities for only the terms composed of $n$ sequentially-adjacent symbols, for some chosen, fixed $n$. Markov models of all orders are based on this idea: standard Hidden Markov Models as used in protein sequence modelling corresponds, roughly, to analysis of only a small subset of the terms in the third tier up from the bottom of the lattice diagram.

Dataset is at least an $\Omega(Mk)$ or $\Omega(|A|^k)$ computation, depending upon the relative sizes of $M$, $k$ and $|A|$.

A comprehensive probabilistic model of the database must be able to specify probability
Figure 4.3: The figure illustrates the consideration of all possible pairwise correlations. This corresponds to analysis of the third tier up from the bottom of the lattice diagram. This particular heuristic shortcut is the one taken by most previous and current work on inter-residue correlations in protein and RNA sequence families.

estimates for $\sum_{k=1}^{N} \binom{N}{k} |\mathcal{A}|^k$ terms. This means, for example, that for a tiny heptapeptide family, each sequence having a length of seven amino acid residues, there are $1,801,088,540$ terms to specify. For an unrealistically small RNA of fifteen nucleotides in length, over the smaller four-base alphabet, there are $30,517,578,124$ terms.
Figure 4.4: This figure depicts a partition of the variables in the toy dataset. A partition is one particular and important kind of componential model of a sequence family or other aligned dataset. In a componential model, a set of $N_Y$ latent $y_i$ variables is found to "generate" or "explain" a larger set of $N$ observable variables $c_i$. In a partition model, $N_Y \leq N$, each $c_j$ is generated by exactly one of the $y_i$, and typically $N_Y < N$. The observables corresponding to one latent variable form a kind of clique, and presumably are highly correlated with each other and relatively uncorrelated with variables outside the clique. Above, the observables are formed into three cliques: $(c_1, (c_2, c_5, c_6)$, and $(c_3, c_4)$.

Clearly the models can become intractably huge. What about the *space of possible models* through which a modelling/learning procedure must search? Consider a latent-variable model.
which seeks to explain correlations between sets of observable variables by positing latent variables whose states influence the observables jointly (as in MacKay's model (MacKay, 1994), for example). Since each model must specify a set of \( k \)-tuples of variables, and there are \( 2^v \) such sets, there are \( 2^{2^v} \) possible models in the worst-case search space.

Various methods for estimating higher-order probabilities will circumvent the combinatorial explosion through severe prior restrictions on the width \( k \) (see figure, Page 117), the locality (Page 116), the number, or the degrees of correlation of the higher-order features sought, and on the kinds of models entertained (Page 118).

### 4.2.6 Is Protein Sequence Family Analysis a Database Problem?

Previous sections discussed the protein sequence family analysis problem as a problem in modelling a database of discrete-valued records: such a viewpoint is useful because it focuses on the interrelationships among residues (columns) in the aligned set of sequences. However, two important aspects of this protein analysis problem take it outside the realm of classical database problems:

1. **Local sequential structure**: Protein sequences are sequences, after all. There is a unique natural ordering on the column variables in a protein sequence database, unlike an insurance policy database. While it is true that many previous sequence analysis methods focused on the local sequential structure to the exclusion of higher-order and more distant interactions, one must take care to account for the generally higher degrees of correlation between sequential neighbors compared with those between arbitrary distant pairs of residues.

2. **Correlation among the sequences**: In database modelling, it is typically assumed that the records — in our case, the \( M \) protein sequences — are *independently and identically distributed (i.i.d.)*. This assumption is manifestly untrue in the protein sequence domain, as the members of a protein (or DNA, or RNA) family share an evolutionary history, and therefore result from a tree-structured generation process.

The goal, then, is to recognize the sequential nature of the data and beware of possible phylogenetic (evolutionary, as opposed to structural) correlation features, but to use the database assumptions to derive fast and effective methods for detecting inter-residue correlations.
4.3 Importance in Protein Structure Prediction and Analysis

Other sections of this chapter explore and assess the feasibility of detecting correlations between variables (which we represent as columns) in a large dataset of aligned sequences. If this technology is to be applied to the macromolecular sequence domain, there remains the key question: Do such detectable correlations exist in macromolecular sequence data, and, if so, how can they really be used to support structure prediction or classification?

These questions are addressed within the following two sections, with reference made to a representative sample of relevant work in molecular, evolutionary, and structural biology and biochemistry.

4.3.1 Discovery of Evolutionarily-Conserved Structural Constraints

Three questions are addressed in this section:

1. What kinds of evolutionarily conserved multi-residue structural or functional constraints might one expect to find by detecting correlations between columns in a multiple sequence alignment?

2. Have correlation-detection efforts in fact found important structural or functional constraints?

3. How much information do such discoveries provide towards predicting or determining a molecule's native tertiary structure?

4.3.1.1 What Do We Expect to Observe?

A protein family is the set of amino acid sequences that are believed to share a common global tertiary structure. The theory and observation of protein folding and evolution supports the general idea of evolution and conservation within a protein family:

- **Functional constraints** are conserved in surface residues:

- **Structural constraints** are conserved in core residues:

- **Mutational drift** dominates in loop residues:

  Functional constraints often involve other molecules — such as other proteins, nucleic acids, lipids, metals, O₂ or other small molecules.
The kind of structural constraints expected to be conserved throughout evolution of a protein family are mainly those involving a few key residues that stabilize a conformation. Where electrostatic interactions are deemed important, one might expect to find a conservation of net charge across two or more sequence positions. When one of two electrostatically interacting residues carries a positive charge, its "partner" residue (presumably close in 3D structure even if distant in sequence) should be negatively charged, and vice versa. The situation is similar for packing constraints. One might reasonably expect sections of the protein core volume to vary only slightly across the many different proteins in the same structural family, while non-core regions might display large volume variability. Thus one might expect to find pairs or small \( k \)-tuples of residues that display mutually compensatory mutations with respect to side-chain volume — when a "Large" mutates to a "Small", another "Small" must mutate into a "Large".

to put it simplistically.

4.3.1.2 What Has been Observed?

Neher et al. (Neher, 1994) attempted to quantify the frequency of compensatory changes within a single protein family by using physico-chemical property indices for amino acids and then estimating Pearsonian correlations between columns in an alignment. They attempted to get around the small-dataset problem with a bootstrap-inspired resampling scheme based on the examination of pairs of sequences from the family. Their study of the myoglobin family of protein sequences found the degree of compensatory mutation to be low for the property of side-chain volume but high for electrical charge — close to the correlation level expected for perfect conservation of local charge. The authors speculate that because their column-pair analyses focused only on contact-neighbour pairs of residues, they were able to detect a very locally-acting constraint like charge conservation but not a more distributed constraint like conservation of volume. (In other words, a single positively-charged residue must be in contact with its single negatively-charged structural partner, whereas a set of compatible-volume partners may comprise more than two residues and need not all be in contact.) Altschuh (Altschuh et al., 1987b: Altschuh et al., 1988) has also found some evidence of coordinated mutation in the evolution of protein structural families.

While most studies, to date, of compensatory mutation focus on highly-conserved "core"-type regions of protein structures. Korber et al. analyzed the highly-variable V3 loop of the HIV-1 envelope protein. The researchers performed robust bootstrapped estimates of the pairwise
mutual information for all column-pairs from a set of 31 columns representing V3 residues. They found a set of about seven pairs that showed considerable and statistically-significant mutual information, and their analysis of the particular attributes (amino acids) suggested a particular pattern of highly likely compensatory mutations. Although the authors did not argue or provide evidence for any particular properties or relationships being conserved, subsequent mutational analysis experiments in the laboratory indicated functional linkage between some of the pairs of sites with high mutual information. Because the V3 region is known to be both functionally and immunologically important, it is suggested that such analyses might be important in the search for HIV/AIDS vaccine design. (We performed experiments and analysis on a larger, updated version of the same dataset, as we report in Section 4.6.5 below.)

Other recent work adds to the evidence that functional and structural constraints can be detected from covariation analysis of aligned sequence positions, especially if the results of such mathematical analysis are supported by subsequent double-mutant cycle analysis. In another study (Horovitz et al., 1994) the authors report the use of this combined computational and laboratory experimentation on the GroEL family of chaperonin proteins. Their study revealed the prediction and verification of an interaction between two predominantly cysteine residues in the structure.

4.3.1.3 How Can These Observations Be Used?

Even if we cannot always figure out exactly which physico-chemical constraints are conserved among a set of correlated residues, it is worth asking whether the mere detection of such correlations can be used to predict 3D structural proximity. If so, then even a few such predictions might provide crucial information in global structure prediction, through the use of distance-geometry constraints (Havel, Kuntz and Crippen, 1983; Crippen and Havel, 1988; Chan and Dill, 1990; Sibbald, 1995; Cariani and Goel, 1985), perhaps followed by energy-minimization or molecular dynamics simulation, as discussed briefly in Chapter Two.

Again, detection of correlated mutations does not always imply spatial proximity. Putting aside the merely spurious correlations and the artifacts of poor estimation methodology, there will remain many instances of coordinated mutation that reflect non-local function synergism.

There is much ongoing work in this area, but results thus far indicate that the prediction of pairwise inter-residue distances from correlation information can provide an improvement of 1.4 to 5.1 times over random contact predictions (based on size, molecular weight and amino acid
composition of the protein). However, such quoted results depend on the particular proteins and methods used and especially on the coverage of the contact prediction method, i.e., the percentage of the residues for which the method provides any prediction of a sequence-distant contact (Shindyalov, Kolchanov and Sander, 1994; Goebel et al., 1994). By itself, this kind of information is not nearly enough for ab initio prediction of global conformation, although it may be enough in many cases to distinguish between two or more alternative conformational models, or to provide additional constraints for energy minimization and molecular dynamics optimization.

4.3.1.4 Correlated Positions and RNA Structure Prediction

The prediction of RNA secondary and tertiary structure is another important and active area of research. Here, too, the discovery of covarying positions in an aligned set of sequences can play a role. Applying the technique to RNA sequences poses a slightly different set of challenges. One faces a smaller alphabet, hence possibly more noise in any simple counts of joint symbol occurrences. RNA sequences can be much longer than typical protein sequences, with correspondingly more severe combinatorial explosion. On the other hand, if we can find significant pairwise correlations we can go further in exploiting the implied proximity constraints - we know that pairwise interactions are the crucial components of secondary structure and most tertiary structure is secondary structure in the world of RNA folding (Sakakibara et al., 1995; Turner, Sugimoto and Freier, 1988; Zuker, 1989; Gutell et al., 1992).

4.3.2 Building More Sensitive Protein Classification Systems

As is discussed above, the conditional independence assumption is troublesome for modelling amino acid sequences because the forces that confer structure and function in a protein are mediated through specific amino acid interactions. To miss such interactions can doom structure prediction: can it also hamper the simpler task of mere sequence classification? The answer is "yes".

Consider a classifier based on first-order amino acid statistics for some hypothetical protein sequence family. Suppose that two particular highly correlated positions $c_1$ and $c_2$ are crucial for the family's shared tertiary structure. It might be that these positions represent a contact-pair of residues and hence that their covariation represents a structural constraint — for example, conservation of net charge. If, throughout the family of sequences, the positively- and negatively-
charged amino acids alternate over these two positions, then the simple first-order single-column statistics for each of the two positions might display no significant preference for amino acid identity or even for a particular amino acid property (like "Positively Charged"). A new member of this protein family tested against the classifier might have the requisite pair of oppositely-charged amino acids at positions $c_1$ and $c_2$, but the classifier would miss this key feature. Only a classifier built upon higher-order statistics can detect and exploit such pairwise and generally $k$-ary signs of class membership.

Once the significant inter-residue correlations have been found, they can be built into representations that make for fast and sensitive classifiers based upon, for example, the methodologies of graphical models (Buntine, 1994; Klingler and Brutlag, 1994).

### 4.4 Previous and Alternative Methods

#### 4.4.1 First-Order Models

In order to use fast and relatively simple algorithms for model generation and sequence-to-model matching, many popular methods in the 30-year history of computational molecular biology embody an assumption of statistical independence between residue positions. Setting

$$p(\tilde{s}) = p(s_1, s_2, \ldots, s_N) = \prod_i p(s_i)$$

and

$$p(\tilde{s} | C_i) = p(s_1, s_2, \ldots, s_N | C_i) = \prod_i p(s_i | C_i),$$

where $C_i$ is some structural or functional class.

This is the basis of the Profile method, for example. The single-residue marginal probability priors in a profile reside implicitly in two tables of amino-acid-to-amino-acid substitutivity/matching weights, one based on general mutation patterns observed across all known proteins, and the other based on the training set of aligned sequences for the family being modelled. The weights are calculated from single-column amino acid frequencies.

#### 4.4.2 Direct Calculation of Pairwise Probabilities

Perhaps because of the computational complexity and data complexity of computing all pairwise correlations between positions in long sequences, there has been little reported work on this kind
of analysis in DNA, RNA, and proteins. This is changing. A few relatively recent papers (Gutell et al., 1992; Korber et al., 1993) discussed the use of site-pair mutual information calculations in structural and functional analysis of RNA and protein molecules.

Gutell et al. (1992) found that 2-position correlations not only help reveal the base-pairings underlying the secondary structures of, e.g., tRNAs, but also reveal aspects of tertiary structural interaction and other non-basepairing and non-contiguous interactions.

The Los Alamos group (Korber et al., 1993), as mentioned earlier, also estimated pairwise inter-column correlations as a way of finding possible inter-residue interactions important in structure and function.

The tiny subset of the higher-order variables space analyzed by pairwise methods is illustrated on Page 117. One can make the heuristic guess that a set of $k > 2$ columns characterized by high pairwise correlations also display significant higher-order ($k > 2$) correlations. This corresponds, more or less, to considering the transitive closure of the “Correlated With” binary relation, and there are many possible ways to do this. Like any heuristic, it can lead to trouble: both false positives and false negatives are possible.

### 4.4.3 $n$-gram Methods

Fixed-length “$n$-grams”, or contiguous $n$-symbol substrings, as employed by search utilities like FASTA and BLAST (Lipman and Pearson, 1985a; Altschul et al., 1990), offer one way to truncate the full power set expansion. (See figure Page 116.) The collected $n$-gram frequencies of order $n$ also form the basis for $n-1$th order HMMs. $n$-grams have been used for decades in the statistical analysis of language and in data compression and cryptography. They have the virtue of being very fast to compute, but they do not of themselves offer a direct way to discover long range interactions. Hunter et al. (1992) used BLAST on every protein in the NRDB protein database, applying various measures of separation and overlap to identify interesting regions or clusters of associated $n$-grams. Like the HMM-based methods, these methods focus on sequentially-local interactions between residues. Whereas the database-oriented methods allow widely-separated sites to belong to one interesting $k$-tuple feature, the method of Hunter et al. coalesces neighboring $k$-grams into single, connected regions of biological interest.
4.4.4 Hidden Markov Models and Induction of CFG’s and CSG’s

Hidden Markov Models (HMMs) have been used widely and with increasing success in recent years, in the modelling of protein (Haussler et al., 1992; Smith, White and Stultz, 1991; Baldi and Chauvin, 1994), DNA, and RNA (Sakakibara et al., 1995) sequences.

A first-order discrete HMM can be viewed as a parameterized stochastic grammar for recognition of symbolic sequences, or as a production system for generation of such sequences. The model is adaptive, in that the parameters can be estimated and re-estimated in order to force the model to better fit a set of data. A model is specified by a set of states $S$, an alphabet $A$ of symbols, a state transition probability matrix $T = (t_{ij})$ and a symbol emission probability matrix $E = (e_{ia})$. In protein sequence modelling $S$ typically includes special states $\text{INSERT}$ and $\text{DELETE}$. Data is considered to be generated by the model as the model changes from state to state and in each state emits a symbol. An HMM system in state $i$ has probability $e_{ia}$ of outputting symbol $a$ and probability $t_{ij}$ of moving into state $j$.

HMMs may be trained by adjusting the parameters to optimize some objective function measuring the fit of the model to the data. Most often, the likelihood $P(\text{data} \mid \text{model})$ is optimized, usually by some variant of the EM algorithm (Dempster, Laird and Rubin, 1976) for which convergence (to a local optimum) is guaranteed (Bahl, Jelinek and Mercer, 1983; Bengio and Frasconi, 1996; Dempster, Laird and Rubin, 1976).

Although some groups have reported significant success in modelling protein sequence families with HMMs (Haussler et al., 1992; Smith, White and Stultz, 1991), nonetheless there are probably great improvements to be made in learning time and model robustness by the “hardwiring” of pre-selected higher-order features into HMMs. (This has been investigated for HMM-like recurrent neural networks, in different domains (Giles et al., 1992).)

Some of the same reasons why HMMs are very good at aligning the sequences in the first place, using local sequential correlations, make such methods less useful for finding the important sequence-distant correlations in data that has already been partially or completely aligned. The phenomenon responsible for this dilemma, termed “diffusion”, is examined in some detail in recent work by Bengio and Frasconi (Bengio and Frasconi, 1996).

A first-order HMM, by definition, assumes independence among sequence columns, given a hidden state sequence. Multiple alternative state sequences can in principle be used to capture longer-range interactions, but the number of these grows exponentially with the number of $k$-tuples of correlated columns.
Recognition that many of the features of biological interest do not fall within the formal language classes representable (let alone learnable) by HMMs (Searls, 1988; Sakakibara et al., 1995; Jiménez-Montaño, 1984) has led some researchers to construct and train more powerful models for generation and recognition, such as context-free stochastic grammars (Sakakibara et al., 1995). As such methods sample higher-order feature combinations than do first-order HMM methods, their time, space, and data complexity are correspondingly greater.

### 4.4.5 Neural Networks

Neural network models and their training algorithms now exist in sufficient number and diversity that they have become essentially another model of universal computation. Most "standard" statistical methods can be implemented in a neural network framework and many neural network methods have been shown to pre-exist in the literature on statistics, information theory, optimization, or control theory. This intellectual cross-pollination is certainly evident in the area of unsupervised learning methods, wherein Becker and Plumbley (1996) have outlined three major subareas:

1. **Information-preserving methods.** Including principal component and principal surface analyses;

2. **Density-estimation methods.** Including competitive learning and similar formulations of clustering and latent class modelling;

3. **Higher-order feature-extraction methods.** Which one might attempt to use at higher (later) stages of processing, in a multi-stage system.

