Algorithms for the Identification of Functional Sites in Proteins

John Anthony Capra

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Mona Singh

June 2009

Abstract

Proteins play an essential role in nearly every process carried out by the cell. In accomplishing this incredibly diverse array of functions, proteins interact with one another and other molecules in their environments. The interactions of proteins with other molecules are mediated by specific amino acids. For a given protein, the identification of the residues that participate in its interactions can be a crucial step in understanding its function. Knowledge of these so-called functional sites can guide further experimental analysis of the protein and aid drug design and development. The large number of protein sequences and structural models that have become available over the past 10 years present an exceptional opportunity to use the methods of computer science and statistics to identify protein functional sites, and thereby further biological understanding.

This dissertation investigates the computational prediction of functional sites from protein sequence and structure data. First, we consider the estimation of evolutionary sequence conservation from a multiple sequence alignment of homologous proteins—a common first step in the identification of functionally important sites. We introduce a fast, information theoretic algorithm for scoring conservation and demonstrate that it provides state-of-the-art performance in predicting catalytic sites, ligand binding sites, and protein-protein interface residues. Second, we examine the identification of a class of functional residues that cannot be identified by considering sequence conservation alone: those that determine functional substrate specificity within homologous protein families. We combine sequence information with structural models to build the first large dataset of these specificity determining positions (SDPs). This dataset enabled the first large-scale analysis of sequence-based SDP prediction methods. We demonstrate that *GroupSim*, a new method we developed, outperforms existing approaches. Finally, we focus on the prediction of ligand binding sites when both evolutionary sequence information and structural models are available. We introduce *ConCavity*, a new algorithm which directly integrates sequence conservation information into structure-based surface pocket identification. This algorithm provides significant improvement over earlier methods and establishes the complementarity of sequence and structural evidence in ligand binding site prediction. Overall, our work significantly improves our ability to identify functional sites from protein sequences and structures.

Acknowledgements

My advisor, Mona Singh, provided invaluable help throughout my graduate career; her constant support has been essential to my development as a researcher. Without her patience and steadfast guidance, this work would not have been possible.

I am grateful to Tom Funkhouser, Olga Troyanskaya, Chris Floudas, and Fred Hughson for taking the time to serve on my committee.

Eric Banks, Jesse Farnham, Peng Jiang, Zia Khan, Elena Nabieva, Alex Ochoa, Anton Persikov, Jimin Song, Tao Yue, and Elena Zaslavsky have all provided valuable feedback on my work over the past several years.

I am also indebted to Rob Williams for getting me started in science and respecting me when I probably didn't deserve it.

My family's unwavering love and support are the foundation for all I have accomplished. I cannot overstate their importance to me.

I have been ludicrously fortunate to spend time with some exceptional people over the last five years. Forrester, Harlan, Ted, and Janek made graduate school more fun than I could have hoped. Adam and Justin consistently provided crucial escapes from Princeton and much-needed perspective. Alex and Michael were comforting reminders of my roots. Sage enriched my life in ways that I will never fully appreciate. I consider them all part of my family.

Finally, I doubt that I would've made it through without JD, Ajihei, or the towpath.

Contents

Chapter 1

Introduction

1.1 Motivation

Protein molecules are ubiquitous in the cell. They come in a dazzling variety of shapes and sizes and perform thousands of different functions crucial for life. Enzymes catalyze chemical reactions that drive essential processes related to cell growth, maintenance, and death. Other proteins transport materials within cells and between cells and their environments. Proteins like tubulin and collagen help determine the shape and structure of cell types. Transcription factor proteins control the transcription of DNA to RNA, and thus play a central role in the cell's ability to process hereditary information. These examples represent only a small fraction of the wide range of functions proteins perform. Incredibly, this astonishing diversity of protein function and form arises out of the combination of a simple set of building blocks: the 20 amino acids.

Proteins accomplish nearly all of their functions by interacting with other molecules. The nature of and participants in these interactions are as varied as the functions in which they are involved. Many proteins combine with other proteins to form stable functional complexes, while other proteins interact with each other in brief transient contacts. A significant number of proteins bind small molecules found in their environments. Still others, like the transcription factors, bind very specific positions on DNA and RNA molecules.

In each of these examples, a subset of the protein's amino acids participate in the interaction. A change of the identity of the amino acid residue found in one of these positions can have a large effect on the protein's function. The resulting changes in structure and chemistry might neutralize the protein's ability to bind the substrate or potentially cause the protein to bind a different molecule

more favorably. In extreme cases, the mutation could cause the protein to not fold properly at all. In contrast, some other amino acid positions are tolerant of such changes in residue identity; a mutation at one of these positions may have little effect on the protein's function.

Our goal is the analysis and prediction of these "functionally important" sites—positions which directly participate in a protein's functions by interacting with other molecules. Identification of these sites is an important step towards fully understanding the molecular mechanisms by which proteins accomplish their functions. In addition to providing direct evidence about the mechanism of protein function, knowledge of functionally important sites can improve the transfer of molecular function annotations between similar proteins by focusing comparisons on the relevant regions of the sequence or structure. Accurate information about functionally important positions can provide a starting point and framework for carrying out experimental analysis, e.g., with targeted mutations. Models of functionally important positions are also useful to drug design and development; knowledge of the relevant binding sites can guide the design of inhibitors and antagonists.

1.2 Data

Specific amino acid residues can play a wide range of roles in a protein. In other words, there are many different types of functionally important residues. Some residues will be necessary to maintain the proper overall fold of the protein. Others will participate directly in the protein's interactions with other molecules and its environment. Often an amino acid will have multiple roles. Residues involved in different functions have different properties. For example, catalytic sites—residues in enzyme active sites that directly participate in substrate binding and the catalytic mechanism of the enzyme—exhibit an amino acid distribution skewed towards charged residues and are usually found in concave pockets on the protein surface.

There are several sources of data which can be used to distinguish functionally important sites by defining properties in which sets of functional residues may differ from those that are not involved in the function. The most basic and widely available is the primary amino acid sequence of the protein. With the current state of sequencing technology, it is possible to obtain the sequence of nearly any protein of interest. The primary sequence itself contains relevant information about the function of individual residues that can be obtained by analyzing patterns and motifs in the sequence. For example, it is possible to identify the signature of certain structural motifs and to predict the secondary structure with reasonably high accuracy.

Once the primary sequence of a protein is known, it is informative to search for other proteins

with similar sequences. With nearly 1000 genomes completely sequenced and several thousand more in progress [1, 2], it is very common to find a number of similar proteins. If such proteins are found, it is likely that they evolved from a recent common ancestor. As a result, the similarities and differences between their sequences contain a font of information about the relative mutability of each position. Because of their importance to maintaining the functions of the protein, the sites we seek are often not tolerant to mutation. In a sense, nature tests how sensitive each amino acid position in a protein is to mutation as sequences evolve away from a common ancestor. Comparison of homologous protein sequences in a multiple sequence alignment, which attempts to align the corresponding positions in each protein, provides a window into this "natural experiment". Amino acid positions in these alignments which have accumulated a number of mutations since the last common ancestor are unlikely to be functionally important because these changes have not had a significant effect on the fitness of the organism. On the other hand, columns that are very conserved, i.e., do not exhibit a range of amino acids, are likely to be important, because few mutations have been accepted at these positions. This evolutionary conservation analysis assumes an accurate alignment, but is a very common and powerful approach to investigating the function of proteins and their residues. Each project described in this dissertation makes use of estimates of sequence conservation.

The 3D conformation of a protein is crucial to its function. 3D models of protein structure contain much information about the spatial relationships between residues that is not apparent when considering only the primary protein sequence; parts of the protein that are distant in sequence may fold up to be nearby in space. Thus knowledge of the 3D structure of a protein provides a wealth of information about the functional role of residues. Techniques such as nuclear magnetic resonance (NMR) spectroscopy and X-ray crystallography allow models of the 3D structure of many proteins to be determined. There are currently over 50,000 structures deposited in the PDB [3], and this number is increasing rapidly as a result of "structural genomics" projects such as the Protein Structure Initiative [4]. These models of protein 3D structure provide many hints about the function of residues within proteins. Residues that are important for maintaining the structural stability and proper fold of the protein are often very hydrophobic and found buried in the core of the structure. Residues involved in interactions with other proteins are usually found in large, relatively flat surface patches. Positions that bind small molecules are found in concave pockets on the protein surface. The arrangement and organization of secondary structure motifs also provides relevant information. For example, sites that bind ligands are much more likely to be found in the less structured loop regions between stable secondary structure motifs (alpha helices and beta sheets). These data and

observations can be used to improve the prediction of functional sites in general and the assignment of particular functions to specific residues. The projects described in Chapters 3 and 4 combine analysis of protein 3D structures with sequence data.

1.3 Contributions

This dissertation considers the computational prediction of a range of functional sites from protein sequence and structure data. Though we focus on several specific methods and types of functional sites, two general, high-level questions motivate the work: What can a given type of information about a position, e.g., evolutionary sequence conservation or a structural feature, tell us about its functional role in the protein? And how can this feature be used to improve the prediction of such residues in realistic settings? We present the results of three projects related to these questions in three self-contained chapters. The first two are joint work with Mona Singh and were previously published in *Bioinformatics* [5, 6]. The third is joint work with Mona Singh and Tom Funkhouser and is in submission. We now highlight our main contributions.

First, we introduce a new method for the estimation of evolutionary sequence conservation from a multiple sequence alignment of homologous proteins based on an information-theoretic measure called the Jensen-Shannon divergence (JSD). We demonstrate that our *JSD* method provides state-of-the-art performance in three common real-world prediction tasks, and that it does so in a small fraction of the time required by the previous best method. Conservation analysis is one of the most common and powerful approaches to identifying functionally important sites from sequence data, so our method represents a significant and widely-applicable advance. In spite of the prevalence of conservation analysis, there are few agreed upon best practices for its use. As a result, tens of conservation measures have been introduced across a range of prediction settings over the past 20 years. To address this problem and evaluate *JSD*, we developed a comprehensive evaluation framework for conservation methods that is grounded in common real-world prediction tasks—the identification of catalytic sites, ligand binding sites, and protein protein interaction interfaces (PPI). This framework enables us to make meaningful statements about the performance of different methods in different settings. For example, though many methods perform quite well in ligand binding site and catalytic site prediction, they all have more difficulty identifying protein-protein interaction interfaces. Due to its running time and performance, our *JSD* algorithm is the method of choice in each setting. Its speed enables fast genome-scale analysis and the ability to modify alignments on the fly.

Next, we turn our attention to identifying residues that determine protein substrate specificity. These so-called specificity determining positions (SDPs) are those in which a properly chosen amino acid substitution can change the preferred ligand or interaction partner of the protein. This type of functional site provides a fascinating challenge; they can rarely be predicted by traditional conservation analysis alone. The differences between proteins within the same protein family that bind similar ligands are often so small that current clustering and alignment methods that consider sequence information alone cannot distinguish them. The prediction and analysis of SDPs has also been significantly hindered by a dramatic lack of known SDPs. Obtaining reliable experimental information about the effect of amino acid substitutions on substrate binding is a painstaking process that requires extensive site-directed mutagenesis followed by binding assays. Our first contribution in this area is the development of a bioinformatics approach that combines information from sequence, structure, and experimental data to automate an analysis often undertaken by hand to identify likely SDPs in enzymes. The properties of the resulting dataset are in good agreement with those of known SDPs, and our method identifies several previously known SDPs. This large dataset of over 400 enzyme SDPs allows us to characterize their physicochemical and evolutionary properties; SDPs show significant differences from both catalytic sites and the overall residue background in this analysis. The dataset, created by integrating sequence, structure, and experimental data, also enables the evaluation of SDP prediction methods based on sequence information alone (the most common situation faced by researchers). We find that *GroupSim*, a fast, baseline method we developed, outperforms a representative set of previous approaches. By releasing our method and the new dataset, we have not only improved SDP prediction, but provided a foundation upon which future improvement can be built.

Finally, we return to the prediction of small molecule ligand binding sites, but we now consider the situation in which both evolutionary and structural evidence are available. To identify ligand binding pockets and residues, we introduce *ConCavity*, a new approach that directly integrates evolutionary sequence conservation data into structure-based pocket prediction. This is one of the first algorithms to directly integrate these two common approaches. In comprehensive comparisons of both sequence and structure based methods, we demonstrate that *ConCavity* vastly outperforms both types of previous method. In addition, we find that the best performing methods based on structural information alone outperform the best of those that only consider conservation data. Next, by analyzing the predictions of a representative set of methods, we explore the relationship between evolutionary sequence conservation, structure, and function. Overall, sequence and structure based algorithms predict many different positions, but these predictions provide largely complementary

information about ligand binding. For example, considering sequence conservation identifies many positions throughout the protein that are evolutionarily conserved for a variety of reasons not related to ligand binding. Similarly, structural pocket prediction methods find many surface pockets that do not bind ligands. Combining these two features in *ConCavity* yields significant improvement over either alone, and significantly extends the state-of-the-art in predicting ligand binding sites.

1.4 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce our new evolutionary sequence conservation estimation method, and demonstrate that it provides excellent performance in several common settings. In the third chapter, we describe the creation of the first large dataset of specificity determining positions (SDPs) and highlight several resulting insights about SDP properties and prediction. Chapter 4 gives the details of *ConCavity*, a new approach for predicting ligand binding sites that combines sequence conservation with structural features. In the final chapter, we summarize our conclusions and discuss some of the implications of these projects. We close by highlighting some potential avenues for further work.

Chapter 2

Predicting Functionally Important Residues from Sequence Conservation

2.1 Introduction

One of the most important and widely studied problems in protein sequence analysis is identifying which residues in a protein are responsible for its function. Knowledge of a protein's functionally important sites has immediate relevance for predicting function, guiding experimental analysis, analyzing molecular mechanisms, and understanding protein interactions.

Many computational methods have been developed to predict functionally important residues given a protein sequence. In this chapter, we focus on one of the most common approaches: the analysis of a multiple sequence alignment (MSA) of the protein and homologous sequences in order to find columns that are preferentially conserved. These sites are presumed to be functionally or structurally important because they have accepted fewer mutations relative to the rest of the alignment.

Conservation analysis has proven to be a powerful indicator of functional importance and has been used to detect residues involved in ligand binding [7, 8], in protein-protein interaction interfaces [9–11], in maintaining structure [12–14], and in determining protein functional specificity [15–17]. Conservation analysis has also been used in conjunction with structural information in many of these

applications (e.g., [18, 19]).

Computational methods for identifying functional residues that do not use conservation exist, but they typically require structural information and are usually employed in the unusual case where there is an absence or paucity of sequence homologs. Such structural approaches (review, [20]) work by either identifying local shared structural patterns (e.g., [21–23]) or by identifying residues in the protein structure with unusual electrostatic and ionization properties(e.g., [24, 25]). Many recent methods have used conservation along with other predictors of functional importance (e.g., solvent accessibility, secondary structure, catalytic propensities of amino acids, etc.) in a statistical learning framework [26–28]. It has been found that conservation is the single most powerful attribute in predicting functional importance in these settings [29].

While analysis of conservation is a very common approach with an intuitive basis [30], there is no universally agreed upon technique. Here, we introduce and evaluate a new information-theoretic measure for estimating sequence conservation that is motivated by the notion that conserved positions are under significant evolutionary pressure, and that positions under pressure are expected to have amino acid distributions very different from those of columns under no pressure [31]. We quantify this difference using the Jensen-Shannon divergence and an appropriate background "no pressure" distribution. We also give a window-based extension of our algorithm that incorporates the estimated conservation of sequentially adjacent residues into the score for each column; this window approach can be applied to any conservation scoring method that gives columnwise scores.

To compare the Jensen-Shannon divergence conservation measure to previously proposed methods, we create three data sets that correspond to different types of functional residues—catalytic residues, residues close to ligands, and residues in protein-protein interfaces—and give the first large-scale evaluation of several popular conservation measures in identifying functional sites.

We consider six previously introduced methods for estimating the conservation of a column within an MSA. The first and most commonly used method estimates conservation by calculating the Shannon entropy of the amino acid distribution of each column [32]. The second attempts to take amino acid similarity into account by partitioning the amino acids into stereochemically similar groups and then calculating the Shannon entropy in terms of this partition [33, 34]. The third incorporates the similarities between amino acids by adapting the von Neumann entropy to operate on an substitution matrix [9]. The fourth calculates the relative entropy [35] between a column distribution and a background distribution [31]; it is similar to our measure in that it attempts to identify sites that have amino acid distributions very different from those of columns under no evolutionary pressure. The fifth takes all pairs of amino acids in a column and sums their

pairwise similarity according to a similarity matrix [12]. The sixth, Rate4Site, is a sophisticated, computationally intensive approach that builds a phylogenetic tree for a family of protein sequences and infers the rate of evolution at each site [36].

We evaluate how well these seven conservation measures perform in identifying functional sites using ROC curves and by analyzing how often functional sites are within the top ranking sites. Our main findings are: (1) Jensen-Shannon divergence and Rate4Site perform similarly, and are not significantly outperformed by any other method on any data set. However, Jensen-Shannon divergence is several orders of magnitude faster, suggesting its use in many applications, such as genome-scale analyses. (2) The performance of Jensen-Shannon divergence improves when using our approach for incorporating the conservation of neighboring positions in the protein sequence. Incorporating the signal from neighboring residues also improves the performance of the other six methods. While considering the conservation of positions neighboring in 3D has been previously shown to improve predictions [18], structural information is often unavailable. As a result, this finding has immediate relevance in conservation analysis. (3) Many of the conservation methods that explicitly incorporate amino acid similarity fail to consistently improve upon the simple Shannon entropy method. (4) As compared to identifying catalytic sites and residues near ligands, all the conservation methods tested are only weakly predictive in identifying residues in the protein-protein interface. This confirms previous analyses [9, 11] and suggests that conservation should only be used as one component in ensemble methods for predicting interaction interfaces [27, 28].