Most relevant to the focus of this chapter are some of the methods in the third category, which make constraining assumptions about the kinds of higher-order statistical structure one expects to find and capture in the data.

The GMAX algorithm (Pearlmutter and Hinton, 1986) is able to discover statistical dependencies (correlations) between the input lines into a processing unit. It does this by adjusting the weights on the input lines so as to maximize the divergence $D(p||q)$, where $p(u)$ is the output distribution of the unit $u$ in response to the real (structured) input data, and $q(u)$ characterizes the outputs that would be expected if the input lines were independent. The exact calculation of $q(u)$ is infeasible (at least in the discrete case in which we are interested), because it would require
sampling of all $2^N$ possible states of the $N$ input units: so in practice one typically estimates the
distribution by sampling from carefully-contrived input vectors with independent components.
A new extension of the GMAX algorithm to multiple output units has been successfully tested
(De Sa and Hinton, 1997).

The search for dependencies among inputs is not far removed from the converse task of
finding and separating the independent components, or causes, of a dataset. Neural network
researchers have been at the forefront of this factorial learning subfield, and their collaboration
with statisticians and statistical physicists has produced several theoretically interesting algo-
rithms (Ghahramani, 1995b: Marques and Almeida, 1996). These methods make use of Bayesian
(and MDL, MML, and stochastic complexity) principles in managing the bias/variance tradeoff:
the tradeoff is typically implemented by limiting the collection of higher-order terms to some
predefined order $k$, with $k = 2$ or $k = 3$ most common.

It is worth noting that the standard Boltzmann Machine (Ackley, Hinton and Sejnowski, 1985) is essentially a second-order Gibbs model, and thus fits into Miller's framework discussed
in Section 4.2.2.1 above. In a standard (second-order) Boltzmann machine, all potentials of
order greater than two are simply ignored. This constraining assumption makes learning (com-
putation of the potentials) feasible, and corresponds, in our protein modelling problem, to the
direct estimation of all pairwise joint symbol probabilities and correlations (as illustrated in the
figure on Page 117). Higher-order Boltzmann Machine architectures have also been investigated
(Galland and Hinton, 1990).

4.4.6 MacKay's Density Networks

MacKay (MacKay, 1994) applies Bayesian probability density modelling, in the form of a "den-
sity net", to the problem of protein family modelling.

Essentially, he posits a latent variable model $p(c_1, c_2, \ldots, c_N | y_1, y_2, \ldots, y_L)$ for the gener-
ation of outputs (amino acid symbols) over the observables $c_1, c_2, \ldots, c_N$ (columns) given the
states of the latent variables $y_1, y_2, \ldots, y_L$. Hierarchical Bayesian learning is used to adjust
hyper-parameters and parameters in the model.

The core ideas of the MacKay approach are illustrated in in rough schematic form on Page
130. Each vertical half of the page represents one trained density network and the data upon
which it was trained. The data is the same set of toy sequences used throughout the chapter.
Six units (circles), at the bottom of the page, represent the latent variables $\{y_i\}$. (Not all of
them need be "used" in a given trained network.) Above them are six units representing the observable variables \{c_i\}, and above these are the output data, the amino acid sequences typically considered as "input" to a modelling algorithm. Connection lines (shown as line segments) implement the mapping from the latent variables to the observable variables. Weights \{w_h\} on the connections parameterize the mapping. (Connections with large weights are shown in bold, smaller weights as thin line segments, and connections with zero weights are not drawn.) A set of hyper-parameters \{\alpha_l\} (not shown) govern such aspects of the model as the number of effective latent variables, and the number, initial values, and ranges of the weights, in order to achieve particular regularization goals. The regularization goals in this case serve not only to enhance learning and generalization, but also to produce a model that makes sense (to human experts) in the protein modelling domain. Two different hypothetical trained density nets are shown side by side, each one representing a different model of the relationships between latent and observable variables — and hence also of the relationships among the observables. The first model, on the left-hand side, explains the apparent correlation between columns 2 and 4 in terms of a single latent variable that is responsible for them, and treats the other columns as independent. The second model, on the right-hand side, puts together columns 1.5 and 2.3.4. and treats column 6 as independent. Latent variable units with only zero connections may be considered as "unused" — in such cases, an effective dimensionality reduction is achieved.

MacKay's model is aimed at a comprehensive modelling and as such it is computationally very expensive. For example, for the standard alphabet of twenty amino acids, and sequences of length \(N = 200\), the model and training procedure would use about 80,000 parameters. It is also arguably the most flexible and principled approach to the comprehensive modelling problem to date. The modeler may set priors on the hyper-parameters in order to build in whichever kinds of statistical structure one expects or hopes to find in the data. Such bias is made explicit and therefore controllable: and, as in any Bayesian learning approach, it can in principle be overcome by the data.

To date, published results are available only for experiments performed on toy models and sequences (\(M = 4\) sequences, \(N = 4\) columns) and a toy model (\(L = 2\) latent variables) for the globin family of proteins (\(M = 400\), \(N\) unspecified, probably \(\approx 150\)). The method works well on the toy data, and is claimed to have found interesting subclass structure in the Globin class of protein sequences, by virtue of the latent componential features discovered in the training (MacKay, 1994).
Figure 4.5: Above is an extremely simplified schematic of two possible neural network latent variable models for the toy dataset. The left-hand model “explains” the data by postulating a single generator (latent variable $y_2$, corresponding to the second node from the left on the bottom row) for the observable variable corresponding to columns 2 and 4, and a single generator for column 3. Low magnitude weights (dashed lines) indicate weak explanations for the columns 1, 5 and 6. The right-hand model postulates single latent generators for column cliques (1, 5) and (2, 3, 4).

### 4.4.7 What Kind of Method is Needed?

As with the other comprehensive modelling methodologies, the MacKay approach would presumably benefit from a fast preprocessing stage that could find candidate subsets of correlated observable variables and allow one to pre-set, that is, to bias, some of the priors and hyper-priors accordingly. This would in many cases greatly reduce the search time and hence overall running time of the MacKay-inspired methods, though at some risk of missing an optimal but well-hidden solution.

Clearly, several well-studied and effective methodologies exist for the comprehensive modelling of protein sequence families. In each case, the mathematical machinery is in place to handle and detect very local and low-order statistical structure in the data. In each case, the difficulties with computational complexity and statistical estimation arise in the attempt to account comprehensively for all possible non-local and higher-order interactions between residues, i.e., columns, in the aligned sequence data.
Perhaps such algorithms are trying to do too much. Easier progress in modelling might be made if one were to use HMMs or density networks in conjunction with a fast. heuristic preprocessor that focuses explicitly on the detection of plausible non-local interactions while sacrificing a degree of precision in modelling these interactions. We present such a procedure below.

4.5 Coincidence Detection: A Novel Data-Mining Method

One goal of the research described in the current chapter is the development of a fast algorithm for detection of correlations between columns in a set of aligned protein sequences from one family or superfamily. This section presents a general and novel methodology that can perhaps serve as one important component in a suite of computational tools for the exploration and “mining” of data.

The method is designed to get around the central obstacle to higher-order feature discovery in the sequence family modelling problem: We do not want to specify or limit, \textit{a priori}, the number of possible $k$-tuples of correlated columns, the width $k$ of any of them, or the degrees of correlation involved; and yet we do not want to explicitly represent and process latent variables or parameters for the exponentially-many possible $k$-tuples. Therefore, the method must be able to recognize and analyze the occurrence of patterns that provide evidence for $k$-ary correlations \textit{whenever} they arise, and to analyze such patterns \textit{only} when they arise — rather than set up data structures for higher-order patterns that may or may not ever appear.

In other words, the exponentially-large multidimensional contingency table (described in Section 4.2.5 above) is in general going to be exceedingly sparse, and we seek an HOF discovery method that will exploit this sparseness.

4.5.1 The Basic Logic Behind the Method

The logical structure of the method is best motivated and understood by first considering the simple case of binary data.

Suppose there are $M$ sequences of bits, each sequence of length $N$. Consider a single pair of columns $c_1$ and $c_2$, each of which we shall treat, as usual, as a random variable. Loosely speaking, two such variables are positively correlated if they “turn on and off together” (have the same values of either 1 and 0 for a given input sequence) “very often”. They are negatively correlated if one of them is off when the other is on, very often. One can trivially extend this to
If one were to sample \( r < M \) sequences at a time, and found that very often a particular set of \( k \) variables showed exactly the same pattern of 1's and 0's over the \( r \) sequences, one would suspect the variables to be positively correlated. That is to say, *global similarity* (of \( M \)-length bit-strings) can be inferred from a large number of *local exact matches*. Positive correlation among binary variables is simply global similarity of their characteristic strings of values observed over the set of data records. This basic scheme was used by Williams and Kleinmuntz (1969) as an approximator for Pearsonian correlation over pairs (\( k = 2 \)) of strictly binary discrete variables. The larger is \( M \) and the more sampling is done, the more confidence we can have in our estimation of the correlation. These ideas can be made more precise by formulating a set of hypothesis tests that compare the *actual* number of times an \( r \)-sample produces exact matches among the \( k \) variables under consideration to the number of such exact matches *expected* if the \( k \) variables were independent. Resampling, counting exact matches, and simple, fast hypothesis testing provide the foundation for the discrete data-mining procedure presented in this section.

Before putting the simple ideas described above into a working procedure, they must be extended to handle non-binary discrete data. This is done simply by noting that for each column variable \( c_j \), the pattern of occurrence of any particular symbol \( a_i \) in that column is a binary vector. Thus, for example, the symbol \( B \) appears in column 2 in our toy dataset in sequences 1 and 5, and therefore the occurrence pattern string for attribute \( B @ 2 \) is 100010, a binary string with '1's in the first and fifth positions.

An exact match, as discussed above, for the symbol \( B \) in column 2 and the symbol \( D \) in column 4, is an instance of \( h \) simultaneous co-occurrences of these two *symbol@column* patterns. If these co-occurrence patterns are observed "suspiciously often" over many random \( r \)-samples, it suggests that the joint pattern \( (B @ 2, D @ 4) \) itself occurs suspiciously often, e.g., more often than would be predicted from the respective first-order (single-column) marginal probabilities, and therefore that the columns are correlated.

Thus, the basic procedure is organized as a series of iterations of sampling (subsets of \( r \) sequences), binning (the attributes with identical occurrence strings), and testing (observed match frequencies against expected match frequencies).
4.5.2 Definitions

Some definitions are necessary to make the above ideas more concrete and to help describe and analyze the algorithms used.

Let us define an attribute as a single symbol@column feature.

Define a coincident set, or cset to be a pattern comprising the joint appearance of \( 1 \leq k \leq N \) attributes.

An \( r \)-sample is a set of \( r \) of the \( M \) records (protein sequences) drawn at random, and we assume\(^3\) that the \( M \) sequences are independent and identically distributed (i.i.d.)

A cset has, for a given \( r \)-sample, a particular incidence vector, which is its binary-encoded record of occurrences (denoted by ‘1’) and non-occurrences (‘0’) over the \( r \) data items in the sample.

A match (or coincidence) of size \( h \) is said to occur, in a given \( r \)-sample, for a given cset \( \alpha = (a_1^{\alpha} \otimes c_1^{\alpha}, \ldots, a_k^{\alpha} \otimes c_k^{\alpha}) \), when \( a_1^{\alpha} \otimes c_1^{\alpha} \) appears in \( h \) out of the \( r \) records, \( \ldots \), and \( a_k^{\alpha} \otimes c_k^{\alpha} \) appears in \( h \) out of the \( r \) records, and they all appear in exactly the same \( h \) out of \( r \) records.

The reader will note that a match means that each attribute in the collision occurs only when the other attributes do, and that therefore the marginal frequency for each attribute is exactly the same as for all the others, and exactly the same as for any joint attribute of any arity \( n \leq k \), within the collision, within the \( r \)-sample.

4.5.3 Outline of Procedure

With the definitions in hand, and before delving further into the mathematics behind the method, we summarize the four basic components of the method:

- **Representation:** The occurrences of an attribute in a set of data items are summarized in a binary incidence vector.

- **Sampling:** Take \( r \) sequence records at a time, from a uniform distribution.

- **Binning, and Coincidence Detection:** Throw the attributes into bins, according to their incidence vectors. These vectors act like \( r \)-bit addresses into a very sparse subset of \( 2^r \)

---

\(^3\)As noted elsewhere, this assumption is of course false. The members of a protein or RNA or DNA family are by definition related to each other in ways described by a phylogenetic tree. Further research in this area should test the effects of the independence assumption, and/or attempt to build the phylogenetic relationships into the column-correlation modelling.
address space. All the attributes in one bin constitute a cset. Record the cset and the number \( h : 0 \leq h \leq r \) of occurrences.

- **Hypothesis Tests:** Compare the observed number of occurrences of each coincidence with the number expected under the null hypothesis of statistically independent columns.

In practice, since we seek a fast procedure and are willing to accept heuristic assumptions and sub-optimal accuracy, there is much room for flexibility and compromise in both the form of the hypothesis testing and its placement within the overall procedure. We note again that the goal is a data-mining method, and so the emphasis is on discovery of interesting correlations, rather than on precise quantification of their magnitude or statistical significance. We therefore want only as much hypothesis testing as is needed to find the correlations.

### 4.5.4 Expected and Actual Frequencies of Coincidences

The coincidence detection method works by finding coincidences among the occurrences of individual symbol@column features, and deciding whether such coincidences are suspicious, that is, whether the observed number of such coincidence occurrences is greater than expected. Here we define what is expected and what is observed.

#### 4.5.4.1 Expected Size \( h \) of a Match

The basis for the "expected" part of the hypothesis test is the probability of a match, or coincidence, of size \( h \) in a given \( r \)-sample for a cset \( \alpha = (a_1^\alpha \oplus c_1^\alpha \ldots \ldots a_k^\alpha \oplus c_k^\alpha) \). We begin by defining it for the simple case of two attributes, \( A \oplus c_1^\alpha \) and \( B \oplus c_2^\alpha \), which for simplicity will be denoted \( \alpha = (A@1, B@2) \) or just \( (A, B) \).

The probability of an exact match of size \( h \) for \( (A, B) \) given a draw of \( r \) of the \( M \) sequences may be modelled by the function

**Definition 4.1**

\[
f_{match}(\alpha, h, r) = \frac{r!}{h!(r-h)!}p(A, B)^h p(\hat{A}, \hat{B})^{r-h},
\]

where \( \hat{A} \) refers to the occurrence of a symbol other than \( A \) in column 1, and similarly for \( \hat{B} \) in column 2.
This is obtained from the multinomial distribution:

\[
\frac{r!}{z_1!z_2!z_3!(r - z_1 - z_2 - z_3)!} p(A, B)^{z_1} p(\tilde{A}, \tilde{B})^{z_2} p(A, \tilde{B})^{z_3} p(\tilde{A}, B)^{r - z_1 - z_2 - z_3},
\]

which gives the probability for the possible outcome of finding \(z_1\) occurrences of \((A, B)\), \(z_2\) occurrences of \((\tilde{A}, B)\), \(z_3\) occurrences of \((A, \tilde{B})\), and \((r - z_1 - z_2 - z_3)\) occurrences of \((\tilde{A}, \tilde{B})\) in a drawing of \(r\) sequences. However, in a match we have, by definition, \(z_2 = z_3 = 0\) because \((A@1)\) and \((B@2)\) only occur together.

The general case, for \(k \geq 2\), requires the same simple form for the \(f_{\text{match}}()\) function, because all but two of the larger number of \(p()\) factors in the multinomial expression for the general case vanish once again with zero exponents:

**Definition 4.2**

\[
f_{\text{match}}(\alpha, h, r) = \frac{r!}{h!(r-h)!} p(a_{i_1} @ c_{j_1}, \ldots, a_{i_k} @ c_{j_k})^h p(\tilde{a}_{i_1} @ c_{j_1}, \ldots, \tilde{a}_{i_k} @ c_{j_k})^{r-h}).
\]

The probability of a match of size \(h\) for the \(k\) attributes which make up a potential cset has been defined in terms of the joint probability \(p(a_{i_1} @ c_{j_1}, \ldots, a_{i_k} @ c_{j_k})\): but the point of all of this is to test the hypothesis that the columns are independent, and therefore this hypothesis should be built into the use of the \(f_{\text{match}}()\) function by substituting \(\prod_{l=1}^k p(a_{i_l} @ c_{j_l})\) for \(p(a_{i_1} @ c_{j_1}, \ldots, a_{i_k} @ c_{j_k})\) and \(\prod_{l=1}^k (1 - p(a_{i_l} @ c_{j_l}))\) for \(p(\tilde{a}_{i_1} @ c_{j_1}, \ldots, \tilde{a}_{i_k} @ c_{j_k})^{r-h}\). Of course, if one wants to test against other null hypotheses, that is easy enough to do.

For example, if \(p(A@1) = 0.2\) and \(p(B@2) = 0.1\), and if \(r = 4\). then (rounding to five decimal places) we obtain:

\[
\begin{align*}
    f_{\text{match}}(\alpha, 1.4) &= 0.029860 \\
    f_{\text{match}}(\alpha, 2.4) &= 0.001244 \\
    f_{\text{match}}(\alpha, 3.4) &= 0.000023 \\
    f_{\text{match}}(\alpha, 4.4) &= 0.000001 \\
    f_{\text{match}}(\alpha, 0.4) &= 1 - \sum_{j=1}^4 f_{\text{match}}(j)
\end{align*}
\]

\(^1\text{Because our } r\text{-sampling is sampling "without replacement", it would be more technically correct to use a hypergeometric distribution. This would not significantly affect either the results or the computational complexity of the method in most cases. For the sake of simplicity of exposition, we use the multinomial approximation, which is known to behave reasonably when } r \ll M.\)
and the expected size \( E_{\{\alpha\}} f_{\text{match}}(\alpha, h, r) = \sum_{j=0}^{4} f_{\text{match}}(\alpha, j, r) j = 0.032418 \). Here \( E_{\{h\}} \) denotes the expected value as averaged over the possible values of \( h \).

### 4.5.4.2 The Observed Sizes of Matches

We now know how to estimate the expected size of a coincidence for any given cset composed of \( k \geq 2 \) attributes. Here we consider how to count the observed coincidences and record their sizes for use in explicit or implicit hypothesis tests.

One can implement a procedure which collects \( T \) \( r \)-samples, for each of them binning the attributes that occur within the sample, and recording the coincidences that occur (where each coincidence represents a set of \( k \) attributes that are found within the same bin). For each such cset found, a data structure will be created which records the identities of the attributes and the size \( h \) of the coincidence. If additional matches are found for the cset in subsequent \( r \)-sampling iterations, then the new count of “hits” may be added to the running total stored in the cset’s record.

Exactly which kinds and amounts of information should be stored for each discovered cset within each iteration depend on exactly which kind of hypothesis testing is to be performed. This is discussed in the next section.

### 4.5.5 Hypothesis Tests

The hypothesis testing per se in a data-mining procedure should be minimal and fast. Elaborate verification of the “interestingness” of discovered features should be the responsibility of whichever hypothesis testing or comprehensive probabilistic modelling methods are to be used after this preprocessing stage.

In accord with this general principle, one of the many possible combinations of (a) expected match-size estimate, (b) observed match-size statistic, and (c) hypothesis test are described below.

Suppose that, for each discovered cset \( \alpha \), for each iteration \( i \) of the \( r \)-sampling, the procedure stores the count \( h_{\alpha,i} \) (let us call it simply \( h_i \), with the cset assumed) of occurrences, that is, the number of 1’s in the incidence vector. \( h_i \). At the end of the sampling, one has a total number of matches \( h_{\text{total}} \) for each stored cset.
How does one go about deciding whether or not to accept a set as suspicious, on the basis of \( T \) and \( h_{\text{total}} \)? A Chernoff bound on tail probabilities provides a reasonable answer.

Let random variable \( X_i \) hold the value \( h_i \) for each iteration \( i \), and let \( X = \sum_{i=1}^{T} X_i \), and note that \( 0 \leq X \leq T \cdot r \). The method of Chernoff-Hoeffding bounds (Hoeffding, 1963) provides the following theorem:

Let \( X = X_1 + X_2 + \cdots + X_n \) be the sum of \( n \) independent random variables, where \( l_i \leq X_i \leq u_i \) for reals \( l_i \) ("lower") and \( u_i \) ("upper"). Then

\[
P[X - E[X] > \delta] \leq \exp\left(\frac{-2\delta^2}{\sum_i (u_i - l_i)^2}\right). \tag{4.2}
\]

For our purposes, we set \( n = T \) and \( l_i = 0 \) and \( u_i = r \) for all \( i = 1, 2, \ldots, T \), and we thereby obtain

\[
P[X - E[X] > \delta] \leq \exp\left(\frac{-2\delta^2}{Tr^2}\right). \tag{4.3}
\]

Given this relation, it is easy to choose a threshold \( \delta \) given \( T \) and a desired confidence level. (However, see the note on false negatives below.) Alternatively, given a desired confidence level and an observed difference between observed and expected coincidence counts, one can decide how many iterations are needed to achieve the desired confidence. This requisite value for \( T \) is the major aspect of the method’s data complexity.

For example, suppose for cset \( \alpha \) we have \( E[X_i] = 0.003 \) for all iterations \( i \). \( T = 1000 \) iterations of the \( r \)-sampling are performed, and \( r = 7 \). Hence \( E[X] = 0.003 \cdot T = 3 \), but we observe \( X = 402 \) for the final sum of match occurrences for \( \alpha \). The probability of this observation, given the independence assumption for the component attributes of \( \alpha \), is bounded by

\[
P[X - E[X] > 402] \leq e^{-\frac{402^2}{3^2}} = e^{-6.596}. \tag{4.4}
\]

Hence the probability is on the order of 0.00137.

The Chernoff-Hoeffding formulation of our hypothesis tests also provides simple insights into the sample complexity of the procedure. Suppose some desired confidence level \( P^* \) is fixed in advance: by algebraic rearrangement of Inequality 4.3 given above, and by using the fact that \( E[X] = T \cdot E[h] \), we obtain estimates for the necessary number \( T^* \) of \( r \)-samples:

\[
T^* = \frac{-\log P^* r^2 + 4X E[h] + (-\log P^* r^2(-\log P^* r^2 + 8X E[h])} {4E[h]^2}{1/2}. \tag{4.5}
\]
\( T^* \) therefore grows with the negative log of the confidence level, which in our formulation is given in terms of the likelihood of the observed data given the inter-column independence assumption. This logarithmic sample complexity is fairly typical for algorithms employing i.i.d. random sampling and tail probability bounds.

4.5.6 The Main Procedure

An outline of the basic coincidence detection procedure is given below, and a pictorial representation of the main steps in the algorithm is presented in the figure on Page 139.

Procedure to find suspicious coincidences:

0. begin
1. read(\texttt{FAMILY});
2. read(\texttt{R, T});
3. compute\_first\_order\_marginals(\texttt{FAMILY});
4. csets := { }
5. for \texttt{iter} = 1 to \texttt{T} do
6. \quad \texttt{sampled\_family} := \texttt{rsample(\texttt{R,FAMILY})};
7. \quad attributes := \texttt{get\_attributes(sampled\_family)};
8. \quad all\_coincidences := \texttt{find\_all\_coincidences(attributes)};
9. \quad for \texttt{coincidence} in \texttt{all\_coincidences} do
10. \quad \quad if \texttt{cset\_already\_exists(coincidence, csets)}
11. \quad \quad \quad then \texttt{update\_cset(coincidence, csets)};
12. \quad \quad \quad else \texttt{add\_new\_cset(coincidence, csets)};
13. \quad \quad endif
14. \quad endfor
15.* csets := \texttt{cull\_uninteresting\_csets(csets): /* Optional */}
16. endfor
17. for \texttt{cset} in \texttt{csets} do
18. \quad expected := \texttt{compute\_expected\_match\_frequency(cset)};
19. \quad observed := \texttt{get\_observed\_match\_frequency(cset)};
20. \quad \texttt{stats} := \texttt{update\_stats(cset, hypoth\_test(expected, observed))};
21. endfor
22. print\_final\_stats(csets, stats);
Figure 4.6: Operation of the Coincidence Detection Method: Three iterations of the $r$-sampling (for $r = 3$) on the toy dataset are depicted, top to bottom. For each iteration, the left-hand box represents the dataset, with outlined entries representing the sampled rows. The right-hand box represents the set of bins into which the attributes collide. For example, in the first iteration, $A@1$, $B@2$, and $D@4$ all occur in the first and second of the three sampled rows, so they each have incidence vector 110 and collide in the bin labelled by that binary address. Bins containing only a single attribute are ignored; and “empty” bins are never created at all. All bins are cleared and removed after each iteration, but collisions are recorded in the $Csets$ global data structure.
4.5.7 Computational Complexity of the Method

The coincidence-detection procedure described above is a heuristic and probabilistic approximation algorithm. Like all such algorithms, it does not lend itself to easy analysis, because many of the operations depend on the particular distribution of objects and symbols in the dataset. In this section, we give bounds on the time, space, and data complexity of the method in terms of two distribution-dependent parameters $L$ and $T$. We then present derivations and arguments for particular estimates of the expected values of those crucial parameters under some very broad distributional assumptions. Ultimately, however, precise analyses of the behaviour of the method can be formulated only from empirical testing on a wide range of real and synthetic datasets from the various application domains. Some such testing is reported in Section 4.6 below.

4.5.7.1 Overall Asymptotic Complexity

Reading the dataset and estimating the first-order marginal probabilities (lines 1-3 of the algorithm as depicted above) takes $O(MN)$ steps.

The first main loop, which iterates over the $r$-samples, is repeated $T$ times. Within the loop, we have: the drawing of the sample (line 5) is $O(rN)$ and the binning of the attributes (lines 6, 7) is $O(rN)$; the search and update of the stored csets table on the $i$th iteration depends on $L_i$, the current size of the table. The expected size and growth rate of this table is discussed below. This table may be stored in any of a number of ways: an optimal storage method would take maximal advantage of the expected size, the expected number of lookups and updates, and the degree of computer memory available. It is not unreasonable to consider hashing and tree-structure schemes which would limit the lookup time to $O(1)$ or at worst $O(\log L_i)$.

Finally, there is the loop beginning at line 16, which processes the $O(L_T)$-size table of all distinct csets stored throughout the sampling and coincidence-detection. Each of the operations here – computing expected match-count (line 17), retrieving observed match-count (line 18), and computing a tail probability estimate as a hypothesis test (line 19) – can be done in essentially constant time (or, at worst, $O(N)$ for with a small linear factor if there are many very-wide csets). Line 22 performs the final printout of results, meaning that some or all of the $O(L_T)$ csets must have their statistics reported.

Therefore, a reasonable estimate of the overall asymptotic time complexity of the method is $O(MN + T(rN + \log L_*) + L_T)$, where $L_*$ is the average size of the csets table over the $T$
iterations. The space complexity can be estimated at $O(MN + L)$.

The numbers $M$ and $N$ are given, and typically fixed, for the particular application. The more interesting components in the complexity estimates are $L$ (shorthand for $L_*$, $L_T$ and all of the $L_i$) and $T$. $L$ depends crucially on the distribution of symbols in the dataset. $T$ depends on the distribution and on the user-defined desired levels of accuracy, as well as on the number $M$ of data objects available.

We explore the possible ranges of $L$ and $T$ in the following sections.

4.5.7.2 Estimating $E[L]$, the Expected Size of the Cset Table

Unfortunately, we find intractable or, at least, beyond the scope of this project, the task of specifying and proving tight and meaningful theoretical bounds on the size and growth rate of $L$. We can estimate the probability of, or give good tail bounds for, any particular $r$-sample submatrix of size $N \cdot r$ symbols, given some assumption about the data distribution. But to estimate some sort of statistical expectation, for some "average" $r$-sample scenario, appears to be intractable. It is possible to give uninteresting worst-case upper bounds based upon how many possible $k$-ary joint attribute occurrences, for all possible values of $k$, can be fit into the sub-matrix corresponding to a single iteration of the procedure, and so on. However, such bounds are exponential in $N$, which is no surprise at all. Of real interest are the expected sizes of $L$ because, in practice, on "reasonable" data distributions, one observes a kind of saturation effect: in the useful ranges of $T$, all of the csets that are likely to occur occur early and are stored in the cset table, after which they are just updated. The cset table does not grow exponentially. However, without the ability to find closed-form expressions for all possible ways to fill in the $N$ by $r$ sub-matrix under different distributional assumptions, we are left with no alternative to empirical estimates from simulations. Fortunately, our empirical analyses, as described in Section 4.6.4, show that $L$ grows not just sub-exponentially but sub-linearly as a function of iterations $T$ and as a function of the number of attributes $N_A$.

4.5.8 Analysis of Error in the Method — Types, Probability, and Bounds

There is a complex space of tradeoffs linking the size $r$ of samples, number $T$ of samples, and the relative risks of the different types of error. First, intuitively: the larger the number $T$ of samples, the smaller is the risk of overlooking the occurrences of a particular joint symbol occurrence. The smaller the sample size $r$, the smaller is the risk of this kind of "false negatives" risk, too.
In fact, if $r = 2$, one can use an ANDing method and be guaranteed to find all potentially interesting joint symbol occurrences, with all interesting subcomponents. Of course, a high level or small granularity of sampling has a cost: Large $T$ and small $r$ raise the expected time and space complexity because they increase the expected size of the stored table of provisionally accepted csets.

In order to explore these issues more rigorously, it is first necessary to understand the coincidence-detection procedure in terms of the three distinct levels of random sampling:

1. The database itself: We may consider the $M$ sequences to have been drawn from some larger underlying population of $M_{all}$ sequences. Alternative assumptions include finite $M_{all}$, infinite $M_{all}$, and the case wherein $M_{all} = M$ (the database is all there is).

2. The $r$-sampling: $r$ sequences are drawn, randomly and identically distributed, without replacement, from the $M$ sequences in the database.

3. The $T$ iterations: The outer loop of the procedure iterates over many $r$-sample events, and records information on the outcomes. In the currently described version of the algorithm, these $r$-sample draws are performed with replacement and i.i.d. The process is modelled by a multinomial distribution.

It is crucial to note that error may be injected at all levels of the procedure, and this greatly complicates any error analysis. In particular, the method requires and makes estimates of probabilities at the first and third levels of sampling: Estimates of first-order marginal probabilities of attributes are made at Level 1; and estimates of the total count of joint attribute coincidences are made at Level 3. In practice it is often convenient to make a few explicit simplifying assumptions, one of which is the treatment of the error at Level 1 as independent from error at Level 3. For example, in assessing the error in higher-order joint probability estimates due to the iteration of $r$-sampling and binning, one might assume that the single-attribute marginal probabilities input to the procedure are correct. In practice, our sampling procedures are as vulnerable to the problems of truly small sample sizes as any other estimation procedure; and it might be wise to look for proper Bayesian versions of some of our estimates and assumptions, in order to better overcome small-sample difficulties.
4.5.8.1 Error at Level 1

We consider here the probability of a situation wherein two attributes \(a_1\) and \(a_2\) are in fact correlated (or uncorrelated) yet appear in the finite \(M\)-sequence database to be uncorrelated (or correlated). That is, we seek to estimate \(P(\text{Corr}(a_1, a_2) | \text{Corr}(a_1, a_2))\), where \(\text{Corr}(a_1, a_2)\) is an estimate of correlation from database frequencies and \(\text{Corr}(a_1, a_2)\) is the true correlation value of the underlying variables. This estimate depends in turn on \(P(\hat{q}(a_1, a_2) | q(a_1, a_2))\) and \(P(\hat{q}(a_i) | q(a_i))\) for \(i = 1, 2\), where the \(\hat{q}(\cdot)\) are estimates and the \(q(\cdot)\) are true underlying probabilities.

These estimations are within the purview of classical and Bayesian hypothesis testing, and we refer the reader to a publication by Wolpert and Wolff (Wolpert and Wolf, 1993) for a review and analysis of the general problem. There are many ways to proceed with such estimation, and the choices in our context depend on particular assumptions regarding \(M_{all}\) and \(M\). We present here just one example of a useful result: The probability of spurious inter-column correlations in a truly random (maximum entropy) database.

Consider the database of \(M\) records and \(N_A = N|A|\) attributes. Suppose all attributes have the same probability \(q\) and all columns and attributes are independent. Thus the probability of any joint attribute \(\alpha\) is \(p(\alpha) = p(a_{i_1} \otimes c_{j_1}, a_{i_2} \otimes c_{j_2}, \ldots, a_{i_k} \otimes c_{j_k}) = q^k\). The expected number of records containing that joint attribute is given by the binomial distribution with parameters \(M\) and \(q^k\) to be \(E[X] = Mq^k\). A simple Chernoff bound gives us

\[
P[X > Mq^k + \delta] < e^{-\frac{\delta^2}{2M}}
\]  

(4.6)

Suppose we are using some global threshold \(\Theta\) for considering a joint attribute to be “interesting” (in our own search for “interesting”, the threshold would instead be a function of the different marginal and joint probabilities). That is, accept \(\alpha\) into this category only if the joint frequency \(\hat{q}(\alpha) \geq \Theta\). Then the probability of a particular set of \(k\) attributes falsely appearing to form an “interesting” joint attribute is bounded by

\[
P[X > M\Theta] = P[X > Mq^k + M(\Theta - q^k)] < e^{-2M(\Theta - q^k)^2}.
\]  

(4.7)

The expected number of “interesting” joint attributes is therefore bounded by
and this number is less than 1 if $\Theta > \sqrt{\frac{k \ln N}{M}} + q^k$.

The same estimate was given by Agrawal et al. (Agrawal et al., 1996).

### 4.5.8.2 False Negatives and False Positives at Level 3

The particular version of the coincidence-detection method described thus far is particularly prone to a specific kind of false negative error. While the expected number of coincidences for a given cset $\alpha$ predicts the number of all occurrences of the $\alpha$, the procedure's count of observed coincidences reflects only those instances where the component attributes of $\alpha$ occur alone together. That is, if $\alpha = (a_{i1} \oplus c_{j1}, a_{i2} \oplus c_{j2}, \ldots, a_{ik} \oplus c_{jk})$, then an observed bin containing just $\alpha$ gets counted towards $\alpha$'s occurrence total, whereas a bin containing $\alpha$ along with $\alpha' \oplus c'$ does not. Therefore, the support for $\alpha$'s "suspicousness" is underestimated.

There are several ways to address this problem of one cset "masking" another. One approach is to maintain not just a table of csets but also some of the hierarchical structure relating the different csets. For example, the entry for $(A \oplus 1, B \oplus 2, C \oplus 3)$ could contain a pointer to the entry for $(A \oplus 1, B \oplus 2)$, and vice versa. Then, whenever the count for a cset is updated, the procedure would also update the counters for its lower-order constituents. This scheme has worst-case exponential space complexity, but might not be unreasonably costly in practice. Further investigations should be made into performance in average-case situations.

One could instead adjust the $E[X]$ values by some factor derived from masking estimates. Note, too, that one could obtain better estimates of the degree of the masking effect after some number $T'$ of preliminary iterations. Perhaps setting $T' = T$ is reasonable, i.e., the adjustment could be made at the end of the $r$-sampling, if well-founded masking estimate is available.

Most simply, one could just perform more sampling in order to overcome the masking effect by sheer computational force. For a rough estimate of the severity of the problem, consider the following simplified scenario. A dataset with a cset $\alpha = (a_{i1} \oplus c_{j1}, a_{i2} \oplus c_{j2}, \ldots, a_{ik} \oplus c_{jk})$ and all other $n = N - k$ columns independent, is $r$-sampled for $T$ iterations. Suppose $\alpha$ occurs as a coincidence in a particular iteration. Hence it has a binary incidence vector of length $r$ with some number $h$ of 1's and $r - h$ 0's. Let us pretend for the sake of simplicity that we have only binary attributes (that is, $|A| = 2$ and there is one attribute for each column) and let us ignore attribute probabilities — this will not significantly affect the asymptotic results derived. The
probability that some other attribute \(a\) will have an incidence vector exactly matching that of \(\alpha\) is \(p_{\text{match}} = \frac{1}{2^r}\). The probability that it will not match is \(p_{\text{no}} = 1 - \frac{1}{2^r}\), and the probability that no such attribute \(a\) will co-occur exactly with \(\alpha\) in this \(r\)-sample iteration is therefore \(p_{\text{none}} = (1 - \frac{1}{2^r})^n\). Finally, the probability that \(\text{some}\) other attribute or attributes will coincide with \(\alpha\) in this iteration is

\[
p_{\text{some}} = 1 - \left[(1 - \frac{1}{2^r})^n\right].
\]  

(4.10)

This estimate is sobering; it means that the proportion of coincidences that we are blocked from observing, with the simple version of coincidence detection presented in this thesis, grows very fast as the number of columns grows. The particular constants used in our back-of-the-envelope calculations above may be wrong, but the underlying asymptotic relationships are correct and are illustrated in the figure on Page 4.7. Note that the expansion of \((1 - \frac{1}{2^r})^n\) followed by a few simple algebraic steps make it clear that the \(p_{\text{some}}\) is dominated by a term \(\frac{n}{2^r}\). This tells us, for example, that if we set \(r = 10\), then \(p_{\text{some}} \approx \frac{n}{1000}\), and so if \(n = 500\) variables, then a particular interesting cset may be missed in about one half of all samples. One would do well to use a bigger \(r\) value and/or perform more sampling.