Overall, our testing demonstrates that Jensen-Shannon divergence, used with our window method to incorporate information from sequential amino acids, provides a fast, state-of-the art method for identifying functionally important residues via conservation analysis. This combined approach performs significantly better than Shannon entropy, which is likely the most commonly used method in conservation analysis. Moreoever, we find that our simple heuristic for incorporating the conservation scores from sequentially neighboring amino acids results in improved performance for all methods tested; this suggests that further development of conservation analysis methods should focus on better exploiting the signal from neighboring residues. Finally, our data sets and testing methodology provide a comprehensive framework for gaining an empirical understanding of the real-world performance of using conservation scores to identify functionally important sites, and analysis similar to the one performed here should be useful in evaluating new proposed conservation measures.

2.2 Methods and Algorithms

2.2.1 Conservation scores

Brief descriptions of all methods, previous and new, are given below. For most methods, there are a number of parameters to optimize. We have explored the space of reasonable settings and report the best found parameter settings below. See Valdar [30] for a more complete discussion of similar methods and their evolution.

Preliminaries

Each method takes as input a MSA M of length L over N sequences. Let M_C denote the C-th column of the alignment, and M_{C_i} denote the symbol in column C of sequence i. $M_{C_i} \in AA$, where AA is the 21 element set of amino acids plus the gap symbol.

Gaps. Any column that is more than 30% gaps is ignored in the analysis presented here, because a column with many gaps is unlikely to be functionally important. Additionally, a simple gap penalty was applied to all methods except Rate4Site, which handles gaps in its software. In particular, each raw column score is multiplied by the fraction of non-gapped positions in the column [30]. If sequence weighting is used, the gap penalty is weighted as well. We also performed the analyses ignoring all columns with gaps, and our overall conclusions did not change (data not shown).

Sequence Weighting. An alignment will often contain sequences at a range of evolutionary distances. If an alignment consists of several very similar sequences, all columns may look conserved, and it will be difficult to discriminate positions under evolutionary pressure from those that are not. We implemented the sequence weighting method proposed in Henikoff and Henikoff [37] that rewards sequences that are "surprising." Sequence weighting is used with all methods and results given below, except for Rate4Site, which builds an evolutionary tree as the first part of its analysis.

Estimating probabilities. Let $\mathbf{p_C}$ be the distribution of the set AA in column C; $\mathbf{p_C}$ is computed below using the observed (weighted) frequency of each symbol of AA in the column, with a pseudocount of 10^{-6}.

Previous Methods

We first describe the six previous methods.

Shannon Entropy of Residues. Shannon entropy (SE) [35] is the one of the simplest and most

common measures of conservation at a site (e.g., [38, 39]). It is defined for a column C as:

$$SE(\mathbf{p_C}) = -\sum_{\alpha \in AA} p_C(\alpha) \log p_C(\alpha). \tag{2.1}$$

The SE is smallest for a column with complete conservation.

Shannon Entropy of Residue Properties. The previous method does not take into account biochemical similarity between amino acids. Instead of treating the amino acids as distinct symbols in the entropy calculation, several groups [33, 34] have proposed partitioning the amino acids into stereochemically defined sets, and then computing the entropy of the column with respect to these sets. We refer to this conservation scoring method as property entropy (PE). We use the following grouping [34]: aliphatic [AVLIMC], aromatic [FWYH], polar [STNQ], positive [KR], negative [DE], and [P].

Von Neumann Entropy. Caffrey et al. [9] introduced the use of von Neumann entropy (VNE) [40], a concept from quantum mechanics, as an information-theoretic measure of conservation that incorporates the physicochemical similarity between amino acids. The VNE of a column C is computed as:

$$VNE(C) = -Tr(\rho \log \rho) \tag{2.2}$$

where ρ is the density matrix of column C normalized so that $Tr(\rho) = 1$. The density matrix of a column is computed by creating a matrix where the diagonal elements are the relative frequencies of amino acids in each column (ignoring gaps and without a pseudocount), and all other entries are zero, and multiplying this by target frequencies for an amino acid similarity matrix. We use the BLOSUM62 matrix as suggested in Caffrey et al. [9].

Relative Entropy. Relative entropy (RE), or the Kullback-Leibler divergence, is often used to compare probability distributions [35]. The relative entropy conservation score for a column is defined as:

$$RE(\mathbf{p_C}, \mathbf{q}) = \sum_{\alpha \in AA} p_C(\alpha) \log \frac{p_C(\alpha)}{q(\alpha)}. \tag{2.3}$$

If the background distribution, $\mathbf{q}$, lacks gaps (as it does here), then $\mathbf{p_C}$ will ignore gaps as well. Magliery and Regan [7] have applied relative entropy in order to identify unconserved hypervariable positions, and Wang and Samudrala [31] have applied relative entropy to the problem of finding conserved positions. Unless otherwise stated, we use the overall amino acid distribution in the BLOSUM62 alignments as the background distribution.

Sum-of-pairs measures. The sum-of-pairs (SP) method scores the conservation of a column using a similarity matrix S, where $S(x, y)$ is the similarity score between amino acids x and y. Typically S is a matrix such as one from the BLOSUM series [41]. The SP method encapsulates the overall pairwise similarity between amino acids in a column. The SP measure for a column C is given by:

$$SP(C) = \frac{1}{\sum_i^N \sum_{j>i}^N w_i \cdot w_j} \cdot \sum_i^N \sum_{j>i}^N w_i \cdot w_j \cdot S(C_i, C_j), \tag{2.4}$$

where w_i and w_j are the sequence weights for the i-th and j-th sequences respectively. While transformations have been proposed to make all diagonal elements equal to one [12], or to give immutable amino acids greater self-similarity than mutable ones [30], we have found that untransformed matrices yielded the best performance. All results presented below use the untransformed BLOSUM62 matrix, unless otherwise indicated.

Rate4Site. In contrast to the methods described above, the Rate4Site (R4S) algorithm [36] uses a statistical model of evolution to estimate the rate of evolution, and thus the conservation, at each site. Briefly, a phylogenetic tree is constructed for the input alignment. The rates of evolution are assumed to follow a Gamma distribution, and this distribution is used as the prior in a Bayesian inference scheme. A low rate of evolution means high conservation at a position. We use the freely available source code with the default parameters.

New methods

We describe a new conservation scoring method and an extension that can be applied to any of the methods.

Jensen-Shannon Divergence Score. The Jensen-Shannon divergence (JSD) [42] quantifies the similarity between probability distributions. As compared to relative entropy, it has the advantages of being symmetric and bounded with a range of zero to one. A "background" amino acid distribution $\mathbf{q}$, estimated from a large sequence set, can be used to approximate the distribution of amino acid sites subject to no evolutionary pressure. Then, positions in an alignment that are found to have amino acid distributions very different from this background distribution are proposed to be functionally important or constrained by evolution. JSD is defined for a column C as:

$$D_{JS}(C) = \lambda RE(\mathbf{p_C}, \mathbf{r}) + (1 - \lambda) RE(\mathbf{q}, \mathbf{r}) \tag{2.5}$$

where: $\mathbf{r} = \lambda \mathbf{p_C} + (1 - \lambda)\mathbf{q}$, RE is relative entropy, $\mathbf{p_C}$ is the column amino acid distribution, $\mathbf{q}$ is

a background distribution, and λ is a prior weight. We use $\lambda = \frac{1}{2}$ and have found that it performs better than other options. Unless otherwise stated, we use the overall amino acid distribution in the BLOSUM62 alignments as the background distribution. Using alignment specific backgrounds can provide a slight improvement, but we have found it is not great enough to justify the added complexity.

While to the best of our knowledge, this is the first use of JSD to assess sequence conservation, it has been previously used in the context of comparing sequence profiles [43].

Incorporating sequential residues. Positions near in space and sequence to functionally important residues are known to be more conserved than average [44]. The conservation of spatial neighbors can be exploited to improve prediction of functionally important residues [18]. The conservation of spatial neighbors is stronger than that of positions near in sequence, but 3D structures are often unavailable. Thus we developed the following heuristic method to incorporate the conservation of positions near in sequence into the score for a column:

$$WindowScore(C) = \lambda S(C) + (1 - \lambda)\frac{\sum_{i \in window} S(i)}{|window|} \tag{2.6}$$

where $S(i)$ is the raw score of column i (in this case $D_{JS}(i)$) and $window$ is a set containing the indices of all columns in the window around column C. We find $\lambda = \frac{1}{2}$ and a window size of three residues on either side of C works well. This window technique can be applied to any conservation scoring method that gives columnwise scores. When discussing the windowed version of a method, we will append "+W" to the name of the method. Additionally, we call the non-windowed version of a method "basic."

2.2.2 Data Sets

We have created three data sets that reflect varying contexts in which conservation-based analysis is commonly applied. The data sets are by nature imperfect, as we rarely know all the functionally important residues in a protein. Indeed it is often not clear how to define "functionally important." Moreover, it is difficult to determine whether a position that appears to be conserved, but is not known to be functionally important, is constrained or simply has not had enough time to diverge. To account for this uncertainty, we construct the data sets to include different types of functional sites, with the hope that the shortcomings of one will be less prevalent in another. We will look for consistent results across these data sets, using various performance metrics, to judge the performance of conservation measures.

Catalytic Site Data Set. We created the first data set using known catalytic sites obtained from the Catalytic Site Atlas (CSA) [45], a literature derived database of enzyme active sites and catalytic residues. For each literature based entry in the CSA as of June 8, 2006, we obtained the 3D structure of the protein chain from the Protein Data Bank (PDB) [3]. The structures' sequences were then clustered at 95% sequence identity and redundant structures were removed. Sequence alignments for each remaining structure were then obtained from HSSP [46]. These alignments were filtered to improve alignment quality by removing sequences with more than 95% sequence similarity to the original CSA sequence or whose length was more than two standard deviations away from it. Any alignment with fewer than five sequences was removed. After filtering, 645 alignments with an average of ∼79 sequences per alignment and ∼1900 catalytic sites remained. The annotated catalytic sites for each protein serve as positives (i.e., functionally important residues), and all other residues are negatives.

We note that many positions in protein cores are conserved for structural reasons. We do not want to penalize methods for giving these likely non-catalytic positions high scores. However, many catalytic sites have low relative solvent accessibility (RSA); for example 5% of catalytic sites have 0% RSA [44]. To resolve this tension between leaving out known positives and excluding positions that are likely important but unannotated, we performed the analysis both with and without residues that have RSA less than 1%. There was little change in the relative performance of the measures on all data sets (see Supplementary Data). The results presented here include all columns and catalytic sites. Most sequences do not have known structures, thus this represents the more common scenario in which conservation analysis is applied.

Ligand Distance. The second data set is based on a less restrictive definition of functionally important. The increased conservation found in the binding sites of enzyme ligands [44] is used to compare methods without making many assumptions about the type of functional site sought.

The Enzyme Commission (EC) [47, 48] provides a classification of known enzymes into functional groups. For each EC class, we retrieve all structures present in the PDB. For each structure with resolution better than 2.5Å, we check to see if it contains bound ligands similar to the substrates required for the reaction catalyzed by the enzyme, using a similarity cutoff of 50% as defined by PDBSum [49]. The structures' sequences are then clustered at the level of 95% sequence identity within each EC class, and a non-redundant set is kept for analysis. For each structure that remains in each EC class, we download the alignment from HSSP and filter it as for the catalytic site data set. For each structure, we put all residues within 4Å of any ligand atom into the set of putative positive

residues. This may include some positions that are not functionally important, but the area around the active site contains a strong enough conservation signal that we are able to distinguish between methods by the number of highly conserved positions each predicts near ligands. All remaining residues comprise our set of negatives. We are left with 789 alignments with an average of ∼92 sequences per alignment. The alignments span 495 EC classes and provide an average of ∼1.6 alignments per class. We also performed the analysis excluding all residues that have less than 1% solvent accessibility and obtained similar results (see Supplementary Data).

Protein-Protein Interfaces. We use the data set of Caffrey et al. [9] consisting of 64 protein-protein interfaces: 42 homodimers, 12 heterodimers, and 10 transient complexes. For each interface, they provide an alignment of close homologs and an alignment of diverse homologs. We present results for the close homolog alignments; performance is similar on the diverse alignments (Supplementary Data). Interface residues, comprising the set of positives, are defined as those losing more than 1% relative solvent accessibility on complex formation; as done by Caffrey et al. [9], we compute relative solvent accessibility using NACCESS [50, 51] with a probe size of 1.4Å. All other residues are the negatives. We also evaluated the methods by removing all positions that have less than 1% solvent accessibility in the monomer; results were similar on this modified data set (see Supplementary Data).

2.2.3 Evaluation Methods

Conservation scoring methods are compared on each data set by considering how well they rank the positive set of functionally important residues, as well as by computing receiver operator characteristic (ROC) curves.

For ROC analysis, a ROC curve is constructed for each method on each alignment, and all the ROC curves for a method are averaged across all alignments to obtain its overall curve. For the ligand distance data set, ROC curves are first averaged over each EC class and then averaged across classes; this is done because some EC classes have more alignments than others. We report the area under the ROC curve (AUC) at a range of false positive rates: 0.1 ($AUC_{0.1}$), 0.5 ($AUC_{0.5}$) and 1.0 (AUC_1). The higher the AUC, the better the method has done at identifying functional residues.

In the rank analysis, for each alignment, we compute the conservation scores for all columns and note the rank of the known functionally important columns. We report the fraction of the top 30 ranked columns that are functionally important [31]; however, since the number of positives may be less than 30, we normalize the statistic so that perfect performance (i.e., all possible positives in the

top 30 predictions) gets a score of one. These top-30 statistics are averaged over all alignments, and in the case of the ligand distance data set, are first averaged over EC class and then averaged over the classes.

We use the Friedman test, as implemented in Matlab, to judge whether the performance statistics (e.g., AUC_1) for the methods are significantly different. For the CSA and PPI data sets, when judging each statistic, comparisons of its value on each alignment are considered; for the ligand distance data, comparisons are made between its averaged value for each EC class. Since for each statistic, the values for all pairwise combinations of methods are compared, we further apply a Bonferroni correction to judge significance. The difference in performance of two methods using a particular statistic is called statistically significant if the p-value computed using the Friedman test with a Bonferroni correction is less 0.05.

2.3 Results

We evaluate the seven representative methods on their ability to identify functional sites in the three data sets. The performance statistics—averaged AUCs and top-30—for all basic methods are summarized for the catalytic site (CSA) data set in Table 2.1, the ligand distance data set in Table 2.2, and the close homolog protein-protein interface (PPI) data set in Table 2.3. The relative performance of these methods using the AUC_1 performance statistic is also depicted graphically for the CSA and ligand distance data sets in Figure 2.2. These relationships are not shown for the PPI data set, as the only significant differences on it between methods at the AUC_1 level involve comparisons with the worst performing method. The top-30 improvement provided by using the window heuristic on each method on each data set is given in Table 2.5. We now describe our main findings in detail.

JSD is not significantly outperformed by any other basic method. Tables 2.1, 2.2, and 2.3 show that JSD, RE and R4S perform better than the other four basic methods when considering any of the performance statistics on any of the data sets. The significance chart in Figure 2.2 illustrates that JSD and R4S perform significantly better at the AUC_1 level on both the CSA and ligand distance data sets than all the other methods, including RE. For the CSA data set, JSD outperforms R4S on all criteria, whereas on the ligand distance data set, R4S outperforms JSD using all criteria (Tables 2.1 and 2.2). The differences between these methods on the PPI data set are not significant, but Table 2.3 shows that JSD and RE are the best performing methods on this data set. Overall, the results are similar if we consider the top-30 statistic; JSD, RE, and R4S are all

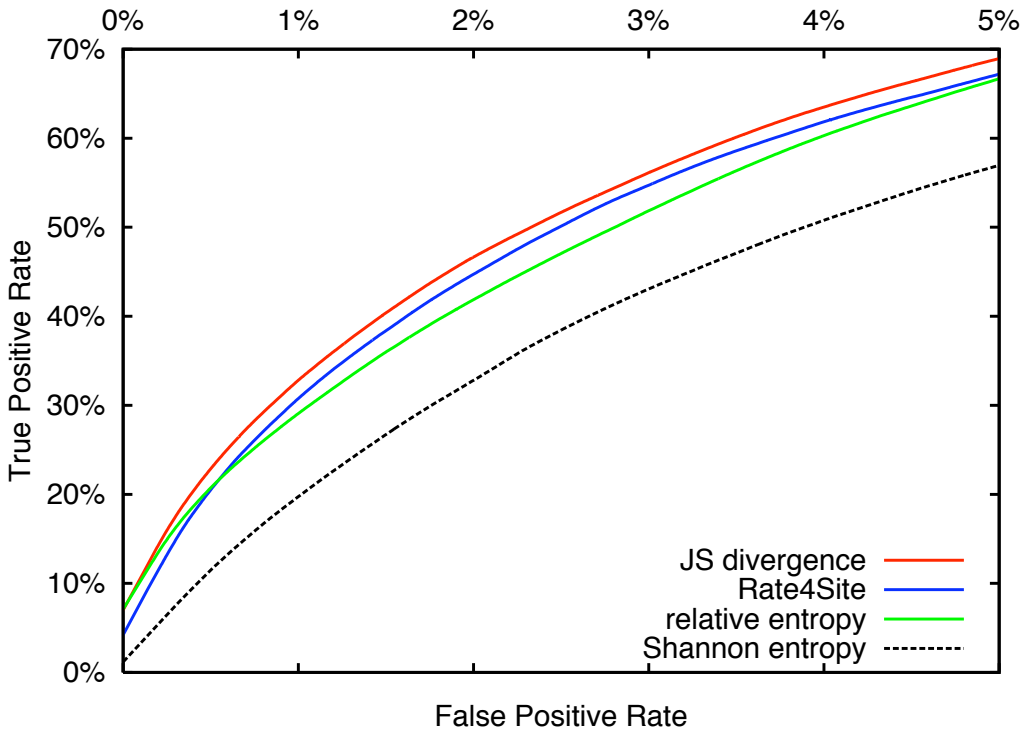

Figure 2.1: High confidence region of the ROC curves for Shannon entropy, JS divergence, Rate4Site, and relative entropy on the catalytic site data set. The three methods all perform significantly better than Shannon entropy.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0524	0.4248	0.9235	0.6783
Property entropy	0.0338	0.3780	0.8749	0.4328
von Neumann entropy	0.0499	0.4211	0.9166	0.6462
Sum of pairs	0.0528	0.4291	0.9271	0.6374
Relative entropy	0.0599	0.4436	0.9428	0.7120
Rate4Site	0.0615	0.4451	0.9412	0.7240
JS divergence	**0.0623**	**0.4464**	**0.9440**	**0.7338**

Table 2.1: Performance statistics for all methods on the catalytic site data set. Area under the ROC curve is given for the 0.1 ($AUC_{0.1}$), 0.5 ($AUC_{0.5}$) and 1.0 (AUC_1) false positive rates. Top-30 is the normalized fraction of the top 30 scoring sites that are functionally important (see text). The best scores are in bold.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0093	0.3238	0.8036	0.3960
Property entropy	0.0049	0.2813	0.7590	0.2822
von Neumann Entropy	0.0089	0.3138	0.7934	0.3816
Sum of pairs	0.0086	0.3141	0.7898	0.3759
Relative entropy	0.0098	0.3311	0.8119	0.4076
Rate4Site	**0.0109**	**0.3394**	**0.8238**	**0.4312**
JS divergence	0.0107	0.3345	0.8153	0.4220

Table 2.2: Performance statistics for all methods on the ligand distance data set. See Table 2.1 for a description of the statistics.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0060	0.1352	0.5203	0.1692
Property entropy	0.0037	0.1160	0.4968	0.1225
von Neumann entropy	0.0059	0.1367	0.5265	0.1670
Sum of pairs	0.0069	0.1380	0.5217	0.1806
Relative entropy	**0.0079**	**0.1529**	**0.5468**	0.1948
Rate4Site	0.0075	0.1466	0.5433	0.1772
JS divergence	**0.0079**	0.1516	0.5437	**0.1960**

Table 2.3: Performance statistics for all methods on the close homolog protein interface data set. See Table 2.1 for a description of the statistics.

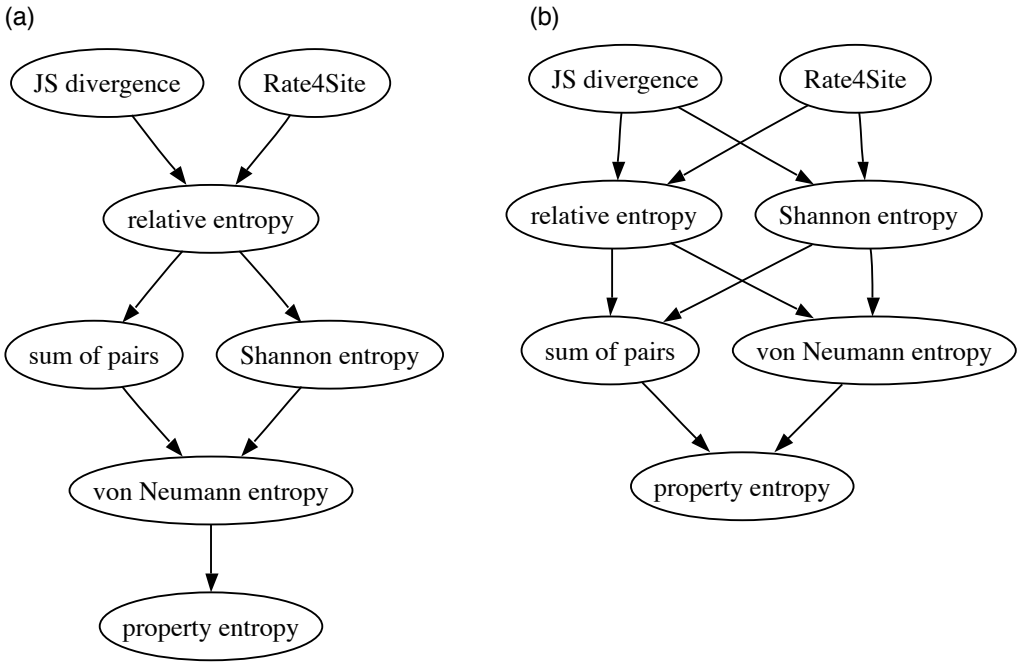

Figure 2.2: Significance relationships at AUC_1 level for the (a) catalytic site and (b) ligand distance data sets. An edge from method X to method Y means that method X performs significantly better than method Y. A path between two nodes implies a significant difference as well.

significantly better than the other methods on the catalytic site and ligand distance data sets. Note that while SE is probably the most commonly used method for identifying functionally important residues via conservation analysis, our evaluation shows that in all settings tested it is outperformed by other methods (e.g., Figure 2.1).

JSD performs similarly to R4S, but is much faster. Overall, JSD and R4S perform similarly; none of the differences in performance observed between them on any of the data sets using any of the statistics are significant (Figure 2.2). However, JSD and the other information-theoretic methods have a significant advantage over R4S when considering run time. Table 2.4 gives (processor) running time statistics for several methods on a benchmark set of 25 randomly chosen alignments from the CSA data set. R4S took over 2.5 hours (9563.22 seconds) to score these 25 alignments while JSD required only 11.81 seconds; JSD finishes scoring all 25 in less time than R4S needs to score the smallest alignment. RE's running time is similar to JSD's. In light of this and the performance results, JSD is the best method for the estimation of conservation in contexts where speed is an issue, such as large, genome-scale analysis.

Incorporating the conservation of sequentially adjacent positions improves performance. Our heuristic for exploiting conservation scores within a sequence window around the residue of in-

Method	Min Time	Max Time	Average	Total
Shannon entropy	0.18s	1.18s	0.45s	11.18s
JS divergence	0.20s	1.21s	0.47s	11.81s
Rate4Site	18.66s	1976.38s	382.53s	9563.22s

Table 2.4: Running time on a set of 25 alignments randomly selected from the catalytic site data set. The Jensen-Shannon divergence takes several orders of magnitude less time than Rate4Site and provides competitive performance. All information theoretic methods have similar running times.

terest can be applied to any scoring method that produces independent column scores. Using a window of size seven (three residues on either side of the current residue), all methods improve on each of the three data sets (Table 2.5), as judged by top-30. The results are similar for the other statistics (see Supplementary Data for all performance statistics for windowed approaches). Note that the window method improves predictions for all types of functional sites, not just those with low conservation. In fact, as Table 2.5 shows, the improvement is greater for sites with high conservation (catalytic residues and residues near ligands) than for sites in protein interfaces.

Figure 2.3 shows the improvement on the CSA data set for SE and JSD when our window approach is used. The figure depicts the high-confidence region of the ROC curve. The difference between JSD+W and SE illustrates the improvement provided by methods introduced in this chapter; at a false positive rate of 2%, JSD+W identifies over 50% of the true positives while SE finds only about 30%. Note that when SE is extended to incorporate the conservation of sequentially adjacent positions, it performs nearly as well as the basic JSD method. This highlights the power of simply using the window approach with existing scoring methods. The consistent improvement provided by the window heuristic suggests that it can improve predictions in a range of settings.

Incorporating relationships between amino acids is not always helpful. Three of the methods considered, VNE, SP and PE, attempt to incorporate information about the similarity of amino acids. One would expect that, since pairs of amino acids have differing physicochemical similarities, incorporation of such information would improve upon other methods that do not. Our evaluation framework allows us to assess this claim by characterizing the performance of VNE, SP and PE relative to the commonly used SE.

Figure 2.2 shows that using the AUC_1 criterion, SE is significantly better than VNE, PE and SP on the ligand distance data set, and significantly better than VNE and PE on the CSA data set. While SP performs better than SE using the AUC_1 criterion, the difference is not statistically

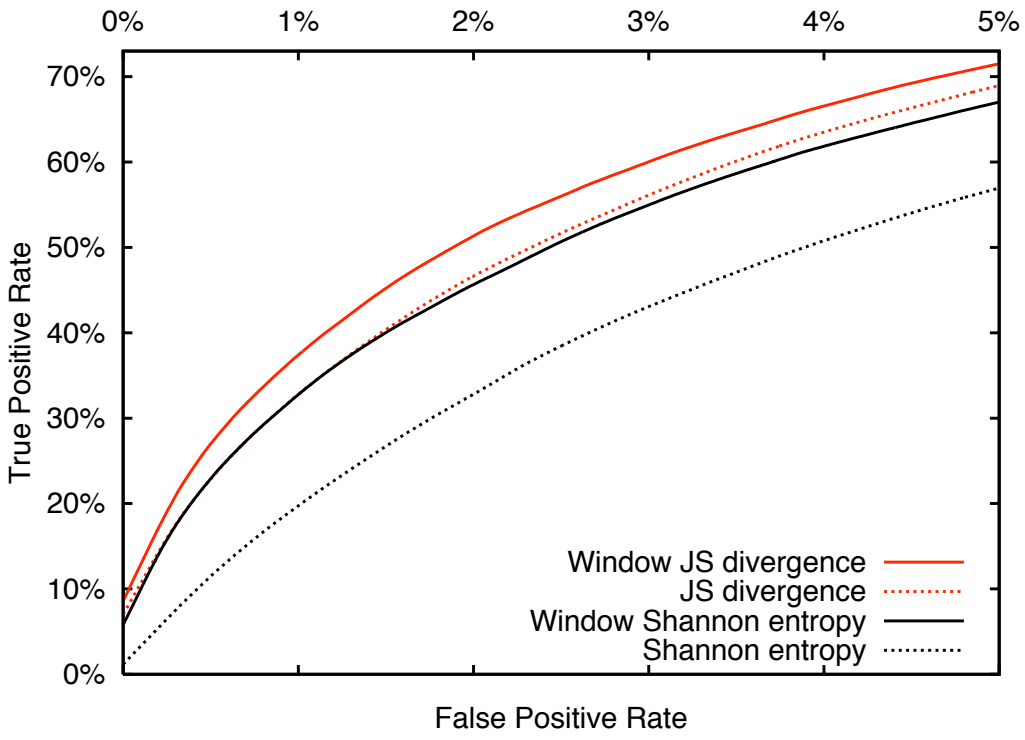

Figure 2.3: High confidence ROC curves demonstrating the improvement for Shannon entropy and JS divergence when used with the window method on the CSA set. The difference between Shannon entropy (dotted black) and Window JS Divergence (solid red) is the improvement provided by methods introduced in this chapter. Similar improvement is seen across methods and data sets.

Method	CSA	Ligand	PPI
Shannon entropy	0.6783	0.3960	0.1692
Shannon entropy+W	**0.7077**	**0.4583**	**0.2000**
Property entropy	0.4328	0.2822	0.1225
Propert entropy+W	**0.5928**	**0.3840**	**0.1772**
von Neumann entropy	0.6462	0.3816	0.1670
von Neumann entropy+W	**0.6979**	**0.4422**	**0.1960**
Sum of pairs	0.6374	0.3759	0.1806
Sum of pairs+W	**0.6995**	**0.4264**	**0.2034**
Relative entropy	0.7120	0.4076	0.1948
Relative entropy+W	**0.7507**	**0.4546**	**0.2205**
Rate4Site	**0.7240**	0.4312	0.1772
Rate4Site+W	0.7197	**0.4795**	**0.2205**
JS divergence	0.7338	0.4220	0.1960
JS divergence+W	**0.7539**	**0.4703**	**0.2205**

Table 2.5: Improvement in top-30 performance provided by the window heuristic for all methods. The better score between the basic method and its windowed version is in bold. Full statistics are provided in the Supplementary Data.

significant. The differences between these methods on the PPI data set are not significant; however, SE performs best of the three as judged by top-30, SP performs best as judged by $AUC_{0.1}$ and $AUC_{0.5}$, and VNE performs best as judged by AUC_1.

We also evaluated using different BLOSUM matrices with VNE and SP. We found that the choice of matrix does not change our overall results. For the SP method, we additionally experimented with alignment-specific matrices. In particular, for each alignment considered, we computed the average pairwise sequence identity and selected the nearest of BLOSUM45, BLOSUM62, and BLOSUM80. This scheme also did not improve overall performance (see Supplementary Table 2.13).

These results highlight the need for large-scale evaluation. While it might be expected that PE, VNE, and SP would improve on SE, none provide any significant gain. In fact, in several settings, some perform significantly worse than SE. VNE was introduced for the prediction of protein-protein interface residues [9], but is not significantly better on the PPI data set as judged by any of the four statistics tested. PE was introduced for the analysis of ligand recognition in transport proteins, and SP for the analysis of DNA-binding proteins. It is possible that these methods could achieve better performance in other specific settings, but the three contexts investigated here are quite common and similar to those in which they were introduced.

Identifying residues in the protein-protein interfaces from conservation alone is difficult. Recently, conservation analysis has been employed to predict and analyze protein-protein interaction sites [9, 10, 27]. Several of these groups have found that it is difficult to predict the interface using various measures of conservation. Here, we find that none of the seven conservation measures studied perform particularly well in identifying residues in protein interfaces (Table 2.3). We report statistics from the close homolog alignments, but results are similar for the diverse homolog set (see Supplementary Data). The AUC_1 values for all methods are approximately .55, as compared to much better performance in identifying catalytic site residues ($AUC_1 \approx .95$) and the sites near ligands ($AUC_1 \approx .80$). However, while the conservation signal on the interface is weak, it is still detectable; all methods other than PE performed significantly better than random guessing. This suggests that conservation alone should not be used to predict residues in protein-protein interfaces. However, it is an important component of ensemble based approaches [28].

2.4 Discussion and Conclusion

Despite the prevalence of conservation based analysis, there are few agreed upon best practices. This work establishes an empirical understanding of the relationships between approaches. We

describe several methods for quantifying conservation and introduce a method based on the Jensen-Shannon divergence, as well as a heuristic for incorporating the conservation signal from sequentially neighboring residues. We then quantitatively compare the performance of all methods in three realistic settings: the identification of catalytic sites, residues near ligands, and residues comprising protein-protein interfaces.

Our evaluation demonstrates that methods such as JSD and RE that incorporate a background amino acid distribution are preferable to SE (Figure 2.1). R4S also provides similar improvement over SE, but is quite slow in comparison to the information theoretic methods (Table 2.4). The speed of JSD would allow researchers to modify alignments and re-predict functional sites on the fly. It also makes large-scale analysis faster and more appealing. While JSD and RE are similar measures, overall RE does not perform quite as well as JSD. RE is unbounded; events that are unlikely according to the background or column distributions tend to contribute more to the RE score than to the JSD score, and this likely causes the difference in performance between the two methods.

We also demonstrate that our window heuristic provides a way to boost the conservation signal, and thus performance, even in the absence of structural information. This improvement is seen across methods and data sets. The approach is fast, flexible, and can be applied to any method that produces column scores.

Perhaps most surprisingly, we find that several methods that intend to improve conservation estimation by incorporating amino acid similarity fail to provide any significant improvement over methods that ignore the underlying chemistry. In fact, some perform significantly worse than Shannon entropy. While it may be the case that incorporating amino acid similarity is not critical for identifying functional sites, it is more likely that the existing set of methods are not adequate, and other as yet undeveloped methods, may be able to exploit better the similarities between amino acids. Additionally, it is possible that the data sets of known functional sites are biased towards absolutely conserved residues, and thus incorporating relationships between amino acids is not essential for good performance on them.

The poor performance of all methods on the protein-protein interface data set demonstrates that the difficulties encountered in previous attempts [9, 27] exist across a range of conservation methods. It is likely that the results could be improved by dividing the data set into transient and obligate interactions [11] and further dividing the interface into central and peripheral residues. Nevertheless, it is clear that conservation alone is insufficient to predict all residues in protein-protein interfaces.

When interpreting the predictions made by a conservation-based method, it is natural to ask

whether a site is important for maintaining structure, for catalysis, or for binding ligands, other proteins or DNA. Conservation alone cannot distinguish among these possibilities; however, features such as amino acid composition, electrostatic potential, and known or predicted structural properties (e.g., secondary structure and solvent accessibility), used along with conservation, can be used within machine learning methods to identify particular types of functional residues (e.g., [27, 29]).

Overall our results highlight the necessity for rigorous evaluation of conservation methods. Conservation analysis is beginning to be applied in settings where the signal is not strong (e.g., the prediction of protein interaction sites). Thus, comprehensive analyses such as the one performed here are increasingly important in order to develop an empirical understanding of the strengths and weaknesses of various methods; this understanding can then be used to guide development of more powerful techniques for estimating sequence conservation in diverse biological settings.

2.5 Supplementary Data

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0524	0.4248	0.9235	0.6783
Shannon entropy+W	0.0613	0.4414	0.9374	0.7077
Property entropy	0.0338	0.3780	0.8749	0.4328
Property entropy+W	0.0498	0.4138	0.9093	0.5928
von Neumann entropy	0.0499	0.4211	0.9166	0.6462
von Neumann entropy+W	0.0593	0.4376	0.9342	0.6979
Sum of pairs	0.0528	0.4291	0.9271	0.6374
Sum of pairs+W	0.0595	0.4509	0.9362	0.6995
Relative entropy	0.0599	0.4436	0.9428	0.7120
Relative entropy+W	0.0649	0.4484	0.9475	0.7507
Rate4Site	0.0615	0.4451	0.9412	0.7240
Rate4Site+W	0.0616	0.4442	0.9416	0.7197
JS divergence	0.0623	0.4464	0.9440	0.7338
JS divergence+W	0.0654	0.4676	0.9461	0.7539

Table 2.6: Comparison of performance statistics for all methods with and without windows on the catalytic site dataset. Area under the ROC curve is given for the 0.1 ($AUC_{0.1}$), 0.5 ($AUC_{0.5}$) and 1.0 (AUC_1) false positive rates. Top-30 is the normalized fraction of the top 30 scoring sites that are functionally important (see text).