**Figure 4.7:** Shown above is a 3D plot of the estimated probability of a typical cset (correlated \(k\)-tuple of attributes) being masked in a given \(r\)-sample, as a function of \(r\) and \(N\). Note that for higher values of \(r\) the probability of this particular kind of error is minimized.

The good news is that the rate of growth of this masking problem is dependent upon
the sample-size parameter $r$: Higher $r$ makes for less masking, and hence fewer extra samples are required to separate signal from noise. This relationship of result quality to choice of $r$ is apparent in our empirical testing thus far, though there seem to be other factors militating against arbitrarily large values of $r$. In our tests on the artificial data, and it seems, on the HIV data, the masking problem has not obstructed us unreasonably, for the values of $r$ and $N$ used.

Referring back to an important issue in the computational complexity analysis of the coincidence detection method, we emphasize here the following relationships. The level of sampling $T$ needed for a given confidence level can grow exponentially with increasing $N$ in the worst case; for good choice of $r$ and for the practical ranges of $N$ explored thus far, the procedure seems to remain in the relatively "flat" portion of the growth curves. Clever use of sub-sampling and block design can keep this problem manageable. We have also seen that the required $T$ grows linearly with the count of observed coincidences $X$ and grows only with the log of the desired confidence level $P$. Empirically $L$, the size of the Csets data structure, appears to grow logarithmically with each of $N$ and $T$ separately. Therefore, in the worst case, $L$ might grow worse than linearly — perhaps even exponentially — with the number of attributes when the masking problem requires exponentially more sampling. Such pathologically poor performance has not been observed thus far, and there seem to be several remedies available to avoid it.

An issue closely related to the $T,r,N$ and confidence/error tradeoffs is that of alphabet size $|A|$. If $|A|$ is small, then for given $T,r$ there are more coincidences, and hence lower risk of false-negative errors. However, to collapse an initial alphabet $A$ into a smaller $A'$ is to presume particular equivalence classes among the original $a_i \in A$; but this may introduce an undesirable bias into the resulting feature discovery process. One thing we would like to emerge from a feature discovery process is exactly this kind of amino-amino (in the protein domain) equivalence relationship, but the "true" or "useful" equivalences may not be those given a priori by the experts, and they may not be static across all protein data but rather very family-dependent and position-dependent (Gribskov, Luethy and Eisenberg, 1990; Nakai, Kidera and Kanehisa, 1988).

We intend to explore all of these tradeoffs in much greater detail in future work.

### 4.5.9 Finding Other Correlated Attributes

We mention here an obvious but interesting and potentially very valuable extension of the method. There is no reason why the dataset input to the coincidence detection algorithm
must be limited to amino acid residue attributes. Given the growing trend towards integrated bioinformatics databases, it is likely that one would want to add attributes reflecting other structural, functional, genomic, medical, or historical features to each protein’s record, and to want to find associations among any \(k\)-tuples of features. Suppose, for our same toy dataset used throughout the paper, that sequences (rows) 1, 2 and 6 are known to be in a special functional \(\text{class}_1\), and the rest are in \(\text{class}_0\).

\[
\begin{array}{cccccccc}
\text{col1} & \text{col2} & \text{col3} & \text{col4} & \text{col5} & \text{col6} & \text{col7} \\
A & B & C & D & E & F & 1 \\
W & U & C & V & E & G & 1 \\
Z & L & C & M & W & M & 0 \\
V & U & C & V & A & G & 0 \\
A & B & C & D & Z & Z & 0 \\
W & L & C & M & E & Z & 1 \\
\uparrow
\end{array}
\]

Example 4.2

Now the subclass designation appears as just another attribute, and the coincidence set method may be run on this augmented dataset. In addition to finding the residue positions which co-occur with interesting frequency, one can now find the residues that are correlated with the particular subclasses. (In this case, it seems the \(\text{class}_1\) sequences are distinguished by the appearance of \(E\) in column 5.)

4.6 Coincidence Sampling: Experiments and Results

Because theoretical estimates and bounds for the Coincidence Sampling methods, as for many probabilistic methods, are so contingent upon fairly detailed assumptions about data distributions, it is important to test the method empirically. It is particularly useful to test the method on datasets that have been explicitly constructed and are known to possess specific distributions, so that over time a clear picture will emerge of the method’s strengths and weaknesses.

4.6.1 Tests on Specially-Constructed Datasets

A suite of programs was designed and implemented to generate datasets with the general form of protein sequence families and with particular kinds and degrees of higher-order inter-column
correlations built into it.

Two such datasets were generated and studied extensively, and these studies are summarized below.

4.6.1.1 General Form of the Experiments

For each of the two experiments, a dataset was generated from an alphabet of the twenty standard amino acid symbols, with relative baseline single-column frequencies based on statistics for the Protein Information Resource (PIR) database. Columns chosen to display conserved physico-chemical properties were generated from special subsets of \( \mathcal{A} \), such as Hydrophobic, PositiveCharged, and so on. Each dataset consists of a set of mutually-independent columns and small cliques of mutually-dependent columns. The dependencies were designed to mimic plausible if idealized examples of real structural constraints as well as to test the general detection capabilities of the coincidence detection procedures.

In each experiment, the procedure was run for both \( T = 500 \) and \( T = 10,000 \) iterations of sampling and binning, and for sample-sizes of \( r = 5, 6, 7, \) and 10.

4.6.1.2 Family0 Dataset

Family0 consists of \( M = 200 \) rows (sequences) and \( N = 25 \) columns (residues), and therefore \( N_A = 25 \cdot |\mathcal{A}| = 500 \) possible attributes. All columns were generated independently, from several different property distributions, except for:

- Columns 1 through 4 covary perfectly, so that the first four symbols in each row are either ARND, CQEG, HILK, MFPS, or TWYV, identically distributed with \( p(\text{each}) = 0.2 \).

- Columns 5 and 6 represent an imperfect conservation of net hydrophathy, with 80% of the sequences displaying either Hydrophobic then Hydrophilic, or the reverse. The other 20% of sequences drew two symbols randomly and independently from the union of the two above-named subsets.

4.6.1.3 Family1 Dataset

Family0 consists of \( M = 200 \) rows (sequences) and \( N = 20 \) columns (residues), and therefore \( N_A = 400 \) potential attributes. All columns were generated independently, from several different property distributions, except for:
- Columns 4 and 5 represent a perfect conservation of net charge, and were generated from a uniform distribution over Positive-Negative and Negative-Positive joint symbol-pairs.

- Columns 7 through 11 represent a very subtle conserved constraint: Noisy, distributed conservation of total side-chain volume. In each sequence, four of these five positions were filled by an amino acid from the Small subset, while the fifth position (chosen randomly from uniform distribution over the five positions) was filled by an amino acid from the general distribution over A. (Hence, this fifth position could be a small, large, or medium-sized amino acid.)

- Columns 19 and 20 represent an imperfect conservation of net hydropathy, generated from the same process as used for the similar pair of columns in Family0 as described above.

4.6.2 Results of Family0 and Family1 Experiments

Tables 169 through 172 report the results from coincidence detection runs on the Family0 data.

The order of presentation of the tables is made with respect to the different combinations of parameter values, as follows: \( r = 5, 6, 7, 10 \), respectively: and for each value of \( r \) the two tables for \( T = 500 \) and \( T = 10,000 \) are shown.

In each table, an ordered list of the csets with the lowest estimated probabilities of coming from independent attributes. That is, all csets \( \alpha = (a_{i_1} \bar{c}_{j_1}, \ldots, a_{i_k} \bar{c}_{j_k}) \) such that \( P(Observed(\alpha) \mid Independent(a_{i_1} \bar{c}_{j_1}, \ldots, a_{i_k} \bar{c}_{j_k}) < 0.9 \) are shown, where \( Observed(\alpha) \) is the observed count of exact matches (collisions) for the component attributes of \( \alpha \), in increasing order with respect to this estimated probability. Hence these are the \( k \)-tuples of attributes most likely to be correlated, given the data and the particulars of the procedure.

4.6.2.1 Family0 Results

The tables show clearly that the coincidence detection procedure has no trouble detecting exactly the csets corresponding to the higher-order column correlations designed into this artificial dataset. Even at only \( T = 500 \) iterations, and even for the lowest value of \( r \) tested, \( r = 5 \) (see Table 169), the results demonstrate an obvious separation in probability values between the truly correlated and spuriously correlated attributes. The probability values for the features from correlated cliques \((1, 2, 3, 4)\) and \((5, 6)\) all fall below \( P = 0.45 \), while all other csets had values \( P > 0.9 \) (and hence do not appear in the table). These results from the worst table, rep-
resenting a relatively small number of sampling steps, are not uniformly impressive — $P \leq 0.45$ does not make for a good tail probability bound — but the separation of signal from noise is nonetheless evident.

Much better results are obtained with slightly higher $r$ values and, of course, with greater numbers of iterations. For $r = 7$ and $T = 10,000$ (Table 171, we find $P < 10^{-6}$ for all of the appropriate joint attributes and $P > 0.85$ for all others.

A back-of-the-envelope calculation here can shed further light on some of the tradeoffs faced in comparing our method with some other methods. In obtaining the results noted in the paragraph above, with $T = 10,000$ iterations of $r$-samples, where $r = 7$, over $N = 25$ columns, we used roughly $T \cdot r \cdot N = 1.75$ million primitive computational steps in finding those significant 4th-order features. In contrast, if we had known that we were looking for significant 4th-order features (and not 3rd-order or 14th-order features), then we could have simply computed estimates of all $25^4/4!$ 4th-order joint probabilities, compared with products of the appropriate marginals, and printed out the highest discovered deviations from independence. That operation could be done in $25^4/4! \cdot M \cdot T'$ steps, where $M = 200$ is of course the number of rows and $T'$ is the number of sampling iterations needed to achieve suitable confidence in the joint and marginal probability estimates. Thus, roughly $3.2$ million $T'$ steps are required, and therefore, even if we allow $T' = 1$, the simplistic approach appears slower than our method, and the simplistic approach can only detect features of order $k \leq 4$. in contrast with our method. It must be noted that these calculations ignore the “overhead” of costs in our algorithm, such as storing and retrieving cssets from a global hash table: nonetheless, the comparison is interesting.

4.6.2.2 Family1 Results

Examining Tables 173 through 178, one sees a pattern similar to that observed in the Family0 results. Again, most highly correlated attribute $k$-tuples are picked out reasonably easily, with best results at parameter settings $r = 7$, $T = 10,000$. However, even in the best runs reported, the procedure failed to separate cleanly the entire set of cssets from the correlated columns versus those representing spurious correlations: While all of the appropriate cssets for (19,20) are clearly detected by their $P < 0.001$ values, only some of the occurring cssets for the column-pair 4,5 appear less probable than all spurious attribute combinations. This is explained by the higher joint entropy of the attribute distribution for (4,5). More iterations are required for our sampling and hypothesis testing to establish statistically significant deviations for the
(K@4, E@5), (E@4, K@5), and so on. Still, the detection of any such cset for a set of columns is evidence of correlation, and the detection of correlated columns is the main combinatorial challenge facing data-mining procedures like ours.

Of more concern, if one looks to apply the procedure to protein sequence data, is the procedure's utter inability to detect the more subtle correlation built into columns 7 through 11. Clearly the particular mechanism of exact match counts is ill-suited to the fast ($T \leq 10,000$) detection of the kind of higher-order distributed covariation posited for some models of evolutionary conservation of volume. It remains to be seen, however, whether any other method could handle such a challenge. The fundamental problem is that the distributed covariation can mean lower $k$-ary joint attribute frequencies for all $2 \leq k \leq 5$ than would be found in non-distributed conservation patterns.

4.6.3 Tests of Time and Space Complexity on Independent-Column Datasets

As discussed in previous sections, it is crucial to find empirical estimates of the size and growth rate of $L$, the size of the Csets data structure in our coincidence detection procedure. This is because, first, the expected time and space complexity of the method depend on $L$, and second, because tight and meaningful theoretical bounds seem impossible to obtain.

We therefore performed a set of experiments to observe how the number of stored csets grows as a function of the number of iterations $T$ and the number of variables $N$ and attributes $V$. An attempt was made to observe in particular the worst-case behaviour, with the assumption that $L$ is maximal when the columns in the dataset are all mutually independent. (Dependencies imply redundancy, meaning that the same collisions occur often: whereas for independent columns the attribute collisions occur haphazardly and we would therefore expect more distinct collisions to occur and hence more csets to be stored over the many iterations of sampling and binning.)

Four datasets of independent columns were generated and tested. The datasets had 50, 100, 200, and 500 columns: this translates to 1000, 2000, 4000 and 10,000 possible attributes, respectively. Each dataset contained 200 rows (artificial amino acid sequences), and each column in every dataset was generated independently from the PIR amino acid distribution.
4.6.4 Results of Experiments on the Size of L

A plot is shown on the figure on Page 152 for the growth of $L$ as a function of $t$, for $N = 200$, $N_A = 4000$, $r = 7$ and $T = 10,000$ iterations. A linear function, interpolated from a $t \approx 100$ neighbourhood within the run, is also plotted for reference and comparison with the plot of $L(t)$.

The important result shown by the figure is the sub-linear growth of $L$. One might well imagine worst-case scenarios involving an exponential explosion of csets. Not only is the growth in csets better than exponential, it is better than linear. This would seem to bode well for practical application of coincidence-detection procedures.

As expected, the cset-growth functions for these probably worst-case datasets are closer to linearity than we observe for the less pathological datasets. The reader may check this by examining the figures. Pages 153 and 153, which pertain to the Family1 and HIV datasets.

Another important result is the sub-linear growth of $L$ as a function of $N$, the number of columns. The figure on Page 154 illustrates clearly that exponential blowup and even "linear blowup" scenarios are overly pessimistic.

![Csets Growth, for Dataset of 200 Independent Columns, $r=7$](image)

Figure 4.8: This plot illustrates $L(t)$, the size of the Csets data structure in our coincidence detection procedure, as a function of the number of $r$-sample iterations, for a dataset consisting of 200 independent columns. The parameter setting $r = 7$ was used. A linear plot is shown for reference.

However, in earlier sections we addressed a noise problem to which the simplest coincidence detection procedure is prone. a problem that might in the very worst case require exponentially
Figure 4.9: Plotted above is $L(t)$ for the Family 1 dataset. The parameter setting $r = 7$ was used. A linear plot is shown for reference.

Figure 4.10: Plotted above is $L(t)$ for the HIV dataset. $r = 7$ was used for this experiment.

high levels of sampling to overcome as $N$ grows very large. This would obviously affect estimates for the growth of $L$ as a function of the growth of $N$ and $T$ together. (In practice, and possibly always if the right values of $r$ are chosen, the problem is quite manageable.)
Figure 4.11: Plotted above is $L(N)$ for $N$ Independent Columns. Sub-linear growth of $L$, the number of stored csets, as a function of $N$, the number of column variables, is observed when the number of iterations $T = 10,000$ is held constant.

4.6.5 Tests on an HIV Protein Database

The Los Alamos HIV Database\(^5\) contains, among other things, the amino acid sequences for the V3 loop region of the HIV envelope proteins. This region is known to have functional and immunological significance, and the discovery of sets of sites linked by evolutionary covariation might have important implications for understanding and preventing HIV infection and replication.

An earlier and smaller version of the same database was used by Los Alamos scientists (Korber et al., 1993) in their analysis of pairwise mutual information between residues (columns).

Experiments were performed on our HIV dataset with the coincidence detection procedure, over a set of different values for $r$ and $T$. Tables of results are shown in Section C and discussed below.

---

\(^5\)The database is maintained by many staff members at Los Alamos National Laboratories, Los Alamos, New Mexico. It can be accessed on the Worldwide Web at http://hiv-web.lanl.gov/.
4.6.6 Results of Experiments on HIV Protein Database

We edited our copy of the aforementioned version of the HIV-V3 dataset in order to focus on the thirty-three residues considered most conserved and most structurally and functionally important by the Los Alamos researchers. Our dataset therefore consisted of \( M = 657 \) rows (sequences) of \( N = 33 \) columns (residues). For the coincidence detection procedure, these 33 columns are transformed into \( N_A = N \cdot |A| = 33 \cdot 21 = 693 \) attributes. As with the artificial datasets, a set of experiments with different values of \( T \) and \( r \) were performed. Coincidence detection runs were done with \( T = 10.000 \) and \( r = 5.6.7.10 \) respectively, and with \( T = 100.000 \) and \( r = 7 \), and finally with \( T = 750.000 \) and \( r = 7 \).

Tables on pages 180 through 182 illustrate the most significant csets (again measured by our procedure's estimation of \( P(Observed|Independence) \)) for the Observed number of coincidences for each detected coincidence of attributes. As one might expect, a clean separation between "probably correlated" and "probably uncorrelated" does not manifest itself at this comparatively low degree of sampling for this real-world dataset. As before, results for \( r = 7 \) and \( r = 10 \) indicate more significant discovered csets than those for \( r = 5 \) and \( r = 6 \). At these former, higher \( r \) values, one sees the emergence of a few csets with "Prob" values less than 0.1: \( (Q@17, D@24) \), \( (N@4, K@9) \), \( (H@12, A@18) \), \( (Q@31, H@23) \) and \( (S@10, F@19) \). All of these csets appear among the most significant csets reported in the more intensive sampling runs (with \( T = 100.000 \) and \( T = 750.000 \)), with the notable exception of \( (S@10, F@19) \). This latter cset is discovered at this low degree of sampling only in the \( r = 10 \) run, and does not appear in the more intensive sampling runs shown, both of which used \( r = 7 \). This further evidence of the sensitivity of coincidence detection to the value of parameter \( r \) — at least at comparatively low degrees of sampling — suggests further research is needed to clarify this issue.