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0093	0.3238	0.8036	0.3960
Shannon entropy+W	0.0128	0.3461	0.8278	0.4583
Property entropy	0.0049	0.2813	0.7590	0.2822
Property entropy+W	0.0098	0.3193	0.7987	0.3840
von Neumann Entropy	0.0089	0.3138	0.7934	0.3816
von Neumann entropy+W	0.0122	0.3408	0.8223	0.4422
Sum of pairs	0.0086	0.3141	0.7898	0.3759
Sum of pairs+W	0.0107	0.3360	0.8140	0.4264
Relative entropy	0.0098	0.3311	0.8119	0.4076
Relative entropy+W	0.0121	0.3499	0.8336	0.4546
Rate4Site	0.0109	0.3394	0.8238	0.4312
Rate4Site+W	0.0135	0.3605	0.8474	0.4795
JS divergence	0.0107	0.3345	0.8153	0.4220
JS divergence+W	0.0132	0.3521	0.8349	0.4703

Table 2.7: Comparison of performance statistics for all methods with and without windows on the ligand distance data set. See Table 2.6 for a description of the statistics.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0060	0.1352	0.5203	0.1692
Shannon entropy+W	0.0091	0.1504	0.5382	0.2000
Property entropy	0.0037	0.1160	0.4968	0.1225
Property entropy+W	0.0068	0.1383	0.5234	0.1772
von Neumann entropy	0.0059	0.1367	0.5265	0.1670
von Neumann entropy+W	0.0092	0.1528	0.5440	0.1960
Sum of pairs	0.0069	0.1380	0.5217	0.1806
Sum of pairs+W	0.0092	0.1486	0.5332	0.2034
Relative entropy	0.0079	0.1529	0.5468	0.1948
Relative entropy+W	0.0097	0.1602	0.5548	0.2205
Rate4Site	0.0075	0.1466	0.5433	0.1772
Rate4Site+W	0.0105	0.1612	0.5618	0.2205
JS divergence	0.0079	0.1516	0.5437	0.1960
JS divergence+W	0.0105	0.1613	0.5548	0.2205

Table 2.8: Comparison of performance statistics for all methods with and without windows on the close homolog protein interface data set. See Table 2.6 for a description of the statistics.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0067	0.1399	0.5205	0.1823
Property entropy	0.0040	0.1223	0.5051	0.1242
von Neumann entropy	0.0064	0.1375	0.5201	0.1783
Sum of pairs	0.0076	0.1403	0.5162	0.1846
Relative entropy	**0.0088**	**0.1483**	**0.5373**	**0.2064**
Rate4Site	0.0079	0.1436	0.5309	0.1921
JS divergence	0.0086	0.1475	0.5339	0.2030

Table 2.9: Performance statistics for all methods on the diverse homolog protein interface data set. See Table 2.6 for a description of the statistics. The best scores for each statistic are in bold.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0567	0.4382	0.9352	0.7137
Property entropy	0.0446	0.4084	0.9065	0.5774
von Neumann entropy	0.0545	0.4307	0.9297	0.6946
Sum of pairs	0.0572	0.4396	0.9367	0.6962
Relative entropy	0.0634	0.4525	0.9487	0.7323
Rate4Site	0.0652	0.4509	**0.9503**	0.7384
JS divergence	**0.0655**	**0.4625**	**0.9503**	**0.7422**

Table 2.10: Performance statistics for all methods on the catalytic site data set ignoring columns with less than 1% relative solvent accessibility. See Table 2.6 for a description of the statistics. The best scores are in bold.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0109	0.3461	0.8307	0.4751
Property entropy	0.0080	0.3245	0.8088	0.4152
von Neumann entropy	0.0105	0.3379	0.8229	0.4570
Sum of pairs	0.0097	0.3357	0.8161	0.4508
Relative entropy	0.0111	0.3468	0.8320	0.4734
Rate4Site	**0.0126**	**0.3599**	**0.8486**	**0.5081**
JS divergence	0.0120	0.3518	0.8370	0.4909

Table 2.11: Performance statistics for all methods on the ligand distance data set ignoring columns with less than 1% relative solvent accessibility. See Table 2.6 for a description of the statistics. The best scores are in bold.

Method	$AUC_{0.1}$	$AUC_{0.5}$	AUC_1	Top-30
Shannon entropy	0.0066	0.1501	0.5505	0.2182
Property entropy	0.0061	0.1463	0.5465	0.1977
von Neumann entropy	0.0065	0.1526	0.5587	0.2154
Sum of pairs	0.0076	0.1500	0.5471	0.2188
Relative entropy	**0.0085**	0.1591	0.5639	**0.2336**
Rate4Site	**0.0085**	**0.1630**	**0.5746**	0.2330
JS divergence	0.0084	0.1581	0.5625	0.2308

Table 2.12: Performance statistics for all methods on the close homolog protein interface data set ignoring buried columns (see main text). See Table 2.6 for a description of the statistics. The best scores for each statistic are in bold.

Method	CSA	Lig. Dist.	PPI
von Neumann entropy BL45	0.9152	0.7915	0.5252
von Neumann entropy BL62	0.9166	0.7934	0.5265
von Neumann entropy BL80	0.9177	0.7944	0.5276
Sum of pairs BL45	0.9216	0.7861	0.5248
Sum of pairs BL62	0.9271	0.7898	0.5217
Sum of pairs BL80	0.9287	0.7929	0.5230
Per-alignment sum of pairs	0.9241	0.7851	0.5256
Relative entropy BL45	0.9430	0.8123	0.5459
Relative entropy BL62	0.9428	0.8119	0.5468
Relative entropy BL80	0.9433	0.8121	0.5479
JS divergence BL45	0.9440	0.8153	0.5436
JS divergence BL62	0.9440	0.8153	0.5443
JS divergence BL80	0.9444	0.8153	0.5443

Table 2.13: AUC_1 performance statistics for all data sets for all methods that use BLOSUM matrices in some way. We have presented a range of BLOSUM distances to show that the choice of particular matrix matters very little to the results. For the sum of pairs of method we also include the results when an alignment specific matrix is used (designated by Per-alignment sum of pairs).

Chapter 3

Characterization and Prediction of Residues Determining Protein Functional Specificity

3.1 Introduction

Proteins can be classified into thousands of families on the basis of similar sequence patterns, shared structural motifs, experimentally determined common functions, or combinations thereof. The proteins within a single family usually share a common general function, but can exhibit a range of more specific functions. The enzymes provide many examples of this sort of family organization. For example, all members of the nucleotidyl cyclase family transform nucleotide triphosphates into cyclic monophosphates, but some act on ATP while others act on GTP. Similar behavior is seen among DNA-binding proteins, proteins mediating protein-protein interactions, and membrane proteins.

A set of proteins within a family that, as described above, share a specific function that is not common to the entire family have been called specificity groups [52]. Specificity groups within a protein family can be determined in a number of ways; for example, within a homologous protein family, the sets of orthologs can each comprise a specificity group. In many families, the amino acids present in a small number of sequence positions determine the particular functional specificity of member proteins. Identification of these specificity determining positions (SDPs) for a protein family is important not only because it provides insight into the mechanisms by which nature achieves its

astonishing functional diversity, but also because it enables the assignment of specific functions to uncharacterized proteins [16]. In addition, since SDPs are usually involved in the identification and binding of substrate molecules, knowledge of SDPs can be relevant to drug design, protein engineering, and pathway analysis.

Nearly all computational methods for identifying SDPs attempt to identify columns in multiple sequence alignments (MSAs) whose amino acid composition is related to the division of the sequences into functional specificity groups. Several early computational methods apply a range of statistical and phylogenetic techniques to the problem [15, 53, 54]. More recently, a number of information-theoretic methods have been developed [16, 17, 52, 55–58]. Other statistical, evolutionary and structural approaches have also been applied to the problem [59–66]. Several methods address the more difficult problem of additionally identifying family subgroup definitions [54, 67–70].

While there are many SDP prediction methods, their further development has been hindered by the small number of proteins for which exhaustive site-directed mutagenesis experiments have identified residue substitutions that switch functional specificity. Previous computational studies and evaluations have used from 2–13 alignments [16, 17, 58, 61–64, 67]. Perhaps as a result, different existing methods encode different assumptions about the column patterns in MSAs that are indicative of a role in determining specificity: some reward columns showing amino acid conservation within specificity groups, and others reward columns with little amino acid overlap between groups. Overall, it is not clear what types of amino acid column patterns in MSAs typify SDPs, what physicochemical properties of SDPs should be exploited for prediction, or how well existing approaches perform relative to each other.

This work addresses these problems by automating a process often undertaken by hand to recognize SDPs in the absence of mutation studies (e.g., see [16, 17]). We build a data set consisting of hundreds of enzyme protein families annotated with specificity groupings and putative SDPs. Using this large data set, we make the following contributions to the analysis and prediction of SDPs. First, we find that putative SDP columns in MSAs are distinct from the residue background with respect to their amino acid distribution, secondary structure distribution, and relative solvent accessibility. In addition, the observed column patterns indicate that amino acid properties such as polarity and size are less conserved between specificity groups in SDP columns than in all columns, suggesting that these properties are used to distinguish among similar ligands. Second, we demonstrate that alignment columns in which at least one specificity group displays both amino acid conservation and low overlap in amino acid usage with other groups are likely important for indicating specificity. This result from our large-scale column pattern analysis is consistent with the recent findings of [64]

on a diverse experimentally-verified set of SDPs from 13 families. Third, we test a representative set of current sequence-based methods that use MSAs and known specificity groupings to identify SDPs [16, 17, 58, 62], and show that they provide surprisingly little improvement over *GroupSim*, a simple method introduced here that uses the same information. Finally, we present a prediction heuristic that considers the conservation of neighboring positions and demonstrate that it improves the performance of all tested SDP prediction methods on our data set. In particular, *GroupSim* combined with the conservation window heuristic outperforms all previous methods tested in predicting SDPs on our large data set of enzyme families. Our main findings regarding performance do not change when considering the smaller set of experimentally-verified set of SDPs in [64], though the size of our data set allows us to better distinguish between methods.

Overall, our data set and testing methodology provide a framework for gaining an understanding of SDPs and SDP prediction methods, and have allowed us to show that even the simple *GroupSim* method introduced here exhibits state-of-the art performance. This suggests ample opportunities for further method development and performance improvement, and our framework provides the necessary foundation upon which this progress can be built.

3.2 Methods and Data

3.2.1 Data Set

Here we describe the computational pipeline used to build a data set of over 400 alignments of homologous enzyme domains each consisting of two specificity groups with columns likely important for specificity, as well as specificity groupings, identified. We refer to this data set as the EC-Pfam data set. While we apply our procedure to only enzymes here, it can be easily adapted to other protein types in the presence of a reliable classification of functional specificity.

We integrate data from several bioinformatics resources: protein sequences downloaded from SwissProt [71] on Jan. 23, 2007, 3D structures from the PDB [3], domain families from release 21.0 of Pfam [72], enzyme function classification from release 23 of the Enzyme (EC) database [48], and experimentally determined catalytic sites from release 2.2.1 of the Catalytic Site Atlas [45]. By defining families and specificity using a combination of Pfam, EC numbers, and sequence similarity, we avoid problems inherent in each approach. Pei et al. [67] attempted a large scale comparative analysis of SDP prediction methods, in which specificity groups were built by sequence similarity and all positions near ligands were considered positives. This previous approach has two problems.

First, specificity does not always follow sequence similarity [62, 73], and thus specificity groupings cannot reliably be obtained in this manner. Second, most residues near ligands are not important for specificity; in particular, many of these residues are well conserved and so a method that selects conserved residues would not identify any SDPs but would perform very well in their evaluation. We describe below how we address these problems.

Alignment Building. We build alignments for families of homologous domains for which we have a reliable way to divide the sequences into groups according to their functional specificity. We start by combining domain data from Pfam and enzyme data from EC. The Enzyme database provides a hierarchical classification of enzymes based on the reactions they catalyze. An enzyme is assigned four numbers, each representing a more specific classification. The first three numbers taken together usually identify the type of reaction catalyzed (e.g., 1.1.1.* identifies an oxidoreductase acting on the CH-OH group of donors with NAD(+) or NADP(+) as an acceptor). The fourth number identifies the particular substrate (e.g., 1.1.1.27 acts on L-lactate and 1.1.1.37 acts on malate). These classifications are based mainly on experimental evidence and do not correspond to sequence identity. By combining EC classifications with Pfam sequence motifs and the sequence similarity cutoffs described below, we ensure that our homologous family and subgroup assignments are supported by both experimental and sequence evidence.

For each Pfam domain, we find the EC assignments (if any) for each member sequence. We consider all pairs of EC numbers present in the Pfam family that overlap through the third position, e.g., 1.1.1.27 and 1.1.1.37; these represent enzymes with similar functions that are acting upon different substrates. We then use BLASTCLUST [74] to cluster all sequences found in the EC group pair by pairwise sequence identity; we require 30% sequence identity over at least 85% of the domain sequence to be a member of a cluster. Now, each cluster contains domain sequences from the same domain family with significant sequence identity and EC numbers that overlap through the third position. Sequences in a cluster are assigned to specificity groups according to their fourth level EC number, which corresponds to their specific substrate. For each specificity group, i.e., set of sequences in a cluster with matching full EC numbers, we remove very similar sequences (those with 95% sequence identity over 85% of both sequences). Any chains from the PDB that contain the domain, EC assignment, and relevant bound ligand or an experimentally identified catalytic site [45] are included in the specificity group regardless of their sequence similarity to one another. A bound ligand is considered relevant if it is at least 40% similar, as computed by the graph-match algorithm used in PDBSum [49]), to the ligand specified by the EC number. We only keep clusters where

both specificity groups contain at least four sequences. As some of the methods tested become very slow on large alignments, we also limit each specificity group to 50 sequences (selecting sequences uniformly at random if necessary). Finally, the cluster sequences are are aligned using ProbCons [75].

Selection of residues near ligands. In enzymes, SDPs are usually found around the active site, near ligands. When evaluating SDP predictions in the absence of experimental data, many researchers have used nearness to relevant ligands as a proxy for importance for specificity (e.g., see [16, 17]). We extend and improve this previous small-scale approach by developing an automated procedure that uses structural information and sequence-based criteria to identify positions likely important for specificity.

Each alignment described in the previous section includes chains from the PDB that contain the relevant domain, EC assignment, and bound ligand or catalytic site. We select residues near ligands in two ways. For each chain, if a relevant ligand is present, we find all residues with an atom within 5Å of a relevant ligand atom and add these to the set of "near ligand" residues. Since many enzymes do not have 3D structures in complex with their substrate, we also use catalytic sites as a proxy for the location of ligands and include all residues within 5Å of a catalytic site. When we refer to the set of residues "near ligands," we also include those found near catalytic sites unless we explicitly state otherwise.

Sequence-based filtering of columns near ligands. The set of residues near ligands and catalytic sites includes many sites that are not important for specificity—for example, sites that are of functional importance to the whole family and thus are conserved across the specificity groups. To remove columns that are unlikely to have an effect on specificity from the set of likely SDPs, we consider three sequence alignment-based filters. Each filter corresponds to a column pattern that has been suggested to indicate importance for specificity.

The *low-overlap filter* ($\mathcal{L}$) seeks to remove all columns for which there is significant amino acid overlap between the specificity groups. For the two group case, the specificity group with higher Shannon entropy (lower conservation) is selected, and the fraction of sequences in the group whose amino acids appear in the other group is found. If it is greater than 0.1, then the column is removed from the putative SDP set. To better handle improperly annotated and poorly aligned sequences, an amino acid must account for more than 5% of the more conserved group to count as a match. Columns that are conserved across the groups are removed by this filter, as are columns which are not conserved but have similar amino acid distributions within the groups. It can be extended to columns with more than two groups by averaging the overlap for each pair of specificity groups.

The *one-group-conserved filter* ($\mathcal{O}$) imposes an additional constraint. A column passes this filter if it passes the low-overlap filter and at least one of its specificity groups is conserved. Here we define conservation as Shannon entropy less than $\frac{2}{3}$ of a bit. (Shannon entropy of a column has a range of 0 for complete conservation to $\approx \log_2(20)$ bits when each amino acid is equally likely.) A column passes the *all-groups-conserved filter* ($\mathcal{A}$) if it passes the low-overlap filter and all of its groups are conserved as defined above. This is the strictest filter. Each filter is a stricter version of the previous (e.g., all columns passing the all-groups-conserved filter also pass the one-group-conserved filter). Table 3.1 illustrates how the filters treat several example columns.

	Columns					Filter	Requirements
Group 1	A	H	K	D	S		
	A	L	S	D	S	Low-Overlap ($\mathcal{L}$)	low group overlap
	A	K	R	D	S		
	A	A	K	D	S	One-Group-Cons. ($\mathcal{O}$)	low group overlap ≥ 1 group conserved
Group 2	A	H	A	F	N		
	A	L	A	Y	N		
	A	E	C	F	N	All-Groups-Cons. ($\mathcal{A}$)	low group overlap all groups conserved
	A	R	V	Y	N		
strictest filter passed:	$\emptyset$	$\emptyset$	$\mathcal{L}$	$\mathcal{O}$	$\mathcal{A}$		

Table 3.1: Alignment column filter behavior on five example columns. The five example columns contain two specificity groups. The empty set symbol, $\emptyset$, indicates that the first two columns do not pass any filters. The strictest filters that the third, fourth, and fifth columns pass are (respectively) the low-overlap, one-group-conserved, and all-groups-conserved filters.

The following analysis requires the distinction of "positive" and "negative" positions. We use each of the filters along with structural evidence to define sets of columns that are likely to be enriched with SDP. The set of positions within 5Å of a relevant ligand passing filter $\mathcal{X}$ is referred to as $SDP_{\mathcal{X}}$. Each filter leads to a different set, but our results are robust to the filter used (see Supplementary Analysis). Section 3.3.1 provides evidence that $SDP_{\mathcal{O}}$, corresponding to the one-group-conserved filter, should be used as the positive set. The set of all columns that do not pass any of the filters is used as the negative set.

Data set statistics. The raw data set consists of 435 alignments. To avoid biasing the data set to larger families with many specific functions, we filter it so that each EC-Pfam pairing is included in no more than one alignment. The full data set is available online. After filtering the 435 alignments, 106 with at least one column in $SDP_{\mathcal{O}}$ remain. Since the observed column patterns depend on the diversity of the alignments, we now provide some summary statistics. These alignments have an average length of 279 positions and contain an average of 41 sequences with a minimum of 11 and

a maximum of 100. The average pairwise sequence identities range from 27.2% to 66.2% with a mean of 42.5% and a standard deviation of 8.5%. The average pairwise sequence identities within specificity groups vary from 25.5% to 88.9% with a mean of 55.2% and a standard deviation of 14.2%. The filtered data set contains 489 putative SDPs.

Experimental Support for EC-Pfam data set. The Lactate/Malate dehydrogenase family has experimentally-determined SDPs and is also found in the EC-Pfam data set. It thus provides an opportunity to compare our data set with experimentally-determined results. A mutation of one residue from Gln to Arg is known to switch the specificity from lactate to malate [16]. Two positions in the alignment (Q117R and E123M) are placed in $SDP_{\mathcal{O}}$ by our automated framework, and the known SDP (Q117R) is in this set.

In addition, several statistical properties of our data set are similar to those of the largest available data set of experimentally-determined SDPs [64]. The percent of alignment columns identified as putative SDPs is 1.2 in the EC-Pfam data set and 1.7 in the experimental data set. In addition, Section 3.3.1 shows that the same two SDP column patterns are over-represented in both data sets. Though neither set of SDPs (computational or experimental) can be thought of as complete, the similarity of these properties between them lends support to our automated approach for building a data set of SDPs.

3.2.2 SDP Property Definitions

In the following analysis, we use amino acid property definitions from several sources. Secondary structure and solvent accessibility (of all chains, ignoring ligands) are taken from DSSP [76]. The eight DSSP states are reduced to helix (H, G, I), sheet (E, B), and loop/other (S, T, C). Amino acid property partitions are adapted from the following sources: charge [(R, H, K), (D, E), (A, N, C, Q, G, I, L, M, F, P, S, T, W, Y, V)] [30], hydrophobicity [(I, L, V, C, A, M, F), (G, Y, W, H, K, T, R, E, Q, D, N, S, P)] [77], size [(A, G, C, S), (V, T, N, P, D), (Q, E, H, K, R, F, Y, W, M, I, L)] [30], polarity [(H, R, K, E, D), (Q, T, S, N, C, Y, W), (A, G, V, L, I, P, F, M)] [44].

3.2.3 Evaluation Procedures

SDP prediction methods are compared by analyzing how well they rank the set of positive columns, $SDP_{\mathcal{O}}$. We use both box plots and precision-recall (PR) curves. To create the box plots, for each of the positives, we compute its rank by counting how many of the positive and negative columns score better than it. We find the minimum, maximum, median, and quartile ranks of the positive

columns for each method in each alignment. We then average each of these statistics over all the alignments and present the results as a box plot. For the PR curves, precision (TP/TP+FP) is plotted on the y-axis, and recall (TP/TP+FN) is plotted on the x-axis. In our PR analysis, a PR curve is constructed for each method on each alignment, and all the PR curves for a method are averaged across all alignments to obtain its overall curve. We use the method and code of Davis and Goadrich [78] for calculating the area under the curve (AUC). Higher AUC corresponds to better performance. Columns with more than 10% gaps overall or with a specificity group containing more than 30% gaps are not considered.

3.2.4 SDP Prediction Methods

We evaluate the performance of a representative set of existing methods for predicting SDPs from a MSA divided into specificity groups against a simple, baseline method. We do not include methods that predict specificity groups as well as SDPs in the evaluation, though such an evaluation would be possible with our data set.

Existing SDP Prediction Methods

Information theoretic methods are frequently used to predict SDPs, so we include several such methods in our evaluation. *Relative entropy (RE)* [16] was one of the first fully-automated information theoretic approaches suggested. Our implementation calculates the average relative entropy between all pairs of group amino acid distributions in a column.

SDPpred [17] has been shown to perform well in previous small-scale evaluations. It calculates column scores by measuring the *mutual information (MI)* between specificity groups and amino acids and comparing it to the *MI* of columns with shuffled amino acid compositions. We evaluate both *SDPpred* and the use of *MI* without shuffling.

Sequence Harmony (SH) [58] scores columns using a linear combination of entropies that rewards difference between the specificity groups without requiring conservation within each of the groups. We include *SH* because it was one of the first methods to explicitly focus on group difference. Columns with tie scores are differentiated by their nearness to other high scoring columns.

The *Xdet* method [62] is selected to represent a set of non-information theoretic methods with similar motivations. It calculates, for each column, the correlation between the similarity of all observed amino acid pairs and the functional similarity of the proteins they represent. Columns in which proteins with similar amino acids have similar functions receive high scores. We use a zero-one

functional similarity matrix with all pairs of proteins in the same specificity group receiving a one. We use the identity matrix as the amino acid similarity matrix, because we found that it works better than other similarity matrices (see Supplementary Analysis).

For *SDPpred*, the publicly accessible web server was used to score alignments. Source code for *Xdet* and *SH* was obtained from the authors. Default parameters were used for all methods. In our implementations of *RE* and *MI*, a pseudocount of one when estimating amino acid distributions was found to yield the best performance.

GroupSim

As a baseline for comparison, we implemented a simple method that considers all pairs of amino acids within and between groups. The average similarity between each pair of amino acids in a group is calculated according to a similarity matrix for each specificity group in the alignment. To reward difference between specificity groups, we compute for each group the average similarity (according to the matrix) of all amino acid pairs containing one amino acid in the group and one not in the group. This per group average is then averaged. The column score is the average within-group similarity minus the average between-group similarity. Higher scores indicate a greater likelihood of being a SDP. We tried a range of similarity matrices from the BLOSUM series [41], but as with *Xdet*, using the identity matrix provided the best results. A simple gap penalty, multiplying the column score by the fraction of non-gap positions in the column, is applied to the scores.

Conservation Window Heuristic

Positions important for determining specificity are often found near the active/interaction site. The residues in enzyme active sites are known to be more conserved than average [44]. If two columns have the same SDP score (according to any method), we might think that the one in the area of greater conservation is likely to be of greater importance for specificity. In order to test this idea, we developed a heuristic that incorporates the conservation of sequentially adjacent positions into the SDP score:

$$ConsWin(C) = \lambda SDP(C) + (1-\lambda)\frac{\sum_{i \in win} Cons(C_i)}{|win|} \tag{3.1}$$

where $SDP(C)$ is an SDP score for column C, win is a set containing the indices of all columns in a window around, but not including, column C. The second term is the average conservation of the window; we use the Jensen-Shannon divergence [5] to estimate conservation. We find $\lambda = 0.7$ and a window size of three residues on either side of C work well. Though the best parameters vary from

method to method, the results are robust across choices of λ. When discussing a method to which this heuristic has been applied, we will append "*+ConsWin*" to the method name.

3.3 Results

The size of the EC-Pfam data set allows us to describe properties of positions that are likely important for specificity. In addition, it enables the comparison of SDP prediction methods on a much larger scale than was possible previously.

3.3.1 Analysis of Positions Important for Specificity

In this section, we characterize a set of residues enriched with SDPs in terms of column amino acid pattern, secondary structure, relative solvent accessibility (RSA), and amino acid property differences observed between specificity groups in the same column. These observations should be useful in future SDP prediction method development.

Two SDP column patterns are over-represented near ligands.

Columns that exhibit amino acid conservation within specificity groups and difference between them have often been sought by SDP prediction methods. However, it has recently been argued that a lack of overlap in amino acid distribution between specificity groups is sufficient to indicate that a column is important for determining specificity [58].

Since there are too few experimentally-verified specificity determining positions to perform a reliable analysis of observed column patterns, we instead use our EC-Pfam data set to address the question. We assume that the set of residues within 5Å of the relevant ligand is enriched with specificity determining residues relative to alignment columns more than 5Å from the ligand. We then count the occurrence of each column pattern described in Table 3.1 in these two sets of positions. If we see significantly more columns of a given pattern near ligands, we attribute this difference to specificity-based pressures. Before performing this analysis, we removed all very conserved columns (Shannon entropy $\leq \frac{1}{3}$ bit) from each set, because these columns are not important for determining specificity and are over-represented near ligands (35.6% $\leq$5Å from ligand and 11.2% >5Å from ligand).

Table 3.2 compares the distribution of column patterns in positions near ligands to the distribution over positions not near ligands; each column is assigned to the strictest filter (pattern) it passes and significance is calculated using the hypergeometric distribution. Two column patterns,

one-group-conserved and all-groups-conserved, are significantly enriched in columns near ligands (P-values of 5.577e-8 and 8.814e-47 respectively). This likely reflects pressure from specificity-based constraints. In contrast, columns with the low-overlap pattern are significantly ($P = 0.012$) more common outside of regions likely important for specificity (8.8%) than in them (6.6%). This suggests that the low-overlap pattern alone is insufficient to indicate importance for specificity; a method that rewards this column pattern is likely to select columns that are far from relevant ligands. These results are consistent with a recent study of SDP in thirteen experimentally-characterized families [64] that found Type II (all-groups-conserved) and Type I (one-group-conserved) columns to be over-represented in SDPs as compared to non-SDPs.

Filter	$\leq$ 5Å from ligand	$>$ 5Å from ligand	P-value
Low-Overlap ($\mathcal{L}$)	0.066 (106)	**0.088** (1550)	0.012
One-Group-Conserved ($\mathcal{O}$)	**0.174** (278)	0.125 (2196)	5.577e-8
All-Groups-Conserved ($\mathcal{A}$)	**0.132** (211)	0.034 (669)	8.814e-47

Table 3.2: Enrichment of column amino acid patterns near ligands. Each row gives the fraction of positions $\leq$ and $>$ 5Å from ligands having the given pattern. The raw count of each pattern is given in parenthesis. Conserved positions were removed prior to the enrichment analysis, and each position is counted only for the most specific filter it passes. P-values were calculated from the hypergeometric distribution. Positions passing the one-group-conserved and all-groups-conserved filters are significantly enriched near ligands. Significant enrichment is shown in bold.

Based on this enrichment, we use the $SDP_{\mathcal{O}}$ set—all columns within 5Å of a relevant ligand passing at least the one-conserved-filter—as positives in the following analysis and method evaluation and refer to positions in this set as "putative SDPs".

The amino acid distribution of putative SDPs is more polar than the background.

Catalytic sites are known to have an amino acid distribution with more charged residues than the background distribution [44]. The amino acid distribution of putative SDPs is also quite different from the background distribution observed in the alignments (χ^2 test P-value = 4e-4 using the distribution in all positions as the expected distribution). Table 3.3 gives these distributions relative to a partition of the amino acids into charged (H, R, K, E, D), non-charged polar (Q, T, S, N, C, Y, W), and all others.

In contrast to catalytic sites, the percentage of charged residues in putative SDPs is similar to the background. However, putative SDPs exhibit more non-charged polar residues than either

	Charged AA	Non-charged Polar AA	Other AA
All Positions	0.24	0.24	0.52
Catalytic Sites	0.66	0.25	0.09
Putative SDPs	0.24	0.31	0.45

Table 3.3: Comparison of amino acid distributions. Putative SDPs are more likely to be a non-charged polar residue than a residue chosen at random. Catalytic sites do not exhibit this bias; instead they are more charged.

catalytic sites or the background. This suggests that the sites that determine specificity are rarely involved in catalytic processes such as proton exchange, and are more likely to take part in the weak non-covalent bonds that often mediate the interactions between enzymes and small molecules.

Putative SDPs are most likely to be found in loop regions.

Table 3.4 shows that the secondary structure distribution of columns likely important for specificity in the EC-Pfam data set is also quite different from the background distribution observed in the alignments. Putative SDPs are significantly more likely to be found in loops, i.e., not in alpha helices or beta sheets, than would be expected by chance (χ^2 test P-value = 3.44e-12 using the distribution in all positions as the expected distribution). Catalytic sites have a similar distribution. This suggests that considering secondary structure predictions could help identify SDPs, but unlike amino acid distribution, might not help distinguish between SDPs and catalytic sites.

	Alpha Helix	Beta Sheet	Loop
All Positions	0.41	0.22	0.37
Catalytic Sites	0.28	0.22	0.50
Putative SDPs	0.27	0.21	0.52

Table 3.4: Comparison of secondary structure distributions. Putative SDPs are much more likely to be in loop regions than would be expected by chance. Catalytic sites show a similar secondary structure bias.

The relative solvent accessibility profile of putative SDPs is different from that of all residues.

The distribution of observed relative solvent accessibilities is markedly different between putative SDPs and all residues in the data set. Figure 3.1 provides a histogram of the relative solvent

accessibilities. Compared to all residues, SDPs are less likely to be extremely buried (30% in the 0-5% RSA range compared to 36% for all positions) or extremely exposed (only 12% at RSA≥40% compared to 23% for all). However, the percentage of SDPs with 5-40% RSA is significantly greater for putative SDP than for all columns. The majority of putative SDPs have relatively low solvent exposure. A similar pattern was observed for catalytic sites [44], and similar forces may explain this somewhat counter-intuitive result. SDPs often require precise positioning and are likely found in large clefts on the protein that are important for binding substrates.

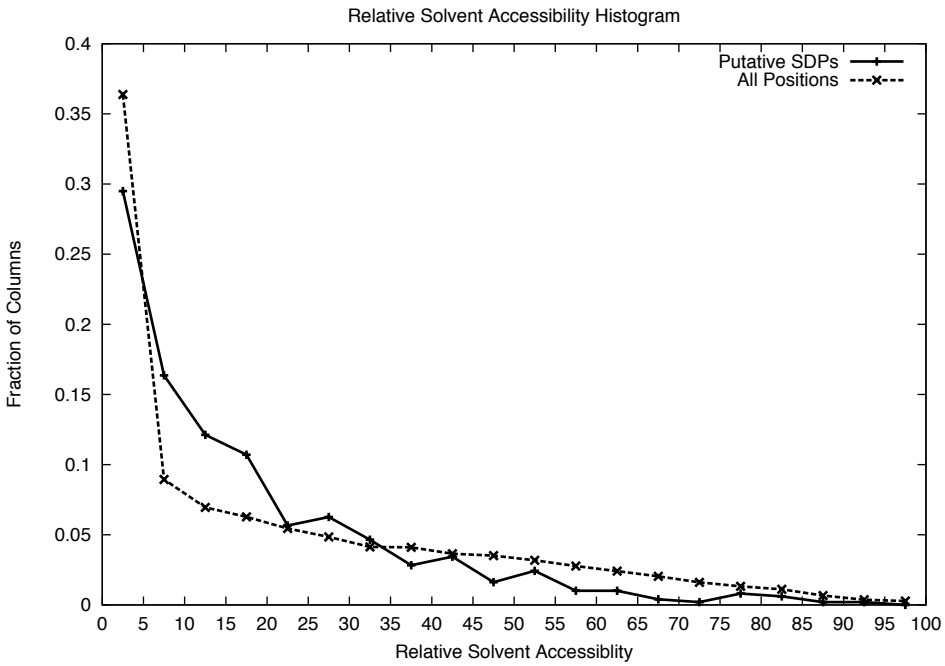

Figure 3.1: Comparison of the relative solvent accessibility (RSA) of putative SDPs ($SDP_{\mathcal{O}}$) to all positions. The RSA range is divided into 20 equal size bins, e.g., the first bin corresponds to RSA between 0 and 5%. Each point represents the fraction of the columns with RSA falling in the bin.

Some amino acid physiochemical properties are less conserved between specificity groups in putative SDPs than expected.

Since SDPs distinguish between possible ligands, they often exhibit conservation of certain amino acid properties within specificity groups and difference—relative to those properties—between them. To identify what properties may be used to distinguish ligands, we analyzed the amino acid changes observed between specificity groups in putative SDPs.

Each row in Table 3.5 represents a partition of the amino acids that corresponds to a property which nature could use to distinguish between ligands. If the property is relevant, we would expect to see amino acid differences that are not conservative, relative to the property partition, between groups in putative SDPs. For each partition, the fraction of all amino acid pairs across specificity groups in putative SDPs that do not conserve the property is reported. The "All Positions" column gives the percentage of non-conservative pairs relative to each property partition over all alignment positions and serves as the background reference point for each partition and the significance calculation.

	Different Between Groups	
Amino Acid Partition	Putative SDPs	All Positions
Polarity	0.656	0.418
Size	0.642	0.450
Hydrophobicity	0.376	0.279
Charge	0.369	0.274

Table 3.5: Average fraction of non-conservative (relative to each partition) amino acid differences between specificity groups by position type . Each row gives the fraction of all amino acid pairs between specificity groups that differ under the given amino acid property partition. All properties are significantly less conserved between specificity groups in putative SDPs than over all positions.

Polarity, size, hydrophobicity, and charge are all significantly less conserved between groups in putative SDPs than in the background. The binomial P-values for the observed differences are infinitesimal. The difference is largest for polarity and size. This suggests that these residue properties may commonly be used to establish different specificity in similar proteins.