Table 183 displays the results for \( T = 100.000 \) and \( r = 7 \). and here it is clear that some separation of signal from noise is taking place amongst the set of HOFs, with seventeen pairwise and three 3-ary correlations appearing within our \( Prob \leq 0.1 \) significance level.

At \( T = 750.000 \), we have more statistically significant detection of almost fifty 2-ary, 3-ary, and up through 6-ary attribute correlations, as shown in Tables on pages 184 and 185.

In order to get a better sense of the possible meanings of these results, let us consider these inter-attribute correlations along with some inter-column correlations in the form of pairwise mutual information estimates performed in our own analysis and also by the Los Alamos group (Korber et al., 1993). Table C.8 displays the highest estimated mutual information values
amongst all \( \frac{N(N-1)}{2} = 528 \) pairs of columns from our 33-column dataset. The estimates were obtained using a Bootstrap-like procedure (Efron, 1983; Efron and Tibshirani, 1991) in which 1000 sample data subsets of \( m = 300 \) out of \( M = 657 \) were drawn (i.i.d.) and run though the standard mutual information calculation. Reported in the table are therefore the mean values over the resampling and the associated standard error values. The reader will note the significant intersection between the set of column-pairs indicated by the top csst values in Tables 154 and 155 and those indicated by the top mutual information values in Table 2.8. The correspondence between the two rankings is not perfect, for a few reasons (besides noise and simple sampling error). First and foremost, while the "suspiciousness" of a single joint-attribute combination certainly contributes to the mutual information within the corresponding set of columns, the behaviour of the other symbols appearing within the columns obviously also can have great effect. Second, we note again the observed sensitivity coincidence detection results to the choice of \( r \), something that probably suggests the need for combining multiple runs at a few different \( r \) values and that certainly suggests further research and testing.

Table C.9 lists the highest statistically significant mutual information values as estimated by the Los Alamos group. We note the overlap between their list and ours, but we emphasize again that group’s use of an earlier, smaller, and perhaps otherwise different database to which we did not have access.

4.7 Comparison of Coincidence Detection with Other Methods

Having presented the coincidence detection method and some illustrative results of its application, we may now compare the method with other methods of discrete HOF-detection, in the protein science domain and more generally.

4.7.1 Protein Family Modelling

First-order methods, which work only with individual residue (single-column) marginal probabilities, can generate only suboptimal classifiers. It is easy to construct hypothetical cases, and not too difficult to find a few actual examples, wherein protein sequence classification can fail when only first-order features are represented and used. Whether this limitation has seriously hampered major computational biology efforts to date is open to question. More clear is the effect of the first-order limitation on the discovery of structural and functional motifs. Biological
macromolecules are characterized by their internal bonded and non-bonded interactions, and these interactions escape detection by first-order analyses of sequences.

Pairwise inter-residue correlation methods discover second-order features that can be useful in the prediction of structure and function and that can be built into classifiers more sensitive than first-order sequence classifiers and fold-recognizers (Bowie, Luethy and Eisenberg, 1991b: Bryant and Lawrence, 1993). To the extent that k-ary interactions are important, and to the extent that such interactions leave traces in sets of homologous sequences — and both of these issues require further study — the pairwise methods are deficient. One can try to infer k-ary correlations from sets of 2-ary correlations (Korber et al., 1993) (essentially by computing the transitive closure of the “CorrelatesWith” binary relation), but this heuristic can lead to trouble: high pairwise correlations among variables \(x, y, z\) do not in general imply, nor are they necessarily implied by, a high 3-ary correlation (as measured by Kullback divergence) of the three variables \(x, y, z\). For the purpose of RNA structural studies, pairwise correlations alone move one closer to tertiary structure prediction than is true for protein analysis, though the existence and importance of some key non-planar tertiary and quaternary interactions suggests a need for higher-order k-ary feature detectors in RNA work as well.

HMMs are now widely used in protein, RNA, and DNA sequence analysis, but their emphasis on sequentially local pairwise interactions limits their utility in structure discovery and prediction.

Explicitly higher-order latent-variable modelling methods, like that proposed by MacKay for protein sequence family analysis (MacKay, 1994), can in principle find all interesting higher-order correlations. when proper hierarchical Bayesian updating methodology is followed. The problem with such methods is the need to specify parameters and hyper-parameters for. and allocate space for. thousands of possible combinations of amino acids (symbols) and/or residues (observable variables) that may not even occur in the dataset.

### 4.7.2 General Data Mining

We developed the coincidence detection procedures primarily for protein sequence analysis and structure prediction: but its more significant impact, if any, will likely be on other data mining problems in other application domains. We can describe a class of problems on which the comparative advantages of the coincidence detection method are most apparent. Such problems are characterized by:
1. a large number of attributes (columns, in our representation):

2. the presumed existence of some number of cliques of highly mutually correlated attributes in the dataset. each member attribute of each such clique being uncorrelated with attributes outside its own clique:

3. lack of prior knowledge as to the precise number, width (k, as in k-ary correlation and kth-order feature), and location of such attribute cliques.

All other data-mining procedures of which we are aware either place prior limitations on the width k of discoverable k-tuples, or implement an exhaustive search, serial (Agrawal et al., 1996) or parallel (MacKay, 1994), over all possible k-tuples of attributes. To put it more simply, the coincidence detection procedure takes as much computation time and memory to find a 43-ary correlation as it takes to find a 2-ary correlation in the same very high dimensional dataset. Other methods, in contrast, either rule out the discovery of the 43rd-order feature or else require the allocation of orders of magnitude more time or space in order to find it.

We summarize some archetypal examples of these other data-mining procedures below.

### 4.7.2.1 Miller's Pattern-Collection Procedure for Gibbs Models

Miller's proposal (Miller, 1993) for Gibbs models of databases is based on the use of Gibbs potentials, and he proposes a hashing method for calculating these special terms. As outlined earlier in Section 4.2, each kth-order potential requires an estimation of a kth-order joint probability density as well as some number of lower-order (typically k\(^{-1}\)th-order) densities. The asymptotic time complexity of Miller's pattern-collection subroutine, the major component of the potential calculation, is, when interpreted in our terminology:

\[
\mathcal{M} \cdot \sum_{k=1}^{K} \binom{N_A}{k} 2^k \approx O(MN^K)
\]

where \(K = k_{\text{max}}\) is the highest order of features for which one will search and by which one will represent database objects. This exponential blow-up prevents one from searching for HOFs of any order k much higher than 4 or 5 in databases with hundreds of attributes.
4.7.2.2 The Agrawal et al. Method for Discovery of Association Rules

The "IBM method" of Agrawal et al. (Agrawal et al., 1996) was developed in perhaps the purest data mining context, the automatic extraction of knowledge-base rules from databases. The authors consider a database of $M$ transactions (objects, rows) and $N$ items (attributes, columns) and seek to extract rules of the form $a \Rightarrow b$. They therefore seek pairs of attributes $a, b$ such that "transactions that contain $a$ tend to contain $b"$, hence those pairs with high values for $p(b|a)$. "People who buy expensive CD players tend to buy single-malt Highland Scotch whisky." is just one example suggesting the potential commercial interest in such methods. (More generally, one can search for sets of attributes with high $p(b_1,b_2,\ldots,b_k|a_1,a_2,\ldots,a_l).$

A rule $a \Rightarrow b$ is said to have:

1. **confidence** $c$ if $c\%$ of transactions containing $a$ also contain $b$ (hence, roughly, if $\frac{p(a,b)}{p(a)} \geq \frac{c}{100}$);

2. **support** $s$ if $s\%$ of transactions contain $a$ and $b$ (hence, roughly, if $p(a,b) \geq \frac{s}{100}$).

The goals behind their method are different from our goals in developing the coincidence detection procedure. However, the different objectives are brought closer together if one focuses on the Agrawal method's discovery of **symmetric rules** (so that the search is for attribute pairs displaying high values for both $\frac{p(a,b)}{a}$ and $\frac{p(a,b)}{b}$), and if one reduces the emphasis on support (so that coincidences that are suspicious, even if occurring rarely, are sought).

The Agrawal method is shown to have $O(||S|| \cdot MN)$ time complexity, where $||S||$ is the sum of all values $Support(\alpha)$ for an exponentially large number of $k$-tuples $\alpha$ of attributes, of any size $1 \leq k \leq N$, that reach a particular stage of processing in the Agrawal procedure. Hence the method is $O(2^N)$ in the worst case. The authors performed a series of empirical tests on what they considered to be realistic datasets for their domain. The experiments showed that the running time of the procedure grew only linearly with the number $M$ of transactions, but they reported no results pertaining to the growth of running time as a function of the number of items, or attributes. In fact, they held the number of items constant at $N_A = 1000$, and their constructed datasets probably contained no correlated $k$-tuples of width $k > 10$. An analysis of their algorithm, which is based on an incremental build-up of $k$th-order cliques from $k-1$th-order cliques, makes clear that the method takes much more computation to find wide HOFs (large $k$) than narrower HOFs (lower $k$) of equivalent statistical significance, in stark contrast to the behaviour of our coincidence detection method.
4.7.2.3 The Paturi et al. Method for Identifying the Most Correlated Pair of Random Variables

In theoretical computer science paper (Paturi, Rajasekaran and Reif, 1995), a method is reported for the problem of finding the most highly correlated pair $X_i, X_j$ of variables from among a large set of $N$ random binary variables $X_1, X_2, \ldots, X_N$. The method is easily extended to finding the most correlated $k$-tuple of random binary variables, but at a significant increase in computational complexity, and only for $k \geq 2$ fixed a priori. The authors use a definition of correlation that has \( \text{Correlation}(X_i, X_j) = P[X_i = X_j] \) over some set of $M$ samples \( \{X_1^m, X_2^m, \ldots, X_N^m\}_{m=1,2,\ldots,M} \).

If we extend their definition to include anti-correlations \( P[X_i \neq X_j] \) and assume their use of the same vector encoding that we use for representing $|A|$-ary variables as binary variables, then their method’s objective approximates ours, except for this crucial distinction: They want to distinguish the highest correlation from the second highest (and all others), whereas we only want to find all high correlations. Much of the computational complexity, both time complexity and sample complexity, of their method can be incurred in trying to separate two or more nearly equally-correlated pairs (or $k$-tuples) of variables.

The two variants of the Paturi method are asymptotically quadratic and sub-quadratic in $N$, respectively, the faster procedure requiring more sampling\(^6\). When the method is extended to search for the biggest $k$-ary correlation, where correlation is now defined as $P[X_{i_1} = X_{i_2} = \ldots = X_{i_k}]$, the time complexity grows to approximately $O(k^2 N^k \log^3 N)$. Search for highly correlated attribute cliques of width $k$ much greater than 5 or 6 in very large datasets is once again ruled out.

4.8 Other Applications of the Coincidence Detection Method

The coincidence detection algorithm presented in this chapter has application far beyond its use in finding correlated residues in protein sequences. Even just within the molecular bioinformatics domain, its use can be extended to finding $k$-tuples of associated features from sequence, structure, function, and literature citations. At least one group has demonstrated the power of “mining” such associations, by using the Agrawal method to find protein structure-function links (Satou et al., 1997).

---

\(^6\)Details of the algorithms and their complexity analysis are beyond the scope of this thesis chapter, and the interested reader is referred to the original manuscript.
It is important to note that none of our method's strengths or effectiveness depend on any aspects particular to the protein science domain. Indeed, the algorithm's chief strength is in finding associations among any fields separated by any semantic or representational distance within a database record, a strength almost wasted on domains in which the key correlations occur between sequentially local fields or features.

The method can be applied immediately to the same kinds of datasets used to evaluate the Agrawal method, for example. The "market basket" problem, in which one analyzes a database of sales transactions (rows) in order to find sets of products (columns) that tend to be bought together, is amenable to solution by our method without any special preprocessing. The author of this thesis is also currently applying the coincidence detection method to small-molecule databases as part of a drug discovery project.

4.9 Discussion and Future Work

Modellers of protein sequence families (or for that matter, of other very large data sets, e.g., census data or text corpora) are thwarted in their attempts to compute very far into a fully higher-order probabilistic model by both the computational complexity of the task and by the lack of data needed to support statistically significant estimates of most of the higher-order terms. In fact, even the huge PIR (Barker et al., 1991) database is too small to allow for a generally satisfactory estimation of all second-order terms for a given family or superfamily (Stolorz, Lapedes and Xia, 1991b; Garnier, Osguthorpe and Robson, 1978a).

One reasonable and common solution to this problem is to compute only a subset of higher-order probabilities, and extract a limited selection of higher-order features for construction of a database model. That is our approach. We suggest that more efficient use can be made of limited computing resources by pre-selecting sets of higher-order features using the correlation-detection algorithms described in this paper, and building the most significant (statistically and in terms of known evolutionary or biophysical importance) into model-based classifiers and predictors based on existing statistical, rule-based, neural network, or grammar-based methods. The discovered HOFs can alternatively be used directly in the prediction or determination of protein structure, when fed into existing methods based on distance geometry or empirically-estimated patterns of cooperativity and folding.

We propose that after further analyses, empirical testing and fine-tuning, the correlation detection methodologies discussed herein, including our own as well as the more exhaustive
search techniques (Agrawal et al., 1996; Paturi, Rajasekaran and Reif, 1995), can be of some advantage in the analysis of protein and nucleic acid sequences and structures, and may provide greater advantage in other data-mining domains.

Our coincidence-detection procedure, based on two levels of Bernoulli-like sampling, a discrete coding trick, and simple Chernoff-Hoeffding probability deviation results, can be viewed as an initial filter for detection of "suspicious coincidences". Such coincidences play an important role in database modelling, machine learning, and perception and pattern-recognition. In database mining and modelling contexts, a procedure for discovery of these features might serve any of several major roles, including:

1. **Preprocessing of large, complex datasets**: Many of the best modelling methods, including Gibbs models, Hidden Markov Models and EM, MacKay's density networks, and related factorial learning methods from the neural network community, could be helped significantly in capturing higher-order interactions without exhaustive search or combinatorial explosion of parameter space if preceded by fast a preprocessing procedure that found plausibly correlated variables in the database.

2. **Visual exploration of large complex data sets**: If coupled to even a simple graphical display interface, a procedure such as ours permits a user to view quickly (with small number of r-samples) the most plausibly interesting higher-order features in high-dimensional data.

3. **Pre-conditioning and redundancy elimination**: Thus far, we have stressed the utility of finding inter-attribute correlations in order to use them in the building of models; but in many optimization, learning, and data-fitting applications, one requires that correlations between variables be found and eliminated, through any of a number of subspace methods like PCA (Press et al., 1988; Becker and Plumbley, 1996).

The coincidence-detection method works by trading off depth of analysis for width (the order $k$) of higher-order features discoverable. The simplest, most naive version of the procedure was shown to perform well in practice at finding highly-correlated attributes, with observed time and space complexity only linear in $M$ and $N$. Further analysis, on a wide variety of different datasets with very different underlying probability distributions should produce a richer collection of results, providing a deeper understanding of the relative merits of various procedures and assumptions in various applications.
Finally, we note that the coincidence-detection procedures are amenable to very fast parallel hardware implementations, on general or special-purpose configurations. The sampling can be split into parallel r-by-n blocks for processing, and the binary-addressing trick suggests one-step RAM address decoding in hardware. We are in the process of developing parallel designs.
Chapter 5

Discussion and Conclusions

It is useful to conclude this thesis by tying together some of the threads running through the earlier chapters.

The focus of the thesis is the problem (PSP) of predicting protein tertiary structure from amino acid sequence, because it is an interesting and important problem in its own right, because it can be expressed in any of several mathematical and computational forms, and because PSP and its subproblems bear resemblance to other important problems in the computational sciences and engineering.

PSP is, in its most general formulation, intractable. It is intractable in both the sense that biomedical scientists cannot yet reliably obtain the structures that they need from the vast sequence databases that are available, and in the formal computer science sense that the problem is provably at least as hard as some well-known intractable problems.

However, while it remains impossible to reliably predict thermodynamically optimal or "native" global conformations from sequence alone, it is nonetheless often possible to come reasonably close. First, one does not always need to predict a complete, detailed native structure. There are both general-purpose and domain-specific heuristics that lead reliably to useful partial predictions, and there is hope for more, because evolutionary and biophysical constraints on functional structures enable a divide-and-conquer approach as well as practical solutions to the component subproblems. Second, there is often much more information available for structure prediction than mere sequence. Even in the absence of x-ray diffraction or nuclear magnetic resonance data, one can exploit information from homologous sequences and phylogenetic information, from databases of stored structure and substructure motifs, and from any of a number of physical, chemical, and biological experiments.
Some of the sources of such exploitable information, as well as some of the ways to combine and exploit it, were discussed in Chapter Two. Several computational methods, representing two rather different kinds of unsupervised learning and motif discovery, were presented in Chapters Three and Four.

In Chapter Two, it was discussed how different sources of information about a protein could be used to augment or replace a detailed theoretical potential function for computationally "folding" a protein. A few points of entry for such empirical information were highlighted:

1. If local structure fragments (secondary structure) can be predicted above some hypothetical threshold level of accuracy, then methods exist for predicting how such structures will pack in a global conformation (Cariani and Goel. 1985: Chothia, Levitt and Richardson. 1977).

2. Empirical contact potentials can be derived from statistical analyses of amino-amino contact or proximity data and of amino acid "preferences" of particular physico-chemical attributes for local residue environments. These empirical potentials can be substituted for theoretical potentials, via an "Inverse Boltzmann’s Law" (Sippl. 1993).

3. Knowledge of particular inter-residue distances or interaction tendencies in a protein can be input to a distance geometry calculation (Havel, Kuntz and Crippen. 1983).

In light of these possible routes to global tertiary structure prediction, the computational methods of Chapters Three and Four can be understood:

- **Chapter Three**: Methods were presented for the simultaneous discovery of local sequence and local structure motifs. To the extent that such motifs are correlated, they are mutually predictive. Some initial success at finding novel secondary structure motifs that are more predictable, in two-class prediction, than the standard helix, sheet and coil, suggests that further research could lead to a secondary structure "language" that enables easier and more accurate tertiary structure prediction than is currently possible.

- **Chapter Four**: A combinatorial data mining method was introduced and analyzed. This "coincidence detection" algorithm can be used to find correlated k-tuples of residues in an aligned family of protein sequences. Such correlated cliques can sometimes give direct evidence of key structural features: more generally, the use of sequence classification and
structure prediction and recognition methods that eschew the simplistic residue independence assumption raises hopes for more general and powerful bioinformatics systems.

It is not uncommon to see claims that some novel representation or information source can improve protein structure prediction and recognition. How would one go about quantifying predictive advantages gained though some claimed improvement?

The obvious way to approach such an assessment would include the design or adaptation of a tertiary structure prediction system that can handle the novel information or representation as an input, and the exhaustive testing, with controls, of the prediction system on a large and diverse set of proteins. Such an undertaking is beyond the scope of this single Computer Science dissertation. One of the many hurdles faced in such a project is the need to find or design a suitable tertiary structure prediction algorithm – there is of course no consensus as to what are the best energy or pseudo-energy functions, best search methods, and so on.

Alternatively, one can make some restrictive simplifying assumptions about protein structure and conformation, and from such simplified models one can tractably estimate the information gain provided by particular pieces of structural knowledge. Sibbald (Sibbald, 1995), for example, reduces protein conformations to chains embedded in cubic lattices, and uses this discretization to count the number of possible conformations with and without a given constraint. Translating frequencies into probabilities and thence into entropies of distributions, he is able to make information-theoretic estimates of, for example, the advantage provided by knowing that residues 18 and 33 are in contact in a given protein.

Another approach is to use any of a number of metrics which have been shown to provide, or are widely believed to provide, reasonable approximations to the information that exhaustive tertiary structure prediction tests would provide. Several of these metrics have been used in this thesis. The Matthews correlation measure was used to estimate indirectly the sequence-to-structure predictability of the novel secondary structure classes discovered by our coupled neural network architecture. Variations of MDL cost assessed the ability of the joint and conditional latent class modelling approaches in that same Chapter Three investigation to find “intrinsic” class structure in the space of 13-residue structure fragments. As discussed at length in Chapter Three, however, “intrinsicness” is just one criterion – along with motif length, number of classes, and others – with potentially great impact on tertiary structure predictive accuracy. As for the association-detection method presented in Chapter Four, we refer to the work of others who have observed, for example, that the prediction of pairwise inter-residue distances from correlation
information can provide an improvement of 1.4 to 5.1 times over random contact predictions (O’Donoghue and Rost. 1995). In addition, we have begun and will continue to test our method against other methods, against hypothesized and known protein structures, and on features beyond mere residue positions (as well in other domains beyond bioinformatics).

5.1 Contributions to Computer Science

Finally, it is appropriate to restate the contributions to the field of Computer Science made by this thesis and the work it represents.

1. In Chapter Three we report new formulations and experiments in the search for computational learning methods that combine information from two or more sources or modalities. In our case the sources are protein primary and tertiary structures, from which we derive novel "secondary structure" classes. But the underlying mathematical and computational problems are not wholly different if one is concerned instead with visual and auditory modalities, or with visual inputs from multiple cameras.

2. The problem of finding all significant correlations among pairs or k-tuples of attributes in a database is ubiquitous in the computational sciences and in medical, industrial, and financial applications. We propose and evaluate a probabilistic algorithm in Chapter Four that has the interesting property of finding significant higher-order k-ary correlations, for all k such that \( 2 \leq k \leq N \) in an N-attribute database, for the same computational cost of finding just significant pairwise correlations. Moreover, k need not be fixed in advance in our procedure, in contrast with other known procedures. The procedure was designed for the task of finding conserved structural relationships in aligned protein sequences, but may have more useful application in other domains. It is, for example, immediately and apparently applicable to the "market basket" problem often cited as a standard in data mining, and we are currently employing it on small-molecule datasets in drug-discovery applications.

3. In addition to the two specific problems and proposed solutions, there are perhaps more general insights to be gained from our analysis of a complex, seemingly intractable combinatorial search problem and our decomposition of it into a set of smaller, simpler subtasks, some of which are amenable to heuristic solution based upon unsupervised machine learning and database probability density estimation.
Appendix A

Additional Results from Chapter Four Experiments: Family0 Data

In all three Appendices, we present tables that summarize additional sets of results from experiments with the Coincidence Detection method of Chapter Four. Presented in this first of three appendix sections are results from experiments using the Family0 dataset, the construction and contents of which were explained in Chapter Four. In each table, the following format is used. The rows of a table correspond to the csets, or coincident sets, discovered during the particular experiment. The csets are ranked from highest correlation (most surprising: lowest estimated $p(\text{observed}|\text{Independence})$) to lowest. “Lowest” here means the lowest of the top ten that are being presented, or lowest of all those with $p(\text{observed}|\text{Independence})$, etc., as indicated. The first column is the index, or rank, of the cset. The second column is the identity of the cset, in terms of the attributes comprising it. The third column displays the actual observed number of hits (a.k.a. matches, occurrences) of the cset, while the fourth column displays the expected number (that is, expected under the assumption of independent attributes). The fifth, final column presents the $p(\text{observed}|\text{Independence})$ value output by the algorithm, which is our measure of correlation or association among the $k$ attributes forming the cset. All values in the tables were output from the Perl program implementing the algorithm: no special post-processing was performed.
Table A.1: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 5$.

Family0 Dataset.
$T = 500$, $r = 5$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>2</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>3</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>4</td>
<td>L5</td>
<td>N6</td>
<td>134</td>
<td>23.568214</td>
</tr>
<tr>
<td>5</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>6</td>
<td>F5</td>
<td>K6</td>
<td>107</td>
<td>22.980741</td>
</tr>
<tr>
<td>7</td>
<td>I5</td>
<td>R6</td>
<td>102</td>
<td>21.600871</td>
</tr>
<tr>
<td>8</td>
<td>V5</td>
<td>D6</td>
<td>97</td>
<td>22.475824</td>
</tr>
<tr>
<td>9</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
</tbody>
</table>

Table A.2: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 10,000$ and $r = 5$.

Family0 Dataset.
$T = 10,000$, $r = 5$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>3</td>
<td>F5</td>
<td>K6</td>
<td>2706</td>
<td>459.614828</td>
</tr>
<tr>
<td>4</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>5</td>
<td>I5</td>
<td>R6</td>
<td>1973</td>
<td>432.017413</td>
</tr>
<tr>
<td>6</td>
<td>L5</td>
<td>N6</td>
<td>3313</td>
<td>471.364273</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>9</td>
<td>V5</td>
<td>D6</td>
<td>2504</td>
<td>449.516484</td>
</tr>
<tr>
<td>10</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
</tbody>
</table>
Table A.3: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 6$.

**Family0 Dataset.**

$T = 500, r = 6$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H1</td>
<td>R2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>2</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>3</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>4</td>
<td>F5</td>
<td>K6</td>
<td>240</td>
<td>17.560963</td>
</tr>
<tr>
<td>5</td>
<td>L5</td>
<td>N6</td>
<td>215</td>
<td>17.653535</td>
</tr>
<tr>
<td>6</td>
<td>I5</td>
<td>R6</td>
<td>197</td>
<td>17.380061</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>9</td>
<td>V5</td>
<td>D6</td>
<td>169</td>
<td>17.543280</td>
</tr>
</tbody>
</table>

Table A.4: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 10.000$ and $r = 6$.

**Family0 Dataset.**

$T = 10.000, r = 6$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>3</td>
<td>F5</td>
<td>K6</td>
<td>4092</td>
<td>351.219267</td>
</tr>
<tr>
<td>4</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>5</td>
<td>I5</td>
<td>R6</td>
<td>3057</td>
<td>347.601211</td>
</tr>
<tr>
<td>6</td>
<td>L5</td>
<td>N6</td>
<td>4894</td>
<td>353.070695</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>9</td>
<td>V5</td>
<td>D6</td>
<td>3857</td>
<td>350.865596</td>
</tr>
<tr>
<td>10</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>11</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
</tbody>
</table>
Table A.5: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 500$ and $r = 7$.

Family0 Dataset.

$T = 500, \ r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>3</td>
<td>L5</td>
<td>N6</td>
<td>392</td>
<td>12.855892</td>
</tr>
<tr>
<td>4</td>
<td>F5</td>
<td>K6</td>
<td>299</td>
<td>13.046625</td>
</tr>
<tr>
<td>5</td>
<td>V5</td>
<td>D6</td>
<td>279</td>
<td>13.312864</td>
</tr>
<tr>
<td>6</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>9</td>
<td>I5</td>
<td>R6</td>
<td>194</td>
<td>13.595552</td>
</tr>
</tbody>
</table>

Table A.6: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family0. These results were produced with parameter settings $T = 10,000$ and $r = 7$.

Family0 Dataset.

$T = 10,000, \ r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>3</td>
<td>F5</td>
<td>K6</td>
<td>5916</td>
<td>260.932501</td>
</tr>
<tr>
<td>4</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>5</td>
<td>I5</td>
<td>R6</td>
<td>4554</td>
<td>271.911047</td>
</tr>
<tr>
<td>6</td>
<td>L5</td>
<td>N6</td>
<td>6758</td>
<td>257.117849</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>9</td>
<td>V5</td>
<td>D6</td>
<td>5410</td>
<td>266.257282</td>
</tr>
<tr>
<td>10</td>
<td>H1</td>
<td>I2</td>
<td>L3</td>
<td>K4</td>
</tr>
</tbody>
</table>
Table A.7: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset FamilyO. These results were produced with parameter settings $T = 500$ and $r = 10$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>3</td>
<td>H1</td>
<td>J2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>4</td>
<td>V5</td>
<td>D6</td>
<td>558</td>
<td>5.233777</td>
</tr>
<tr>
<td>5</td>
<td>L5</td>
<td>N6</td>
<td>552</td>
<td>4.465840</td>
</tr>
<tr>
<td>6</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>F5</td>
<td>K6</td>
<td>511</td>
<td>4.812927</td>
</tr>
<tr>
<td>9</td>
<td>I5</td>
<td>R6</td>
<td>437</td>
<td>5.854572</td>
</tr>
</tbody>
</table>

Table A.8: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset FamilyO. These results were produced with parameter settings $T = 10,000$ and $r = 10$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>R2</td>
<td>N3</td>
<td>D4</td>
</tr>
<tr>
<td>2</td>
<td>C1</td>
<td>Q2</td>
<td>E3</td>
<td>G4</td>
</tr>
<tr>
<td>3</td>
<td>F5</td>
<td>K6</td>
<td>10409</td>
<td>96.258547</td>
</tr>
<tr>
<td>4</td>
<td>H1</td>
<td>J2</td>
<td>L3</td>
<td>K4</td>
</tr>
<tr>
<td>5</td>
<td>I5</td>
<td>R6</td>
<td>9120</td>
<td>117.091443</td>
</tr>
<tr>
<td>6</td>
<td>L5</td>
<td>N6</td>
<td>12335</td>
<td>89.331675</td>
</tr>
<tr>
<td>7</td>
<td>M1</td>
<td>F2</td>
<td>P3</td>
<td>S4</td>
</tr>
<tr>
<td>8</td>
<td>T1</td>
<td>W2</td>
<td>Y3</td>
<td>V4</td>
</tr>
<tr>
<td>9</td>
<td>V5</td>
<td>D6</td>
<td>9703</td>
<td>104.675539</td>
</tr>
</tbody>
</table>
Appendix B

Additional Results from Chapter Four Experiments: Family1 Data

Presented in this second of three appendix sections are results from experiments using the Family1 dataset, the construction and contents of which were explained in Chapter Four. The format of the tables is the same as for the previous appendix, as discussed in the introduction to that section.

Table B.1: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 5$.

Family1 Dataset.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I19</td>
<td>R20</td>
<td>186</td>
<td>24.56219</td>
</tr>
<tr>
<td>2</td>
<td>F19</td>
<td>K20</td>
<td>174</td>
<td>23.841858</td>
</tr>
<tr>
<td>3</td>
<td>V19</td>
<td>D20</td>
<td>161</td>
<td>24.864019</td>
</tr>
<tr>
<td>4</td>
<td>L19</td>
<td>N20</td>
<td>112</td>
<td>21.381376</td>
</tr>
<tr>
<td>5</td>
<td>E4</td>
<td>R5</td>
<td>79</td>
<td>25.378105</td>
</tr>
</tbody>
</table>
Table B.2: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 10,000$ and $r = 5$.

**Family1 Dataset.**

$T = 10,000. r = 5.$

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F_{19}</td>
<td>K_{20}$</td>
<td>3414</td>
<td>$476.837158$</td>
</tr>
<tr>
<td>2</td>
<td>$I_{19}</td>
<td>R_{20}$</td>
<td>4116</td>
<td>$490.924389$</td>
</tr>
<tr>
<td>3</td>
<td>$L_{19}</td>
<td>N_{20}$</td>
<td>2022</td>
<td>$427.627520$</td>
</tr>
<tr>
<td>4</td>
<td>$V_{19}</td>
<td>D_{20}$</td>
<td>4438</td>
<td>$497.280380$</td>
</tr>
<tr>
<td>5</td>
<td>$E_{4}</td>
<td>R_{5}$</td>
<td>1273</td>
<td>$507.562102$</td>
</tr>
<tr>
<td>6</td>
<td>$S_{8}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>352</td>
</tr>
<tr>
<td>7</td>
<td>$S_{7}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>262</td>
</tr>
<tr>
<td>8</td>
<td>$G_{9}</td>
<td>L_{19}</td>
<td>N_{20}$</td>
<td>255</td>
</tr>
<tr>
<td>9</td>
<td>$G_{8}</td>
<td>F_{19}</td>
<td>K_{20}$</td>
<td>229</td>
</tr>
<tr>
<td>10</td>
<td>$L_{3}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>210</td>
</tr>
<tr>
<td>11</td>
<td>$A_{11}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>201</td>
</tr>
<tr>
<td>12</td>
<td>$A_{8}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>195</td>
</tr>
<tr>
<td>13</td>
<td>$S_{12}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>194</td>
</tr>
<tr>
<td>14</td>
<td>$G_{7}</td>
<td>F_{19}</td>
<td>K_{20}$</td>
<td>185</td>
</tr>
<tr>
<td>15</td>
<td>$A_{9}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>177</td>
</tr>
<tr>
<td>16</td>
<td>$A_{8}</td>
<td>L_{19}</td>
<td>N_{20}$</td>
<td>169</td>
</tr>
<tr>
<td>17</td>
<td>$K_{6}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>171</td>
</tr>
<tr>
<td>18</td>
<td>$L_{1}</td>
<td>K_{6}$</td>
<td>647</td>
<td>$503.877563$</td>
</tr>
<tr>
<td>19</td>
<td>$G_{12}</td>
<td>F_{19}</td>
<td>K_{20}$</td>
<td>171</td>
</tr>
<tr>
<td>20</td>
<td>$K_{5}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>162</td>
</tr>
<tr>
<td>21</td>
<td>$I_{3}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>159</td>
</tr>
<tr>
<td>22</td>
<td>$V_{1}</td>
<td>V_{19}</td>
<td>D_{20}$</td>
<td>155</td>
</tr>
<tr>
<td>23</td>
<td>$L_{2}</td>
<td>I_{19}</td>
<td>R_{20}$</td>
<td>146</td>
</tr>
<tr>
<td>24</td>
<td>$E_{4}</td>
<td>K_{5}$</td>
<td>602</td>
<td>$481.200782$</td>
</tr>
</tbody>
</table>

Table B.3: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 6$.

**Family1 Dataset.**

$T = 500. r = 6.$

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$I_{19}</td>
<td>R_{20}$</td>
<td>384</td>
<td>$17.844120$</td>
</tr>
<tr>
<td>2</td>
<td>$V_{19}</td>
<td>D_{20}$</td>
<td>413</td>
<td>$17.806616$</td>
</tr>
<tr>
<td>3</td>
<td>$F_{19}</td>
<td>K_{20}$</td>
<td>285</td>
<td>$17.881393$</td>
</tr>
<tr>
<td>4</td>
<td>$L_{19}</td>
<td>N_{20}$</td>
<td>213</td>
<td>$17.447203$</td>
</tr>
<tr>
<td>5</td>
<td>$E_{4}</td>
<td>R_{5}$</td>
<td>74</td>
<td>$16.889637$</td>
</tr>
<tr>
<td>6</td>
<td>$L_{1}</td>
<td>K_{6}$</td>
<td>51</td>
<td>$17.244706$</td>
</tr>
</tbody>
</table>
Table B.4: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 10.000$ and $r = 6$.

Family1 Dataset.