3.3.2 SDP Prediction Method Evaluation

In this section we evaluate a representative set of recent methods—*relative entropy (RE)*, *mutual information (MI)*, *SDPpred*, *Sequence Harmony (SH)*, and *Xdet*—against our simple method, *GroupSim*.

GroupSim performs competitively with existing methods.

The performance of each method is judged via two complementary techniques. Figure 3.2 gives box plots for each method, and Figure 3.3 shows their PR curves. In general, PR analysis rewards accuracy in the first few predictions whereas the average rank analysis rewards performance equally across all positives. All results reported here are over $SDP_{\mathcal{O}}$, but our main conclusions are not

sensitive to the filters used to select the positives. See the Supplementary Analysis for results on $SDP_{\mathcal{L}}$ and $SDP_{\mathcal{A}}$.

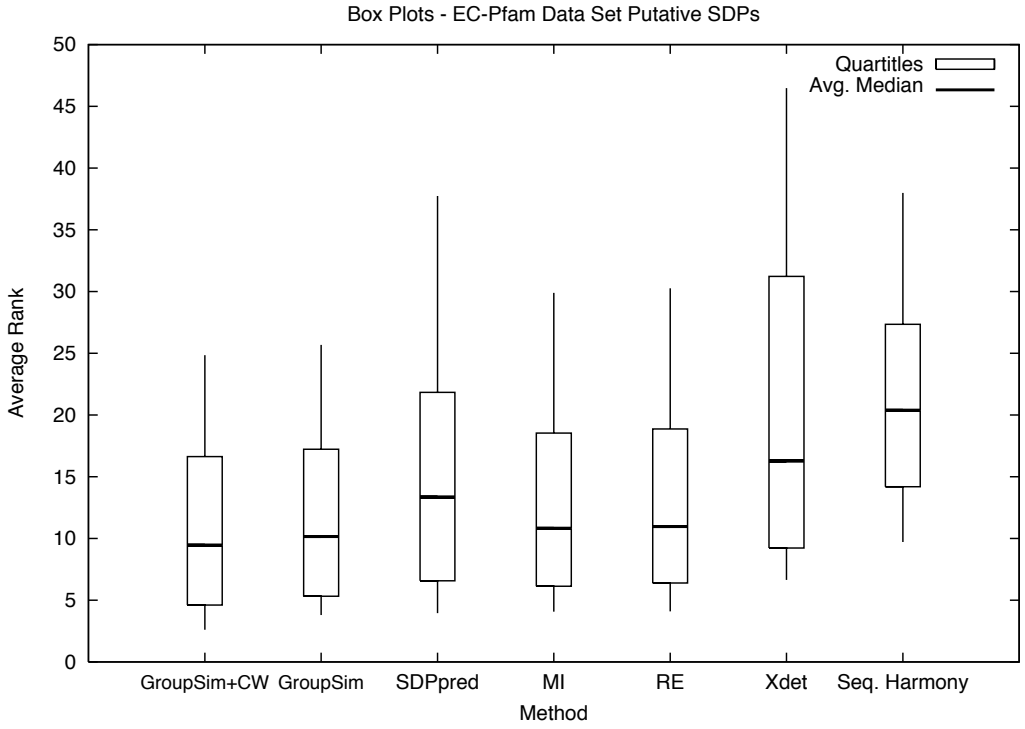

Figure 3.2: Box plots for the SDP prediction methods on the putative SDPs in the EC-Pfam data set ordered by average minimum. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. The simple *GroupSim* outperforms the previous methods in this evaluation, and *GroupSim+ConsWin* improves on it.

The box plots shown in Figure 3.2 demonstrate that when considering the ranks of SDPs, *GroupSim* gives lower average minimum, median, quartiles, and maximum than existing methods. For example, the average rank over all alignments of the first positive found is 3.8 for *GroupSim* and 9.7 for *SH*. Similarly, the low average maximum of *GroupSim* implies that, compared to other methods, it gives fewer positives very poor scores. In PR analysis (Figure 3.3), *GroupSim*'s AUC (.368) is competitive with *MI* (.377) and *RE* (.369), and markedly better than *Xdet* (.328) and *SH* (.243). Only *SDPpred* has a much greater AUC (.400).

The results in these evaluations suggest that none of *GroupSim*, *RE*, *MI* and *SDPpred* clearly performs best in predicting SDPs in all contexts, but that they perform better than *Xdet* and *SH*. *GroupSim* gives the best performance in the average rank analysis while *SDPpred* achieves the highest PR-AUC. Since the PR-AUC focuses on accuracy on the first few positives, this indicates that if a few SDPs are sought *SDPpred* might be better, while if all are sought, *GroupSim* could be better.

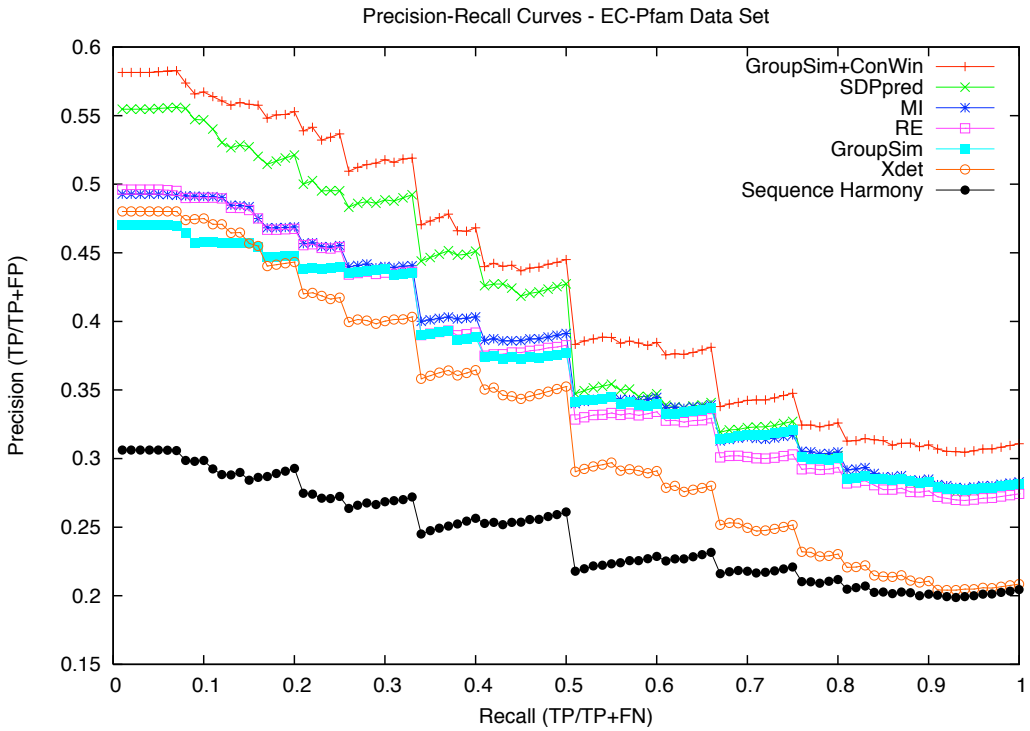

Figure 3.3: Precision-Recall curves for representative SDP prediction methods on the putative SDPs from the EC-Pfam data set. The simple *GroupSim* is competitive with the other methods; *SDPpred* is the only method that substantially outperforms it. *GroupSim+ConsWin* outperforms all methods.

The conservation window heuristic significantly improves method performance.

Figures 3.2 and 3.3 include the *GroupSim+ConsWin* method, which is our basic *GroupSim* method along with a heuristic that incorporates the conservation of neighboring amino acids. This heuristic provides significant improvement over *GroupSim* (P-value 4.2e-7 using Friedman test on PR-AUC) and outperforms all other methods in terms of AUC (.428) and average ranks.

ConsWin can be applied to any SDP scoring scheme that produces scores for each column of an alignment. In addition to *GroupSim*, *ConsWin* provides improvement over the raw version of other methods evaluated. Figures 3.4 and 3.5 illustrate the boost provided by *ConsWin* to *RE*, *Xdet*, and *SH*. Results are similar to those observed with *GroupSim* (Figures 3.2 and 3.3) for all methods except *SDPpred*. The range of *SDPpred* scores for an alignment is often quite large and variable as a result of a few outlier column scores. This requires setting the window λ parameter specially for each alignment.

GroupSim+ConsWin was shown to outperform all current methods tested on the EC-Pfam data set. When *ConsWin* is added to other current methods, *SDPpred+ConsWin*, *MI+ConsWin* and *RE+ConsWin* become competitive with, but not better than, *GroupSim+ConsWin*.

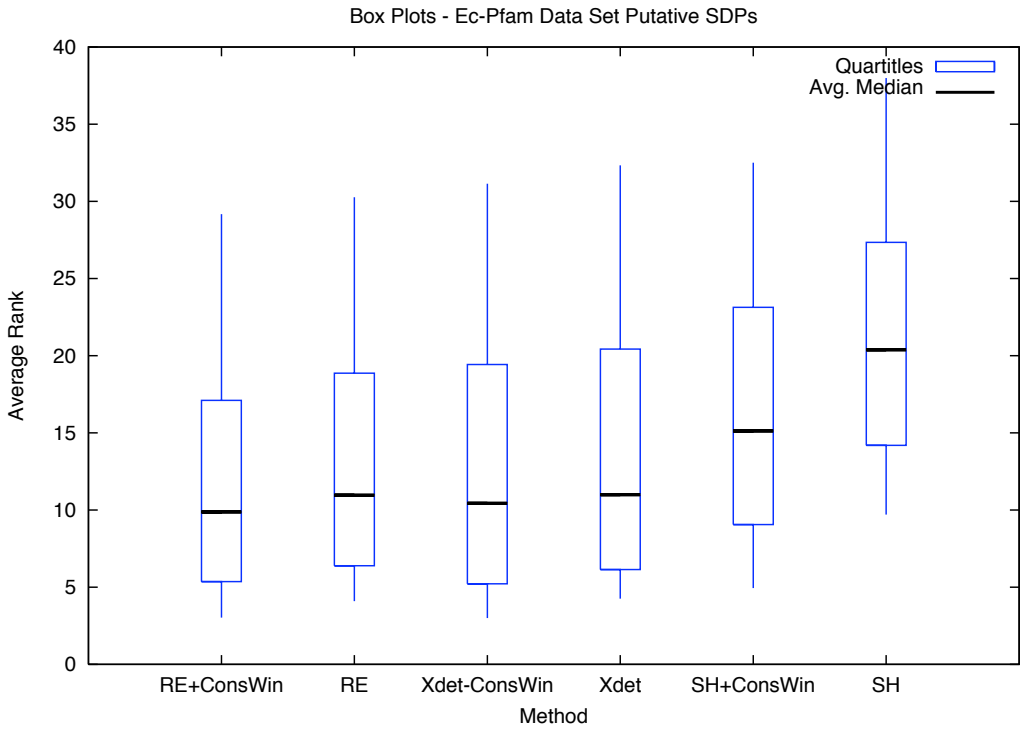

Figure 3.4: Box plots comparing *Xdet*, *SH*, and *RE* with and without the *ConsWin* heuristic on the set of putative SDPs, $SDP_{\mathcal{O}}$. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. The *ConsWin* heuristic improves all methods.

ConsWin works well on our enzyme data set, because residues in enzyme active sites are significantly more conserved than the background. Thus, the assumption that SDPs are near ligands may boost the performance of *ConsWin* on our enzyme data set; however, this assumption is supported in the literature, and is commonly made in small-scale SDP studies. Overall, we believe that the heuristic will be useful in a variety of contexts, but that the conservation signal may not be as strong for all types of interaction. We find that *ConsWin* improves predictions of *GroupSim* on five out of eight non-enzyme families in the experimentally-determined data set of Chakrabarti et al. [64], and 10 out of 13 families overall (Table 3.6). See the next section for these results and more discussion.

Results on an experimentally-derived data set are similar to those on the EC-Pfam data set.

An experimentally-derived data set of 13 alignments was recently described in Chakrabarti et al. [64]. While we believe that the small size of their data set limits its utility in method evaluation, we find that our overall conclusions are similar when this data set is used. Figures 3.6 and 3.7 provide box plots and PR curves for the representative SDP prediction methods on the experimentally-

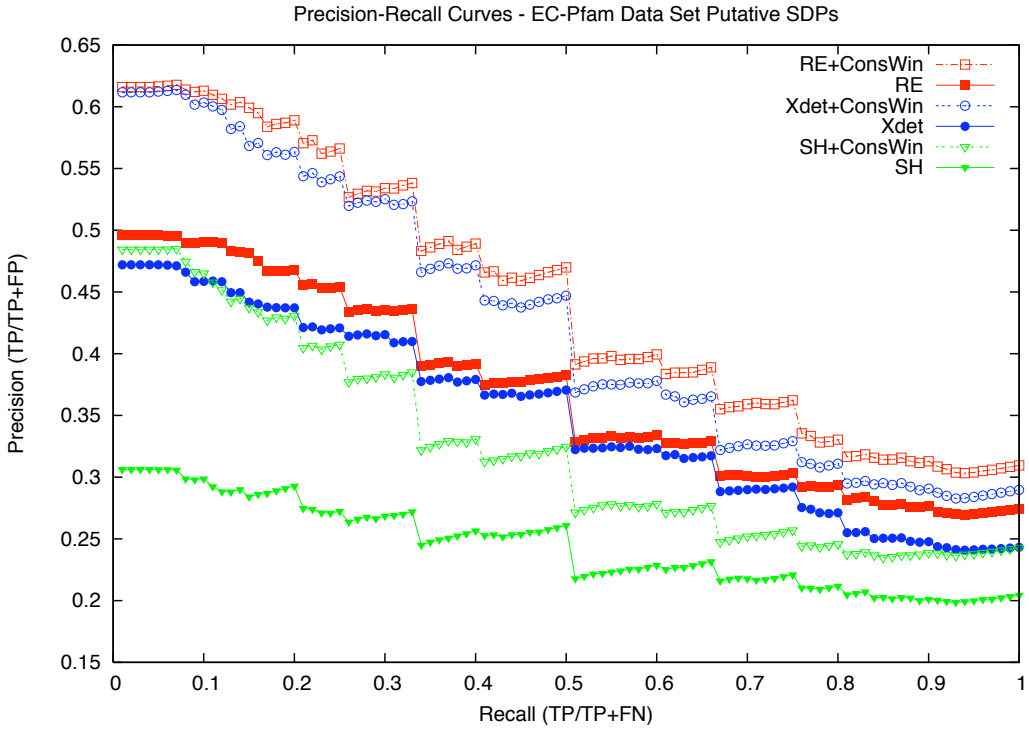

Figure 3.5: Precision-Recall curves comparing *Xdet*, *SH*, and *RE* with and without the *ConsWin* heuristic on the set of putative SDPs, $SDP_{\mathcal{O}}$. The *ConsWin* heuristic improves the performance of all methods tested.

dervided SDP data set. Table 3.6 provides more detail on the composition of this data set and the performance of the *ConsWin* heuristic.

In general, the difference between methods is less pronounced than in our results, but the main trends are similar. *GroupSim* provides competitive performance with all methods tested, and *GroupSim+ConsWin* provides improvement over *GroupSim* in 10 of the 13 alignments (Table 1). However, it is difficult to say that one method is superior to the others on this data set.

As noted earlier, our analysis of SDP amino acid column patterns is in agreement with the column pattern distribution of the experimentally-determined SDPs.

The protein composition of the experimental data set (Table 3.6) is somewhat different from our large data set of enzymes. The experimental data set contains only five enzymes, and four of the alignments contain more than two specificity groups. These differences in composition explain some of the variance in method performance relative to the EC-Pfam data set.

For example, the most notable contrast is the poor performance of *MI* relative to *RE* and *SDPpred*. *MI*'s change in performance is entirely the result of very poor performance on alignments with more than two specificity groups. However, we find that *MI* plus the shuffling normalization is

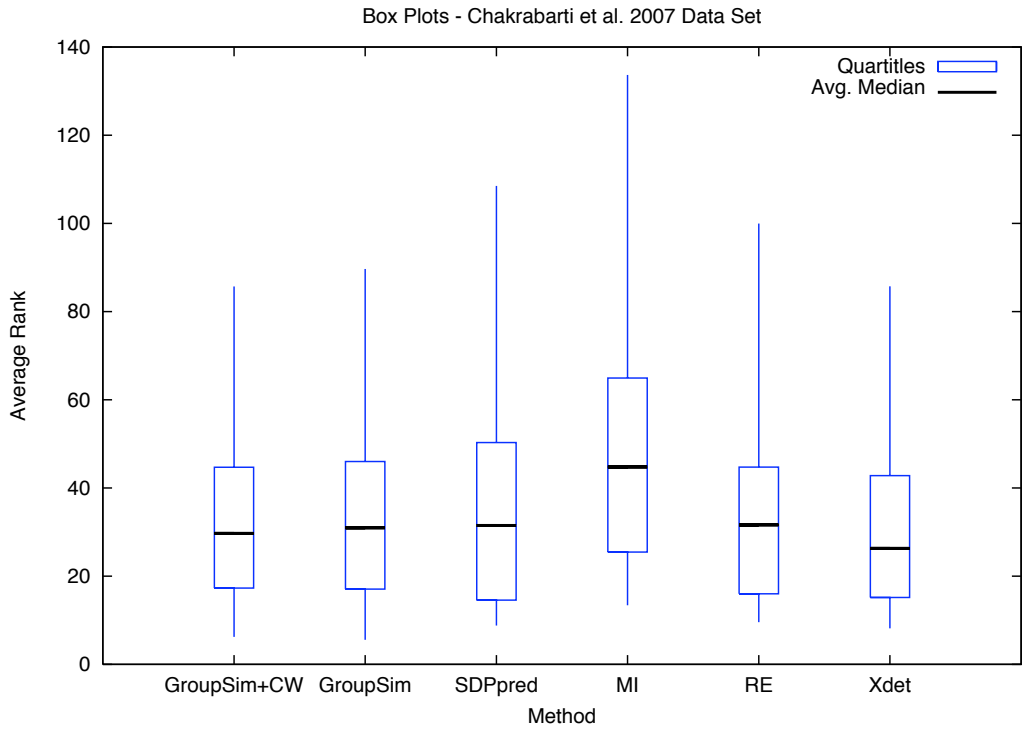

Figure 3.6: Box plots for representative SDP prediction methods and two versions of *GroupSim* on the experimentally-derived data set of [64]. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. The simple *GroupSim* outperforms the previous methods, and *GroupSim+ConsWin* slightly improves on it.

competitive with *SDPpred*, further suggesting the importance of shuffling to the *SDPpred* method.

We were unable to evaluate *Sequence Harmony* on this data set because it cannot handle alignments with more than two groups.

Our data set provides a platform for better understanding prediction methods.

We now give three examples of how our data set can be used to analyze performance tradeoffs between aspects of SDP prediction methods. First, our evaluation reveals that *SDPpred* performs better than *MI* in the PR evaluation. This is interesting, because the only difference between the two is the column shuffling significance procedure applied by *SDPpred* to adjust the *MI* score. We find that shuffling provides similar PR-AUC improvement for *RE* and *GroupSim*, but does not improve the average ranks (see Supplementary Analysis for data supporting this result and others discussed in this paragraph). Second, it is surprising how sensitive *MI* and *RE* are to the magnitude of the pseudocount used; for example, a pseudocount of 1e-6 results in a PR-AUC of .259 for *MI* compared to .377 obtained using 1. Third, we observe that *GroupSim*'s performance is stable with respect

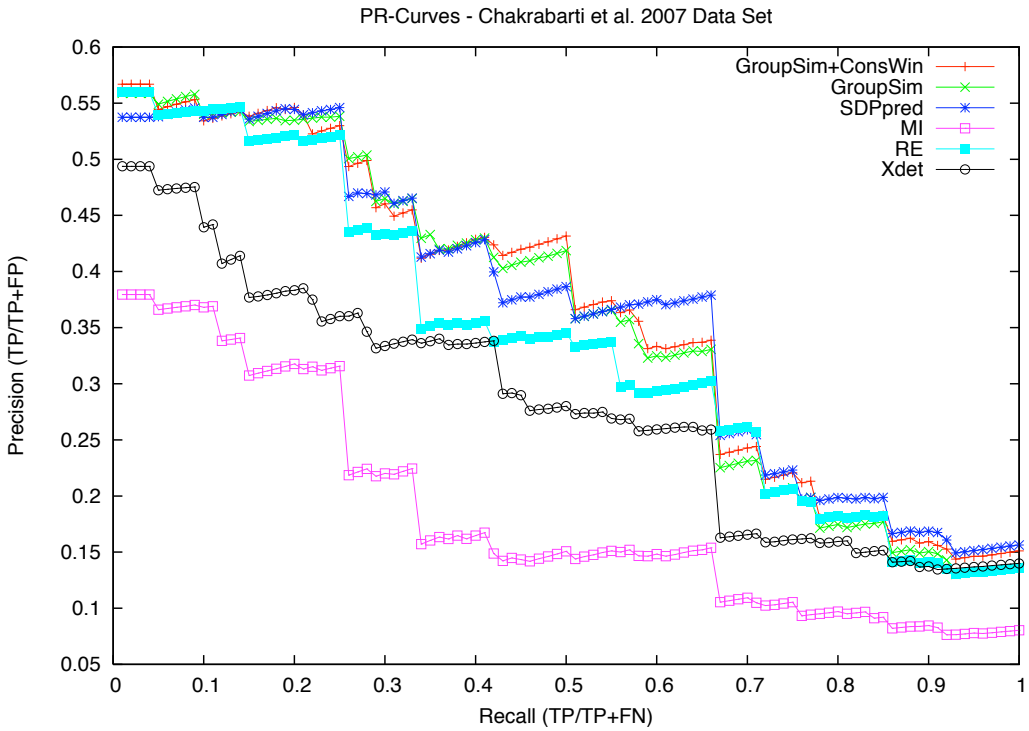

Figure 3.7: Precision-Recall curves for representative SDP prediction methods and two versions of *GroupSim* on the experimentally-derived data set of [64]. The simple *GroupSim* is competitive with the existing method. *GroupSim+ConsWin* provides improvement for 10 of the 13 alignments.

to subgroup sequence diversity; it performs slightly better than *SDPpred* on alignments with very diverse subgroups (data not shown). These observations illustrate the utility of our data set in evaluating design decisions made in SDP prediction methods.

3.4 Discussion and Conclusion

The lack of experimentally identified SDPs with supporting MSAs has impaired the development and evaluation of computational methods for predicting SDPs. We address this problem by automating an approach that researchers often carry out by hand to identify putative SDPs in the absence of mutation studies. The generated data set contains several hundred alignments of enzymes with putative SDPs identified, and has enabled us to characterize properties of SDPs and evaluate the performance of MSA-based SDP prediction methods. Our main findings on this data set hold as well on the diverse set of 13 families with experimentally-determined SDPs of Chakrabarti et al. [64]. Our large data set allows us to better compare methods, and the consistency of our results on the smaller data set lends support to our methodology.

Family	Enzyme	# Specificity Groups	*GroupSim+ConsWin*	*GroupSim*	Difference
Gprotein		11	0.737	0.698	0.038
cd00333		2	0.533	0.517	0.016
cbm9	Y	2	0.175	0.163	0.012
cd00365	Y	2	0.143	0.132	0.011
GST	Y	11	0.737	0.727	0.010
ricin		3	0.219	0.211	0.008
cd00423	Y	2	0.475	0.470	0.005
cd00985		2	0.724	0.720	0.004
cd00363	Y	2	0.020	0.019	0.001
cd00264		2	0.107	0.106	0.001
cd00120		2	0.192	0.197	-0.005
CNmyc		2	0.060	0.066	-0.006
LacI		15	0.625	0.654	-0.029

Table 3.6: Summary of *GroupSim*'s performance on an experimentally-derived data set of Chakrabarti et al. [64]. The last three columns compare the PR-AUC performance of *GroupSim* and *GroupSim+ConsWin*. *ConsWin* provides improvement for 10 out of 13 families overall and 5 out of 8 non-enzyme families.

In our analysis, we find that putative SDPs are quite different from average protein residues in terms of amino acid distribution, secondary structure, and solvent accessibility. Our data suggest that SDPs are often found in environments similar to catalytic sites, but that SDPs' amino acid distributions contain many fewer charged residues and more non-charged polar residues. This suggests that, in enzymes, SDPs are more likely to be involved in the recognition and binding of the substrate than in the catalytic mechanism. We also find evidence that amino acid polarity, hydrophobicity, size, and charge are used to distinguish between similar ligands.

Analysis of our data set suggests that columns in which at least one specificity group is conserved and different from the others are significantly over-represented in regions likely to contain SDPs. This does not imply that columns with other amino acid patterns are never important for specificity, but merely that such patterns more often occur in regions that are unlikely to directly influence interactions with the ligand.

The comprehensive data set and evaluation presented here provide a foundation upon which further progress in predicting SDPs can be built. Improved identification of SDPs will aid protein engineering, pathway analysis, and function prediction. The recent work of George et al. [79] using known catalytic sites to transfer annotations could likely be extended to include SDPs to attain even more specific function predictions. However, the observation that most current SDP prediction methods perform similarly to a simple method, *GroupSim*, suggests that there is much room for improvement. This improvement may come from integrating knowledge about properties of SDPs

into the development of sequence-based methods; for example, here we show that by exploiting the conservation signal from neighboring amino acids, *GroupSim+ConsWin* outperforms all earlier methods tested on our data set of enzyme SDPs. The new SPEER method [64] provides another step in this direction, and the recent work by Fischer et al. [80] on predicting functional residues may provide a framework for integration. Ultimately, improved understanding of the properties and mechanisms of SDPs, via experimental work as well as large-scale analysis and evaluation like we present here, should lead to improved SDP prediction.

3.5 Supplementary Analysis

Results on $SDP_{\mathcal{L}}$ and $SDP_{\mathcal{A}}$ are similar to those on $SDP_{\mathcal{O}}$.

Section 3.1.1 of the main text provides significant evidence that the definition of putative SDPs as the set $SDP_{\mathcal{O}}$ is reasonable. This set includes all columns within 5Å of ligands relevant to the enzyme's catalytic reaction that contain at least one specificity group with a conserved amino acid distribution that is significantly different from that of the other specificity groups in the column. In the main text, all methods are evaluated on this set of positives.

However, it is possible to perform evaluations of the SDP prediction methods using different sets of columns as the positives. Our other column filters (see Table 1 in the main text) provide several alternatives. The $SDP_{\mathcal{L}}$ set provides a less strict definition of positive that includes any columns near ligands in which the specificity group amino acid distributions do not have significant overlap. Figures 3.8 and 3.9 show the box plots and PR-curves for this positive set.

Similarly, we can evaluate on the stricter set of positions provided by the $SDP_{\mathcal{A}}$ set. All these columns are within 5Å of a relevant ligand and each specificity group's amino acid distribution is conserved and different from the others. Figures 3.10 and 3.11 show the box plots and PR-curves for this positive set.

Our general conclusions from the main text—that *GroupSim* is competitive with current methods and that *GroupSim+ConsWin* is the best performing method—hold on both these alternative positive sets. Comparing the results across sets demonstrates that in general method performance improves as the positive set becomes stricter; they do best on $SDP_{\mathcal{A}}$ and worst on $SDP_{\mathcal{L}}$.

One exception to this is the *Sequence Harmony (SH)* method. Its performance relative to other methods on $SDP_{\mathcal{A}}$ and itself on other positive sets is far worse on this strict set. This is in part because *SH* does not explicitly reward conservation within specificity groups and all columns in this

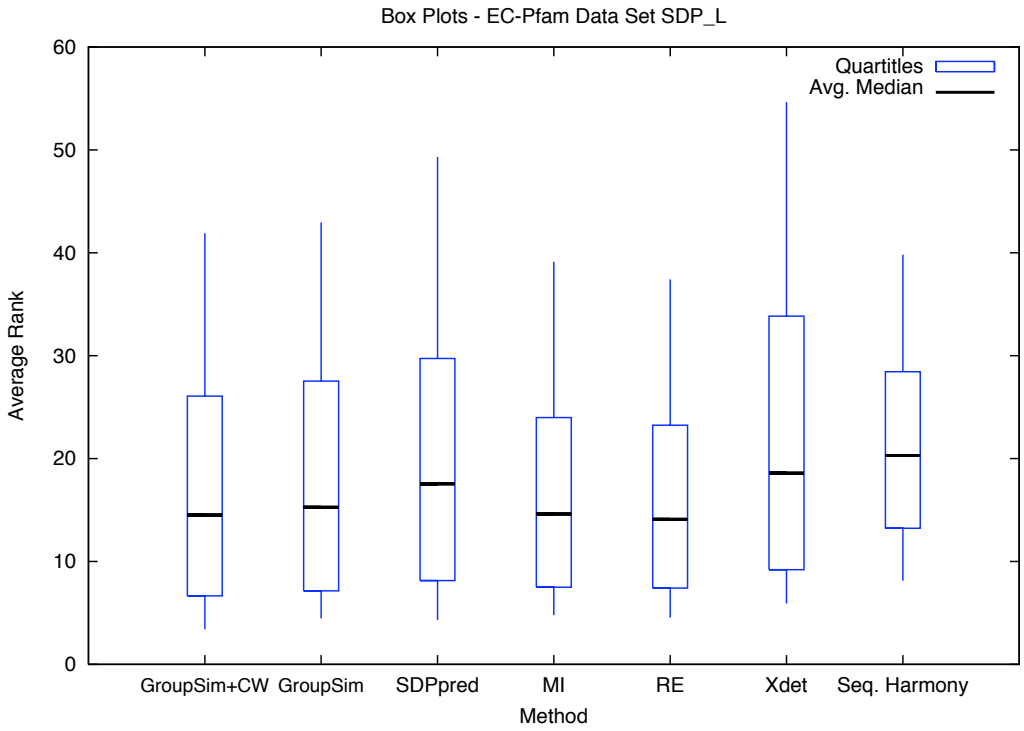

Figure 3.8: Box plots for the SDP prediction methods on $SDP_{\mathcal{L}}$ ordered by average minimum. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. The simple *GroupSim* performs similarly to previous methods, and *GroupSim+ConsWin* improves on it.

positive set are required to have this property. However, *SH*'s handling of gaps, overlap between columns, and tie scores also harm its performance (data not shown).

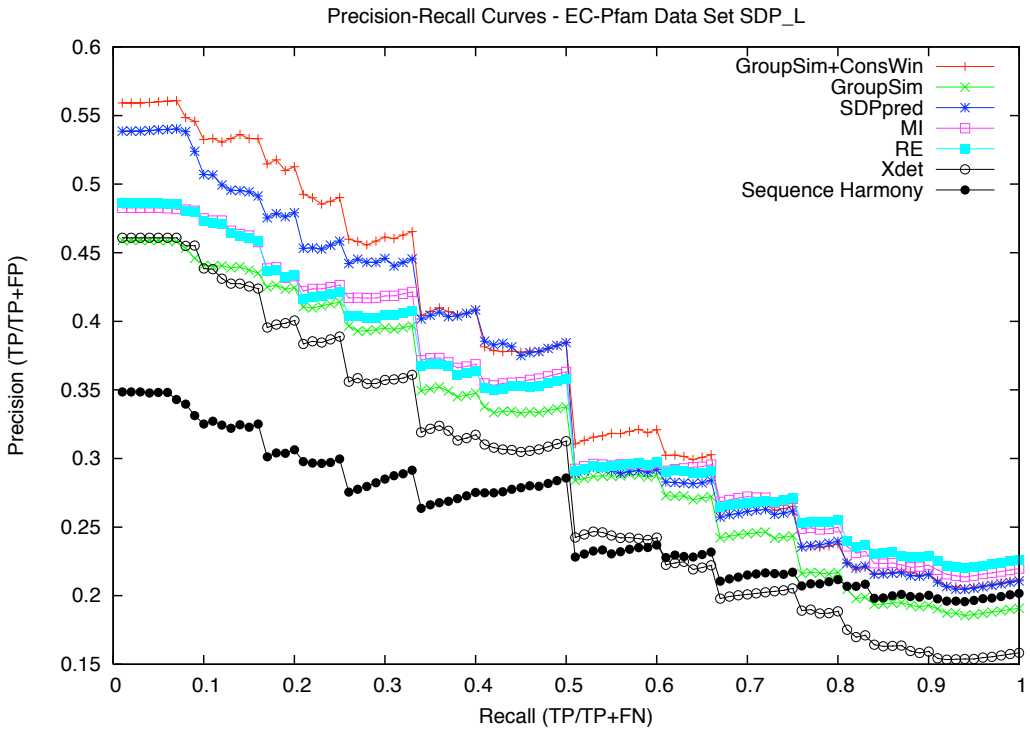

Figure 3.9: Precision-Recall curves for representative SDP prediction methods and two versions of *GroupSim* on $SDP_{\mathcal{L}}$. The simple *GroupSim* is competitive with the other methods; *SDPpred* is the only method that significantly outperforms it. *GroupSim+ConsWin* outperforms all methods.

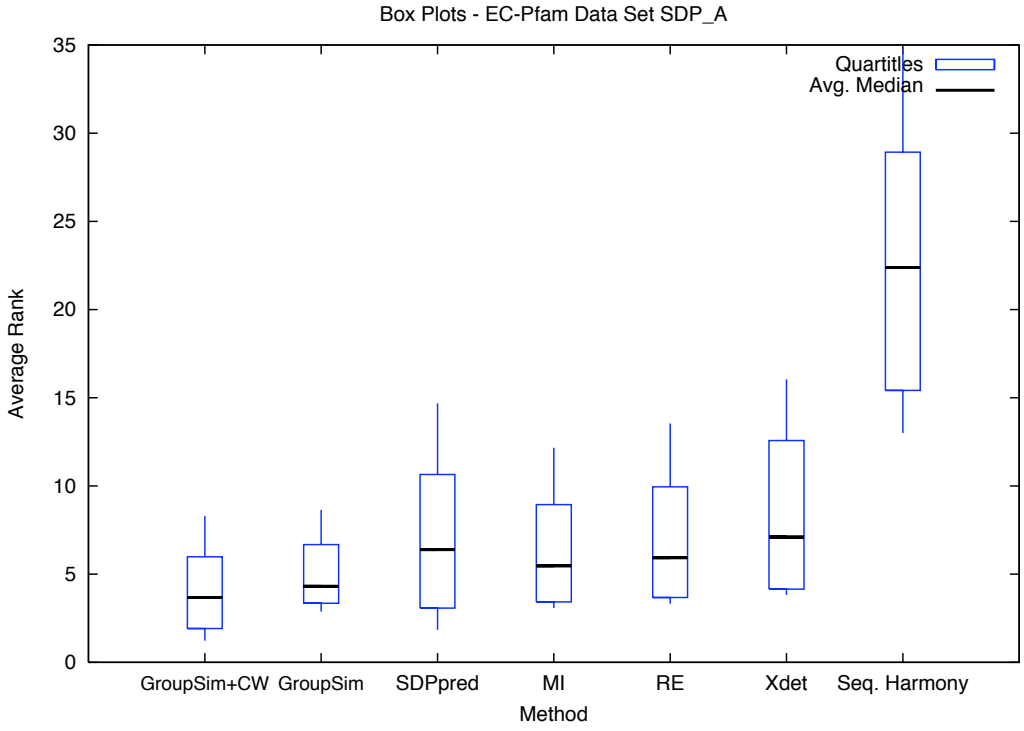

Figure 3.10: Box plots for the SDP prediction methods on $SDP_{\mathcal{A}}$. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. The simple *GroupSim* performs similarly to previous methods, and *GroupSim+ConsWin* provides the best results.