$T = 10.000, r = 6.$

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F19</td>
<td>K20$</td>
<td>6089</td>
<td>357.627869</td>
</tr>
<tr>
<td>2</td>
<td>$I19</td>
<td>R20$</td>
<td>7017</td>
<td>356.882394</td>
</tr>
<tr>
<td>3</td>
<td>$L19</td>
<td>N20$</td>
<td>3925</td>
<td>348.944056</td>
</tr>
<tr>
<td>4</td>
<td>$V19</td>
<td>D20$</td>
<td>7890</td>
<td>356.132317</td>
</tr>
<tr>
<td>5</td>
<td>$E4</td>
<td>R5$</td>
<td>1665</td>
<td>337.792730</td>
</tr>
<tr>
<td>6</td>
<td>$K4</td>
<td>D5$</td>
<td>906</td>
<td>328.620645</td>
</tr>
<tr>
<td>7</td>
<td>$L1</td>
<td>K6$</td>
<td>829</td>
<td>344.894114</td>
</tr>
<tr>
<td>8</td>
<td>$S8</td>
<td>I19</td>
<td>R20$</td>
<td>465</td>
</tr>
<tr>
<td>9</td>
<td>$K4</td>
<td>E5$</td>
<td>740</td>
<td>354.137744</td>
</tr>
<tr>
<td>10</td>
<td>$G9</td>
<td>L19</td>
<td>N20$</td>
<td>320</td>
</tr>
<tr>
<td>11</td>
<td>$G8</td>
<td>F19</td>
<td>K20$</td>
<td>272</td>
</tr>
<tr>
<td>12</td>
<td>$E4</td>
<td>K5$</td>
<td>580</td>
<td>324.752784</td>
</tr>
<tr>
<td>13</td>
<td>$V2</td>
<td>G12$</td>
<td>609</td>
<td>357.291804</td>
</tr>
<tr>
<td>14</td>
<td>$S7</td>
<td>I19</td>
<td>R20$</td>
<td>256</td>
</tr>
<tr>
<td>15</td>
<td>$K6</td>
<td>I19</td>
<td>R20$</td>
<td>247</td>
</tr>
<tr>
<td>16</td>
<td>$A8</td>
<td>L19</td>
<td>N20$</td>
<td>244</td>
</tr>
<tr>
<td>17</td>
<td>$L3</td>
<td>V19</td>
<td>D20$</td>
<td>235</td>
</tr>
<tr>
<td>18</td>
<td>$S12</td>
<td>V19</td>
<td>D20$</td>
<td>237</td>
</tr>
<tr>
<td>19</td>
<td>$A9</td>
<td>V19</td>
<td>D20$</td>
<td>207</td>
</tr>
<tr>
<td>20</td>
<td>$A11</td>
<td>I19</td>
<td>R20$</td>
<td>200</td>
</tr>
<tr>
<td>21</td>
<td>$L3</td>
<td>A9$</td>
<td>548</td>
<td>357.554038</td>
</tr>
<tr>
<td>22</td>
<td>$G12</td>
<td>F19</td>
<td>K20$</td>
<td>205</td>
</tr>
<tr>
<td>23</td>
<td>$V1</td>
<td>V19</td>
<td>D20$</td>
<td>202</td>
</tr>
<tr>
<td>24</td>
<td>$G7</td>
<td>F19</td>
<td>K20$</td>
<td>198</td>
</tr>
<tr>
<td>25</td>
<td>$R4</td>
<td>E5$</td>
<td>493</td>
<td>320.233800</td>
</tr>
<tr>
<td>26</td>
<td>$K5</td>
<td>V19</td>
<td>D20$</td>
<td>171</td>
</tr>
<tr>
<td>27</td>
<td>$E5</td>
<td>F19</td>
<td>K20$</td>
<td>171</td>
</tr>
<tr>
<td>28</td>
<td>$A8</td>
<td>V19</td>
<td>D20$</td>
<td>166</td>
</tr>
<tr>
<td>29</td>
<td>$A7</td>
<td>V19</td>
<td>D20$</td>
<td>165</td>
</tr>
<tr>
<td>30</td>
<td>$L2</td>
<td>I19</td>
<td>R20$</td>
<td>156</td>
</tr>
<tr>
<td>31</td>
<td>$G8</td>
<td>V19</td>
<td>D20$</td>
<td>154</td>
</tr>
<tr>
<td>32</td>
<td>$I3</td>
<td>I19</td>
<td>R20$</td>
<td>155</td>
</tr>
</tbody>
</table>
Table B.5: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 7$.

Family1 Dataset. $T = 500, r = 7.$

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F_{19}</td>
<td>K_{20}$</td>
<td>486</td>
<td>13.038516</td>
</tr>
<tr>
<td>2</td>
<td>$I_{19}</td>
<td>R_{20}$</td>
<td>498</td>
<td>12.611629</td>
</tr>
<tr>
<td>3</td>
<td>$V_{19}</td>
<td>D_{20}$</td>
<td>628</td>
<td>12.398153</td>
</tr>
<tr>
<td>4</td>
<td>$L_{19}</td>
<td>N_{20}$</td>
<td>300</td>
<td>13.841448</td>
</tr>
<tr>
<td>5</td>
<td>$E_{4}</td>
<td>R_{5}$</td>
<td>101</td>
<td>10.928158</td>
</tr>
<tr>
<td>6</td>
<td>$K_{4}</td>
<td>E_{5}$</td>
<td>60</td>
<td>12.411347</td>
</tr>
</tbody>
</table>
Table B.6: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 10,000$ and $r = 7$.

**Family1 Dataset.**

$T = 10,000$, $r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F19</td>
<td>K20$</td>
<td>9504</td>
<td>260.770321</td>
</tr>
<tr>
<td>2</td>
<td>$I19</td>
<td>R20$</td>
<td>10464</td>
<td>252.235800</td>
</tr>
<tr>
<td>3</td>
<td>$L19</td>
<td>N20$</td>
<td>6096</td>
<td>276.828951</td>
</tr>
<tr>
<td>4</td>
<td>$V19</td>
<td>D20$</td>
<td>11796</td>
<td>247.963061</td>
</tr>
<tr>
<td>5</td>
<td>$E4</td>
<td>R5$</td>
<td>1887</td>
<td>218.563156</td>
</tr>
<tr>
<td>6</td>
<td>$K4</td>
<td>D5$</td>
<td>1036</td>
<td>241.612852</td>
</tr>
<tr>
<td>7</td>
<td>$L1</td>
<td>K6$</td>
<td>824</td>
<td>229.515537</td>
</tr>
<tr>
<td>8</td>
<td>$S8</td>
<td>I19</td>
<td>R20$</td>
<td>562</td>
</tr>
<tr>
<td>9</td>
<td>$K4</td>
<td>E5$</td>
<td>798</td>
<td>248.226949</td>
</tr>
<tr>
<td>10</td>
<td>$R4</td>
<td>E5$</td>
<td>694</td>
<td>247.775665</td>
</tr>
<tr>
<td>11</td>
<td>$E4</td>
<td>K5$</td>
<td>606</td>
<td>213.081127</td>
</tr>
<tr>
<td>12</td>
<td>$V2</td>
<td>G12$</td>
<td>641</td>
<td>258.441072</td>
</tr>
<tr>
<td>13</td>
<td>$G9</td>
<td>L19</td>
<td>N20$</td>
<td>388</td>
</tr>
<tr>
<td>14</td>
<td>$G7</td>
<td>F19</td>
<td>K20$</td>
<td>293</td>
</tr>
<tr>
<td>15</td>
<td>$L3</td>
<td>A9$</td>
<td>496</td>
<td>219.418994</td>
</tr>
<tr>
<td>16</td>
<td>$S7</td>
<td>I19</td>
<td>R20$</td>
<td>274</td>
</tr>
<tr>
<td>17</td>
<td>$L3</td>
<td>A12$</td>
<td>464</td>
<td>200.298479</td>
</tr>
<tr>
<td>18</td>
<td>$V1</td>
<td>V19</td>
<td>D20$</td>
<td>227</td>
</tr>
<tr>
<td>19</td>
<td>$G8</td>
<td>F19</td>
<td>K20$</td>
<td>221</td>
</tr>
<tr>
<td>20</td>
<td>$K6</td>
<td>I19</td>
<td>R20$</td>
<td>220</td>
</tr>
<tr>
<td>21</td>
<td>$A9</td>
<td>V19</td>
<td>D20$</td>
<td>214</td>
</tr>
<tr>
<td>22</td>
<td>$L3</td>
<td>V19</td>
<td>D20$</td>
<td>213</td>
</tr>
<tr>
<td>23</td>
<td>$G12</td>
<td>F19</td>
<td>K20$</td>
<td>215</td>
</tr>
<tr>
<td>24</td>
<td>$A8</td>
<td>L19</td>
<td>N20$</td>
<td>210</td>
</tr>
<tr>
<td>25</td>
<td>$I3</td>
<td>I19</td>
<td>R20$</td>
<td>205</td>
</tr>
<tr>
<td>26</td>
<td>$S12</td>
<td>V19</td>
<td>D20$</td>
<td>200</td>
</tr>
<tr>
<td>27</td>
<td>$E5</td>
<td>F19</td>
<td>K20$</td>
<td>201</td>
</tr>
<tr>
<td>28</td>
<td>$K5</td>
<td>V19</td>
<td>D20$</td>
<td>194</td>
</tr>
<tr>
<td>29</td>
<td>$A7</td>
<td>V19</td>
<td>D20$</td>
<td>190</td>
</tr>
<tr>
<td>30</td>
<td>$A11</td>
<td>I19</td>
<td>R20$</td>
<td>181</td>
</tr>
<tr>
<td>31</td>
<td>$D6</td>
<td>L19</td>
<td>N20$</td>
<td>182</td>
</tr>
</tbody>
</table>
Table B.7: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 500$ and $r = 10$.

Family1 Dataset. $T = 500, r = 10$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F19</td>
<td>K20</td>
<td>1014</td>
<td>4.547474</td>
</tr>
<tr>
<td>2</td>
<td>L19</td>
<td>N20</td>
<td>754</td>
<td>6.217420</td>
</tr>
<tr>
<td>4</td>
<td>V19</td>
<td>D20</td>
<td>1196</td>
<td>3.764830</td>
</tr>
<tr>
<td>5</td>
<td>E4</td>
<td>R5</td>
<td>55</td>
<td>2.663106</td>
</tr>
</tbody>
</table>

Table B.8: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the artificial dataset Family1. These results were produced with parameter settings $T = 10,000$ and $r = 10$.

Family1 Dataset. $T = 10,000, r = 10$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F19</td>
<td>K20</td>
<td>20725</td>
<td>90.949470</td>
</tr>
<tr>
<td>2</td>
<td>L19</td>
<td>N20</td>
<td>14377</td>
<td>124.348401</td>
</tr>
<tr>
<td>4</td>
<td>V19</td>
<td>D20</td>
<td>23809</td>
<td>75.296605</td>
</tr>
<tr>
<td>5</td>
<td>E4</td>
<td>R5</td>
<td>1331</td>
<td>53.262127</td>
</tr>
<tr>
<td>6</td>
<td>K4</td>
<td>E5</td>
<td>1122</td>
<td>86.388750</td>
</tr>
<tr>
<td>7</td>
<td>K4</td>
<td>E5</td>
<td>768</td>
<td>76.902536</td>
</tr>
<tr>
<td>8</td>
<td>S8</td>
<td>L19</td>
<td>R20</td>
<td>674</td>
</tr>
<tr>
<td>9</td>
<td>R4</td>
<td>E5</td>
<td>725</td>
<td>103.250934</td>
</tr>
<tr>
<td>10</td>
<td>L1</td>
<td>K6</td>
<td>597</td>
<td>60.848882</td>
</tr>
<tr>
<td>11</td>
<td>V2</td>
<td>G12</td>
<td>564</td>
<td>87.991063</td>
</tr>
<tr>
<td>12</td>
<td>E4</td>
<td>K5</td>
<td>412</td>
<td>54.148057</td>
</tr>
<tr>
<td>13</td>
<td>R17</td>
<td>N18</td>
<td>484</td>
<td>127.171707</td>
</tr>
<tr>
<td>14</td>
<td>D4</td>
<td>K5</td>
<td>457</td>
<td>101.871862</td>
</tr>
<tr>
<td>15</td>
<td>G9</td>
<td>L19</td>
<td>N20</td>
<td>292</td>
</tr>
<tr>
<td>16</td>
<td>I3</td>
<td>S8</td>
<td>386</td>
<td>105.405994</td>
</tr>
</tbody>
</table>
Appendix C

Additional Results from Chapter Four Experiments: HIV Data

Presented in this final of three appendix sections are results from experiments using the HIV V3 dataset, the use, importance and contents of which were explained in Chapter Four. The format of the tables is the same as for the previous appendix, as discussed in the introduction to that section.

After the cset tables are two tables of estimates for pairwise inter-residue mutual information values. The first table shows our own mutual information estimates from our own version of the HIV V3 data. The second table shows the values computed by the Los Alamos group from their somewhat different dataset. As described in Results section of Chapter Four, there are important consistencies between their results, our mutual information estimates, and the pairs and k-tuples of correlated residues implied by our csets results shown in the tables above.

In the first table, the highest ranking pairs of residues are shown in the following format: The first column has the rank. The second column shows the pair of residues (positions, columns in the original data matrix of aligned sequences). The third column displays the estimated mutual information and the last column displays the standard error as calculated in the bootstrap estimation of the mutual information (as described in Chapter Four and the cited references).

In the last table, the Los Alamos group estimates of mutual information for the most highly associated pairs of residues are shown. Only the ranks and pairs are given, without the mutual information estimates themselves.
Table C.1: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 10,000$ and $r = 5$.

HIV Dataset.
$T = 10,000$, $r = 5$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q17</td>
<td>D24</td>
<td>1012</td>
<td>632.553864</td>
</tr>
<tr>
<td>2</td>
<td>R17</td>
<td>T21</td>
<td>901</td>
<td>610.770465</td>
</tr>
<tr>
<td>3</td>
<td>R12</td>
<td>Q17</td>
<td>570</td>
<td>348.605833</td>
</tr>
<tr>
<td>4</td>
<td>L13</td>
<td>W19</td>
<td>Q24</td>
<td>195</td>
</tr>
<tr>
<td>5</td>
<td>N4</td>
<td>K9</td>
<td>A21</td>
<td>226</td>
</tr>
<tr>
<td>6</td>
<td>V11</td>
<td>R12</td>
<td>T18</td>
<td>159</td>
</tr>
<tr>
<td>7</td>
<td>R12</td>
<td>T18</td>
<td>454</td>
<td>318.517747</td>
</tr>
<tr>
<td>8</td>
<td>L13</td>
<td>K31</td>
<td>419</td>
<td>300.333903</td>
</tr>
</tbody>
</table>

Table C.2: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 10,000$ and $r = 6$.

HIV Dataset.
$T = 10,000$, $r = 6$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q17</td>
<td>D24</td>
<td>1177</td>
<td>385.853329</td>
</tr>
<tr>
<td>2</td>
<td>R17</td>
<td>T21</td>
<td>957</td>
<td>368.736702</td>
</tr>
<tr>
<td>3</td>
<td>H12</td>
<td>A18</td>
<td>1047</td>
<td>577.583832</td>
</tr>
<tr>
<td>4</td>
<td>S10</td>
<td>D24</td>
<td>859</td>
<td>424.457490</td>
</tr>
<tr>
<td>5</td>
<td>R12</td>
<td>Q17</td>
<td>656</td>
<td>224.743830</td>
</tr>
<tr>
<td>6</td>
<td>R12</td>
<td>T18</td>
<td>628</td>
<td>283.191527</td>
</tr>
<tr>
<td>7</td>
<td>R17</td>
<td>E24</td>
<td>563</td>
<td>234.477161</td>
</tr>
<tr>
<td>8</td>
<td>H12</td>
<td>R17</td>
<td>760</td>
<td>434.274580</td>
</tr>
<tr>
<td>9</td>
<td>A18</td>
<td>T21</td>
<td>560</td>
<td>315.973734</td>
</tr>
<tr>
<td>10</td>
<td>I11</td>
<td>R17</td>
<td>861</td>
<td>627.014684</td>
</tr>
<tr>
<td>11</td>
<td>L13</td>
<td>W19</td>
<td>Q24</td>
<td>230</td>
</tr>
<tr>
<td>12</td>
<td>A21</td>
<td>D24</td>
<td>619</td>
<td>405.487239</td>
</tr>
<tr>
<td>13</td>
<td>N4</td>
<td>K9</td>
<td>A21</td>
<td>237</td>
</tr>
<tr>
<td>14</td>
<td>V11</td>
<td>R12</td>
<td>T18</td>
<td>220</td>
</tr>
<tr>
<td>15</td>
<td>L13</td>
<td>K31</td>
<td>462</td>
<td>267.211446</td>
</tr>
<tr>
<td>16</td>
<td>G10</td>
<td>H12</td>
<td>324</td>
<td>157.554658</td>
</tr>
<tr>
<td>17</td>
<td>M13</td>
<td>W15</td>
<td>245</td>
<td>84.760597</td>
</tr>
<tr>
<td>18</td>
<td>Q17</td>
<td>K31</td>
<td>384</td>
<td>231.749746</td>
</tr>
<tr>
<td>19</td>
<td>H12</td>
<td>R17</td>
<td>A18</td>
<td>147</td>
</tr>
<tr>
<td>20</td>
<td>N4</td>
<td>K9</td>
<td>H33</td>
<td>309</td>
</tr>
</tbody>
</table>
Table C.3: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 10,000$ and $r = 7$.

HIV Dataset.
$T = 10,000, r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q17</td>
<td>D24</td>
<td>1312</td>
<td>228.829775</td>
</tr>
<tr>
<td>2</td>
<td>N4</td>
<td>K9</td>
<td>2023</td>
<td>996.505631</td>
</tr>
<tr>
<td>3</td>
<td>H12</td>
<td>A18</td>
<td>1175</td>
<td>328.263693</td>
</tr>
<tr>
<td>4</td>
<td>R17</td>
<td>T21</td>
<td>940</td>
<td>216.431391</td>
</tr>
<tr>
<td>5</td>
<td>Q31</td>
<td>H33</td>
<td>3198</td>
<td>2481.050915</td>
</tr>
<tr>
<td>6</td>
<td>R12</td>
<td>T18</td>
<td>879</td>
<td>244.789294</td>
</tr>
<tr>
<td>7</td>
<td>S10</td>
<td>D24</td>
<td>836</td>
<td>232.201517</td>
</tr>
<tr>
<td>8</td>
<td>R12</td>
<td>Q17</td>
<td>720</td>
<td>140.866087</td>
</tr>
<tr>
<td>9</td>
<td>I11</td>
<td>R17</td>
<td>808</td>
<td>360.719364</td>
</tr>
<tr>
<td>10</td>
<td>H12</td>
<td>R17</td>
<td>659</td>
<td>253.717115</td>
</tr>
<tr>
<td>11</td>
<td>R17</td>
<td>A18</td>
<td>720</td>
<td>361.819054</td>
</tr>
<tr>
<td>12</td>
<td>A21</td>
<td>D24</td>
<td>554</td>
<td>236.085429</td>
</tr>
<tr>
<td>13</td>
<td>R17</td>
<td>E24</td>
<td>452</td>
<td>138.843412</td>
</tr>
<tr>
<td>14</td>
<td>L13</td>
<td>K31</td>
<td>537</td>
<td>231.137972</td>
</tr>
<tr>
<td>15</td>
<td>L13</td>
<td>W19</td>
<td>Q24</td>
<td>292</td>
</tr>
<tr>
<td>16</td>
<td>A18</td>
<td>T21</td>
<td>442</td>
<td>165.231990</td>
</tr>
<tr>
<td>17</td>
<td>A18</td>
<td>Q31</td>
<td>H33</td>
<td>480</td>
</tr>
<tr>
<td>18</td>
<td>M13</td>
<td>W15</td>
<td>355</td>
<td>88.975694</td>
</tr>
<tr>
<td>19</td>
<td>N4</td>
<td>K9</td>
<td>H33</td>
<td>340</td>
</tr>
<tr>
<td>20</td>
<td>V11</td>
<td>R12</td>
<td>513</td>
<td>253.001684</td>
</tr>
</tbody>
</table>
Table C.4: The most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 10,000$ and $r = 10$.

**HIV Dataset.**
$T = 10,000$. $r = 10$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$Q31</td>
<td>H33$</td>
<td>3933</td>
<td>883.532458</td>
</tr>
<tr>
<td>2</td>
<td>$N4</td>
<td>K9$</td>
<td>2898</td>
<td>251.248235</td>
</tr>
<tr>
<td>3</td>
<td>$S10</td>
<td>F19$</td>
<td>2245</td>
<td>907.769718</td>
</tr>
<tr>
<td>4</td>
<td>$F19</td>
<td>G23$</td>
<td>2660</td>
<td>1588.173503</td>
</tr>
<tr>
<td>5</td>
<td>$R12</td>
<td>T18$</td>
<td>1155</td>
<td>142.229768</td>
</tr>
<tr>
<td>6</td>
<td>$K9</td>
<td>11$</td>
<td>1230</td>
<td>311.653160</td>
</tr>
<tr>
<td>7</td>
<td>$A18</td>
<td>H33$</td>
<td>1720</td>
<td>990.576490</td>
</tr>
<tr>
<td>8</td>
<td>$K9</td>
<td>H33$</td>
<td>1125</td>
<td>405.874883</td>
</tr>
<tr>
<td>9</td>
<td>$H12</td>
<td>A18$</td>
<td>732</td>
<td>54.213558</td>
</tr>
<tr>
<td>10</td>
<td>$S10</td>
<td>G23$</td>
<td>1492</td>
<td>856.152048</td>
</tr>
<tr>
<td>11</td>
<td>$N4</td>
<td>H33$</td>
<td>1257</td>
<td>689.784961</td>
</tr>
<tr>
<td>12</td>
<td>$A18</td>
<td>Q31$</td>
<td>1188</td>
<td>636.901303</td>
</tr>
<tr>
<td>13</td>
<td>$Q17</td>
<td>D24$</td>
<td>571</td>
<td>42.938312</td>
</tr>
<tr>
<td>14</td>
<td>$V11</td>
<td>R12$</td>
<td>670</td>
<td>143.659674</td>
</tr>
<tr>
<td>15</td>
<td>$I11</td>
<td>R17$</td>
<td>562</td>
<td>61.788305</td>
</tr>
<tr>
<td>16</td>
<td>$N4</td>
<td>I11$</td>
<td>992</td>
<td>498.586806</td>
</tr>
<tr>
<td>17</td>
<td>$R12</td>
<td>Q17$</td>
<td>484</td>
<td>31.204991</td>
</tr>
<tr>
<td>18</td>
<td>$K31</td>
<td>Y33$</td>
<td>578</td>
<td>130.131866</td>
</tr>
<tr>
<td>19</td>
<td>$R17</td>
<td>T21$</td>
<td>479</td>
<td>39.372545</td>
</tr>
<tr>
<td>20</td>
<td>$S10</td>
<td>D24$</td>
<td>451</td>
<td>34.199456</td>
</tr>
</tbody>
</table>
Table C.5: The thirty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 100,000$ and $r = 7$.

HIV Dataset.

$T = 100,000$, $r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$H12</td>
<td>A18$</td>
<td>11686</td>
<td>3282.636926</td>
</tr>
<tr>
<td>2</td>
<td>$N4</td>
<td>K9$</td>
<td>21853</td>
<td>9965.056308</td>
</tr>
<tr>
<td>3</td>
<td>$Q17</td>
<td>D24$</td>
<td>11585</td>
<td>2288.297747</td>
</tr>
<tr>
<td>4</td>
<td>$Q31</td>
<td>H33$</td>
<td>31715</td>
<td>24810.509148</td>
</tr>
<tr>
<td>5</td>
<td>$R17</td>
<td>T21$</td>
<td>9355</td>
<td>2164.313906</td>
</tr>
<tr>
<td>6</td>
<td>$R12</td>
<td>Q17$</td>
<td>7259</td>
<td>1408.660868</td>
</tr>
<tr>
<td>7</td>
<td>$R12</td>
<td>T18$</td>
<td>8380</td>
<td>2447.892936</td>
</tr>
<tr>
<td>8</td>
<td>$S10</td>
<td>D24$</td>
<td>7666</td>
<td>2322.015166</td>
</tr>
<tr>
<td>9</td>
<td>$I11</td>
<td>R17$</td>
<td>8336</td>
<td>3607.193645</td>
</tr>
<tr>
<td>10</td>
<td>$A21</td>
<td>D24$</td>
<td>6342</td>
<td>2360.854285</td>
</tr>
<tr>
<td>11</td>
<td>$H12</td>
<td>R17$</td>
<td>6363</td>
<td>3253.171446</td>
</tr>
<tr>
<td>12</td>
<td>$R17</td>
<td>A18$</td>
<td>7162</td>
<td>3618.190543</td>
</tr>
<tr>
<td>13</td>
<td>$R17</td>
<td>E24$</td>
<td>4451</td>
<td>1388.434119</td>
</tr>
<tr>
<td>14</td>
<td>$A18</td>
<td>T21$</td>
<td>4673</td>
<td>1652.319901</td>
</tr>
<tr>
<td>15</td>
<td>$V11</td>
<td>R12$</td>
<td>5486</td>
<td>2530.016841</td>
</tr>
<tr>
<td>16</td>
<td>$L13</td>
<td>K31$</td>
<td>5224</td>
<td>2311.379719</td>
</tr>
<tr>
<td>17</td>
<td>$N4</td>
<td>K9</td>
<td>H33$</td>
<td>3519</td>
</tr>
<tr>
<td>18</td>
<td>$A18</td>
<td>Q31</td>
<td>H33$</td>
<td>4665</td>
</tr>
<tr>
<td>19</td>
<td>$L13</td>
<td>W19</td>
<td>Q24$</td>
<td>2585</td>
</tr>
<tr>
<td>20</td>
<td>$R17</td>
<td>Q31$</td>
<td>5967</td>
<td>3574.032273</td>
</tr>
<tr>
<td>21</td>
<td>$M13</td>
<td>W15$</td>
<td>3204</td>
<td>889.756945</td>
</tr>
<tr>
<td>22</td>
<td>$V11</td>
<td>R12</td>
<td>T18$</td>
<td>2424</td>
</tr>
<tr>
<td>23</td>
<td>$N4</td>
<td>A21$</td>
<td>6209</td>
<td>4030.321314</td>
</tr>
<tr>
<td>24</td>
<td>$K31</td>
<td>Y33$</td>
<td>4878</td>
<td>2773.817984</td>
</tr>
<tr>
<td>25</td>
<td>$Q17</td>
<td>K31$</td>
<td>3440</td>
<td>1450.098718</td>
</tr>
<tr>
<td>26</td>
<td>$K9</td>
<td>A21$</td>
<td>5614</td>
<td>3692.671816</td>
</tr>
<tr>
<td>27</td>
<td>$F19</td>
<td>D24$</td>
<td>3998</td>
<td>2250.071839</td>
</tr>
<tr>
<td>28</td>
<td>$Q17</td>
<td>A21$</td>
<td>4151</td>
<td>2414.536189</td>
</tr>
<tr>
<td>29</td>
<td>$G10</td>
<td>H12$</td>
<td>2661</td>
<td>953.572593</td>
</tr>
<tr>
<td>30</td>
<td>$H12</td>
<td>E24$</td>
<td>3018</td>
<td>1458.576938</td>
</tr>
</tbody>
</table>
Table C.6: The first twenty-five of the fifty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset. These results were produced with parameter settings $T = 750.000$ and $r = 7$. Note the appearance, at this degree of sampling, of several statistically significant higher-order features with $k \geq 3$.

HIV Dataset.
$T = 750.000$, $r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A18</td>
<td>Q31</td>
<td>H33</td>
<td>36019</td>
</tr>
<tr>
<td>1</td>
<td>A18</td>
<td>T21</td>
<td>33816</td>
<td>12392.399254</td>
</tr>
<tr>
<td>2</td>
<td>A21</td>
<td>D24</td>
<td>45549</td>
<td>17706.407140</td>
</tr>
<tr>
<td>3</td>
<td>H12</td>
<td>A18</td>
<td>86025</td>
<td>24619.776947</td>
</tr>
<tr>
<td>4</td>
<td>H12</td>
<td>R17</td>
<td>48257</td>
<td>19028.783592</td>
</tr>
<tr>
<td>5</td>
<td>I11</td>
<td>R17</td>
<td>64548</td>
<td>27053.952336</td>
</tr>
<tr>
<td>6</td>
<td>L13</td>
<td>K31</td>
<td>39382</td>
<td>17335.347894</td>
</tr>
<tr>
<td>7</td>
<td>L13</td>
<td>W19</td>
<td>Q24</td>
<td>20184</td>
</tr>
<tr>
<td>8</td>
<td>M13</td>
<td>W15</td>
<td>23300</td>
<td>6673.177086</td>
</tr>
<tr>
<td>9</td>
<td>N4</td>
<td>K9</td>
<td>162152</td>
<td>74737.922307</td>
</tr>
<tr>
<td>10</td>
<td>N4</td>
<td>K9</td>
<td>H33</td>
<td>26376</td>
</tr>
<tr>
<td>11</td>
<td>Q17</td>
<td>D24</td>
<td>86891</td>
<td>17162.233105</td>
</tr>
<tr>
<td>12</td>
<td>Q31</td>
<td>H33</td>
<td>233190</td>
<td>186078.818611</td>
</tr>
<tr>
<td>13</td>
<td>R12</td>
<td>Q17</td>
<td>53740</td>
<td>10564.956512</td>
</tr>
<tr>
<td>14</td>
<td>R12</td>
<td>T18</td>
<td>62774</td>
<td>18359.197022</td>
</tr>
<tr>
<td>15</td>
<td>R17</td>
<td>A18</td>
<td>54366</td>
<td>27136.429076</td>
</tr>
<tr>
<td>16</td>
<td>R17</td>
<td>E24</td>
<td>33748</td>
<td>10413.255892</td>
</tr>
<tr>
<td>17</td>
<td>R17</td>
<td>Q31</td>
<td>15065</td>
<td>26805.242087</td>
</tr>
<tr>
<td>18</td>
<td>R17</td>
<td>T21</td>
<td>70301</td>
<td>16232.354294</td>
</tr>
<tr>
<td>19</td>
<td>S10</td>
<td>D24</td>
<td>57772</td>
<td>17415.113746</td>
</tr>
<tr>
<td>20</td>
<td>V11</td>
<td>R12</td>
<td>39546</td>
<td>18975.126308</td>
</tr>
<tr>
<td>21</td>
<td>V11</td>
<td>R12</td>
<td>T18</td>
<td>17628</td>
</tr>
<tr>
<td>22</td>
<td>K31</td>
<td>Y33</td>
<td>36346</td>
<td>20803.634880</td>
</tr>
<tr>
<td>23</td>
<td>N4</td>
<td>A21</td>
<td>45441</td>
<td>30227.409858</td>
</tr>
<tr>
<td>24</td>
<td>Q17</td>
<td>K31</td>
<td>25033</td>
<td>10875.740384</td>
</tr>
<tr>
<td>25</td>
<td>G10</td>
<td>H12</td>
<td>20779</td>
<td>7151.794446</td>
</tr>
</tbody>
</table>
Table C.7: Continuation of the fifty most likely correlated attributes, as estimated by the coincidence detection procedure, for the HIV dataset: csets ranked 26 through 50. These results were produced with parameter settings $T = 750,000$ and $r = 7$. Note the appearance, at this degree of sampling, of several statistically significant higher-order features with $k \geq 3$.

HIV Dataset.
$T = 750,000$, $r = 7$.

<table>
<thead>
<tr>
<th>Rank</th>
<th>CSET</th>
<th>Observed</th>
<th>Expected</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>K9</td>
<td>A21</td>
<td>40098</td>
<td>27695.038620</td>
</tr>
<tr>
<td>27</td>
<td>F19</td>
<td>D24</td>
<td>29121</td>
<td>16875.538795</td>
</tr>
<tr>
<td>28</td>
<td>Q17</td>
<td>A21</td>
<td>29621</td>
<td>18109.021417</td>
</tr>
<tr>
<td>29</td>
<td>H12</td>
<td>E24</td>
<td>22348</td>
<td>10939.327036</td>
</tr>
<tr>
<td>30</td>
<td>N4</td>
<td>K9</td>
<td>I11</td>
<td>15175</td>
</tr>
<tr>
<td>31</td>
<td>S4</td>
<td>T9</td>
<td>T12</td>
<td>V18</td>
</tr>
<tr>
<td>32</td>
<td>N4</td>
<td>K9</td>
<td>A21</td>
<td>11233</td>
</tr>
<tr>
<td>33</td>
<td>N4</td>
<td>Q31</td>
<td>H33</td>
<td>21868</td>
</tr>
<tr>
<td>34</td>
<td>F19</td>
<td>A21</td>
<td>44400</td>
<td>34516.144368</td>
</tr>
<tr>
<td>35</td>
<td>K9</td>
<td>Q31</td>
<td>H33</td>
<td>16593</td>
</tr>
<tr>
<td>36</td>
<td>W19</td>
<td>Q24</td>
<td>16738</td>
<td>7234.038664</td>
</tr>
<tr>
<td>37</td>
<td>E1</td>
<td>N12</td>
<td>10844</td>
<td>1492.835945</td>
</tr>
<tr>
<td>38</td>
<td>K9</td>
<td>E24</td>
<td>13847</td>
<td>4587.312260</td>
</tr>
<tr>
<td>39</td>
<td>K9</td>
<td>R17</td>
<td>33735</td>
<td>24568.179150</td>
</tr>
<tr>
<td>40</td>
<td>T12</td>
<td>V18</td>
<td>23076</td>
<td>14893.617567</td>
</tr>
<tr>
<td>41</td>
<td>R12</td>
<td>A21</td>
<td>15497</td>
<td>7516.155896</td>
</tr>
<tr>
<td>42</td>
<td>N4</td>
<td>K9</td>
<td>Q31</td>
<td>H33</td>
</tr>
<tr>
<td>43</td>
<td>N4</td>
<td>K9</td>
<td>A18</td>
<td>11655</td>
</tr>
<tr>
<td>44</td>
<td>S4</td>
<td>T9</td>
<td>T12</td>
<td>V18</td>
</tr>
<tr>
<td>45</td>
<td>R12</td>
<td>Q17</td>
<td>T18</td>
<td>7452</td>
</tr>
<tr>
<td>46</td>
<td>V11</td>
<td>Q17</td>
<td>14350</td>
<td>7329.962834</td>
</tr>
<tr>
<td>47</td>
<td>H12</td>
<td>T21</td>
<td>23263</td>
<td>16324.923094</td>
</tr>
<tr>
<td>48</td>
<td>Q17</td>
<td>Y33</td>
<td>17288</td>
<td>10374.788061</td>
</tr>
<tr>
<td>49</td>
<td>L13</td>
<td>W19</td>
<td>15536</td>
<td>8921.243955</td>
</tr>
<tr>
<td>50</td>
<td>S17</td>
<td>H28</td>
<td>6529</td>
<td>138.997153</td>
</tr>
</tbody>
</table>
Table C.8: The top thirty-five pairwise inter-column mutual information values for the HIV-V3 dataset, as estimated by our methodology as described in the main text.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Pair i, j</th>
<th>$MI(c_i, c_j)$</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>18</td>
<td>0.340449</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
<td>0.337943</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>21</td>
<td>0.319481</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>24</td>
<td>0.315202</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>24</td>
<td>0.314393</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>24</td>
<td>0.313992</td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>24</td>
<td>0.305609</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>24</td>
<td>0.297498</td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>26</td>
<td>0.290044</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>11</td>
<td>0.289911</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>23</td>
<td>0.285019</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>21</td>
<td>0.284936</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>21</td>
<td>0.278151</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>11</td>
<td>0.277189</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>21</td>
<td>0.273137</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>24</td>
<td>0.262226</td>
</tr>
<tr>
<td>17</td>
<td>21</td>
<td>24</td>
<td>0.260366</td>
</tr>
<tr>
<td>18</td>
<td>11</td>
<td>23</td>
<td>0.260337</td>
</tr>
<tr>
<td>19</td>
<td>11</td>
<td>19</td>
<td>0.249877</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>24</td>
<td>0.248938</td>
</tr>
<tr>
<td>21</td>
<td>19</td>
<td>23</td>
<td>0.242185</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>26</td>
<td>0.239395</td>
</tr>
<tr>
<td>23</td>
<td>9</td>
<td>19</td>
<td>0.238318</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>23</td>
<td>0.23359</td>
</tr>
<tr>
<td>25</td>
<td>24</td>
<td>25</td>
<td>0.222109</td>
</tr>
<tr>
<td>26</td>
<td>6</td>
<td>26</td>
<td>0.220371</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>26</td>
<td>0.220213</td>
</tr>
<tr>
<td>28</td>
<td>6</td>
<td>24</td>
<td>0.218815</td>
</tr>
<tr>
<td>29</td>
<td>9</td>
<td>12</td>
<td>0.214844</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>24</td>
<td>0.213921</td>
</tr>
<tr>
<td>31</td>
<td>10</td>
<td>12</td>
<td>0.2133</td>
</tr>
<tr>
<td>32</td>
<td>9</td>
<td>18</td>
<td>0.21078</td>
</tr>
<tr>
<td>33</td>
<td>11</td>
<td>21</td>
<td>0.210155</td>
</tr>
<tr>
<td>34</td>
<td>11</td>
<td>12</td>
<td>0.209421</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>19</td>
<td>0.20911</td>
</tr>
</tbody>
</table>
Table C.9: The top seven pairwise inter-column mutual information values for the HIV-V3 dataset, as estimated by the Los Alamos group.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Pair $i, j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>
References


Rohmel, J., Streitberg, B., and Tismer, C. A permutation approach to configural frequency analysis (cfa) and the iterated hypergeometric distribution. pages 355–378.