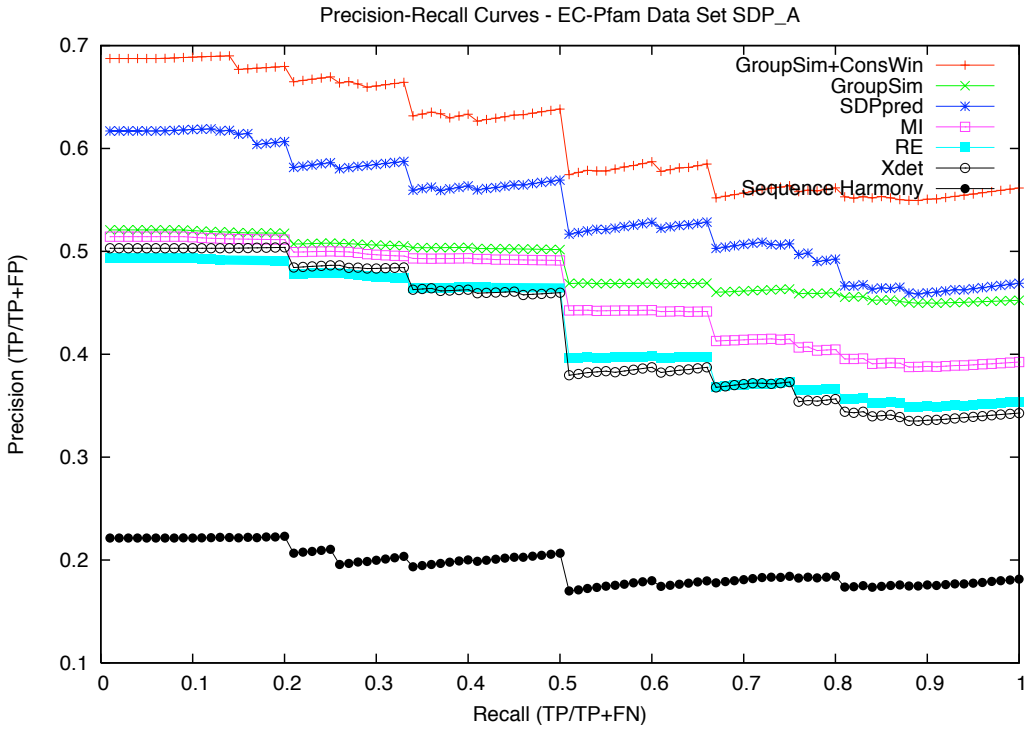

Figure 3.11: Precision-Recall curves for representative SDP prediction methods and two versions of *GroupSim* on $SDP_{\mathcal{A}}$. The simple *GroupSim* is competitive with the other methods; SDPpred is the only method that significantly outperforms it. *GroupSim+ConsWin* significantly outperforms all methods.

MI and *RE* are sensitive to pseudocount used.

Mutual information (MI) and *relative entropy (RE)* both require the estimation of an amino acid probability distribution from a multiple sequence alignment column. A small, uniform pseudocount is commonly added to the observed counts in this distribution estimation step. We have found that the choice of pseudocount is important to the performance of *MI* and *RE*. Figures 3.12 and 3.13 show box plots for *MI* and *RE* across a range of pseudocount values.

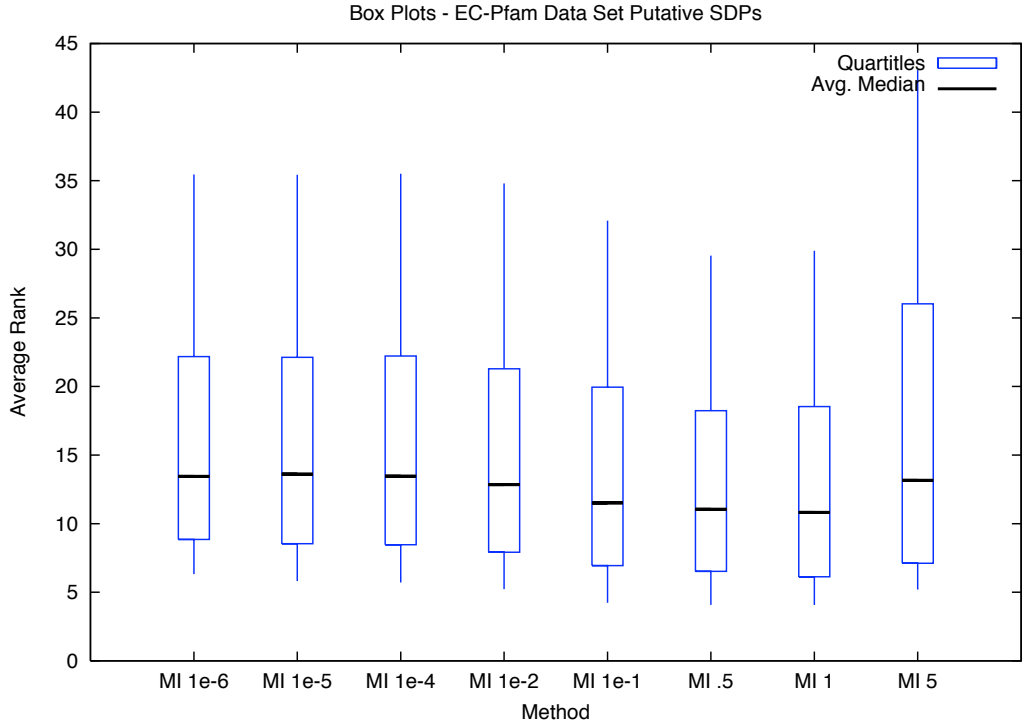

Figure 3.12: *Mutual information*'s performance on the set of putative SDPs ($SDP_{\mathcal{O}}$) is sensitive to the pseudocount used. The number following "MI" gives the magnitude of the uniform pseudocount. A pseudocount of one gives the best performance.

There is a clear trend that favors larger pseudocounts all the way up to a pseudocount of one for this application. We also implemented more complex schemes for computing non-uniform pseudocounts [32], but found that the magnitude dominated the effect on performance. We use a pseudocount of one for both methods in all other analysis presented here and in the main text. This dependence on pseudocount magnitude is likely due to the sparsity of the distributions being compared in this context.

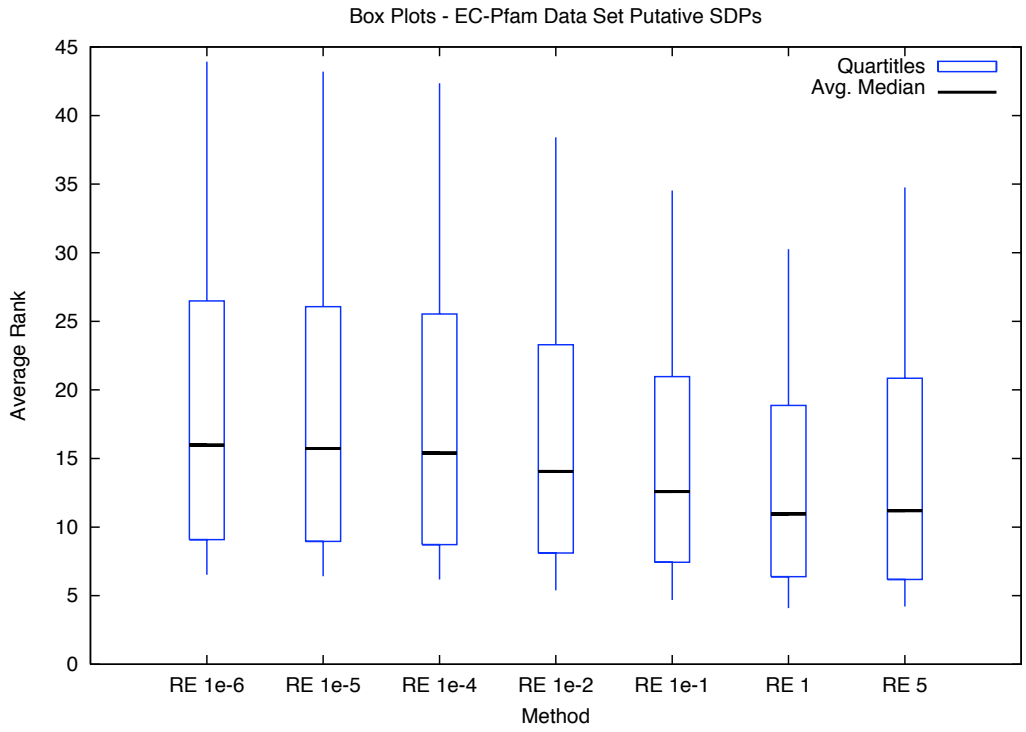

Figure 3.13: *Relative entropy*'s performance on the set of putative SDPs ($SDP_{\mathcal{O}}$) is sensitive to the pseudocount used. The number following "RE" gives the magnitude of the uniform pseudocount. A pseudocount of one gives the best performance.

Use of Similarity Matrices with *Xdet* and *GroupSim* hurts performance.

Several methods for predicting SDPs from multiple sequence alignments (MSAs), allow the incorporation of the relationships between amino acids through the use of a similarity matrix. In our evaluation, we considered two such methods, *Xdet* [62] and our own *GroupSim*. Figures 3.14, 3.15, 3.16, and 3.17 provide box plots and PR-curves for *Xdet* and *GroupSim* with a range of similarity matrices.

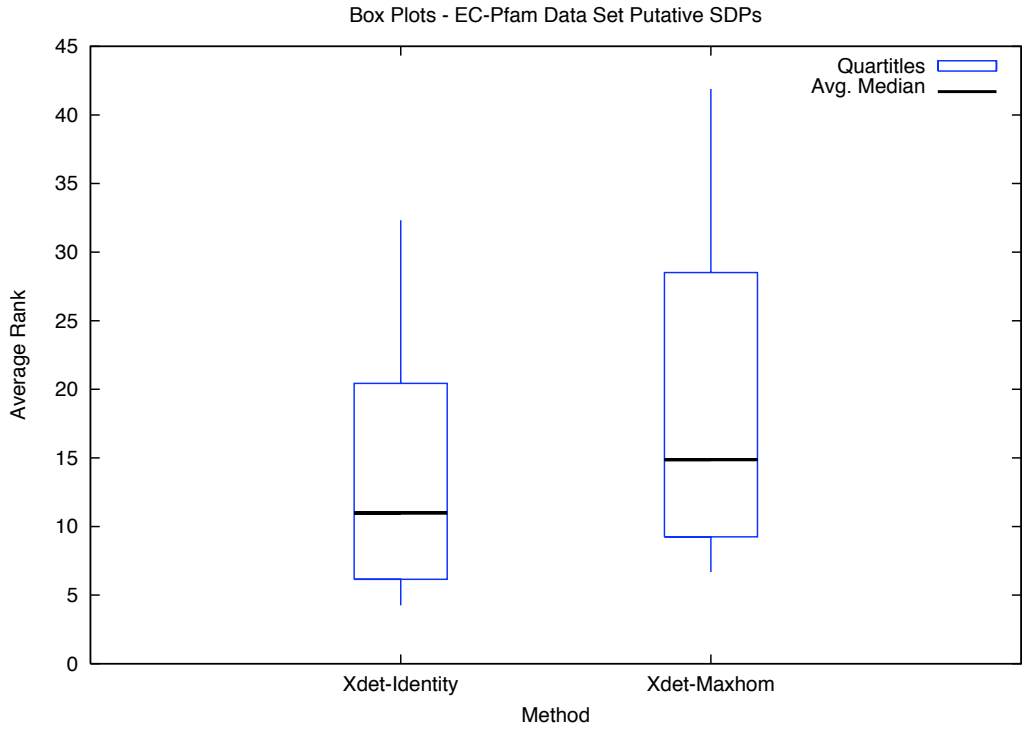

Figure 3.14: Box plots for Xdet using the default McLachlan matrix [81] and using the identity matrix on the putative SDPs ($SDP_{\mathcal{O}}$). Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. Surprisingly *Xdet* with the identity matrix outperforms the use of a similarity matrix.

Both methods obtain the best performance when the identity matrix is used. The identity matrix is used with both methods in all other results presented here and in the main text. This surprising result is similar to an observation we made previously when evaluating methods for estimating amino acid conservation [5].

It seems that the matrix weights dominate the more relevant conservation and difference signals. We are currently investigating techniques for directly incorporating notions of amino acid similarity into SDP prediction methods.

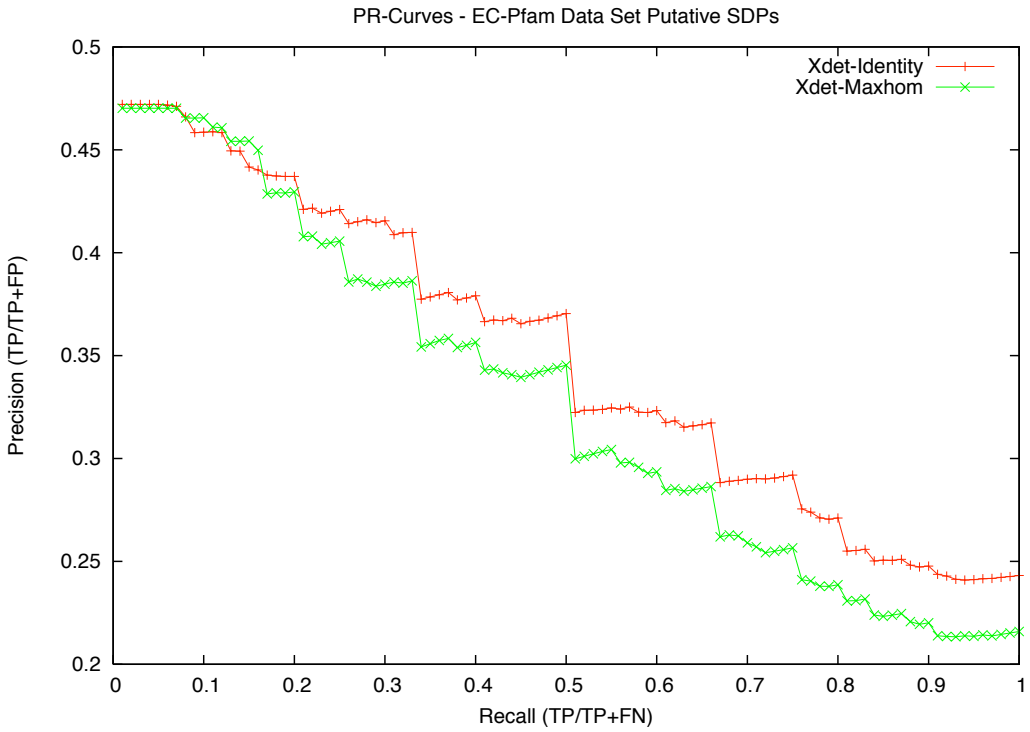

Figure 3.15: Precision-Recall curves for *Xdet* [62] using the default matrix, McLachlan, and using the identity matrix on the putative SDPs ($SDP_{\mathcal{O}}$). *Xdet* with the identity matrix is superior in PR analysis as well.

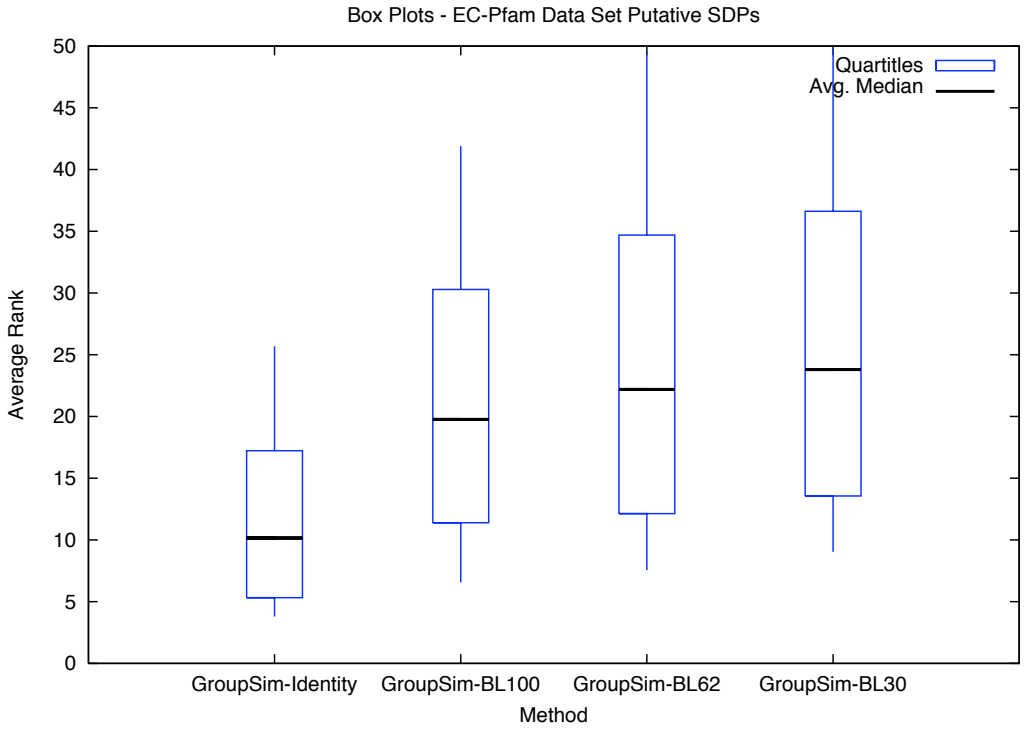

Figure 3.16: Box plots for *GroupSim* using a range of BLOSUM [41] matrices and the identity matrix on the putative SDPs ($SDP_{\mathcal{O}}$). Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. Surprisingly *GroupSim* with the identity matrix outperforms the direct use of a similarity matrices.

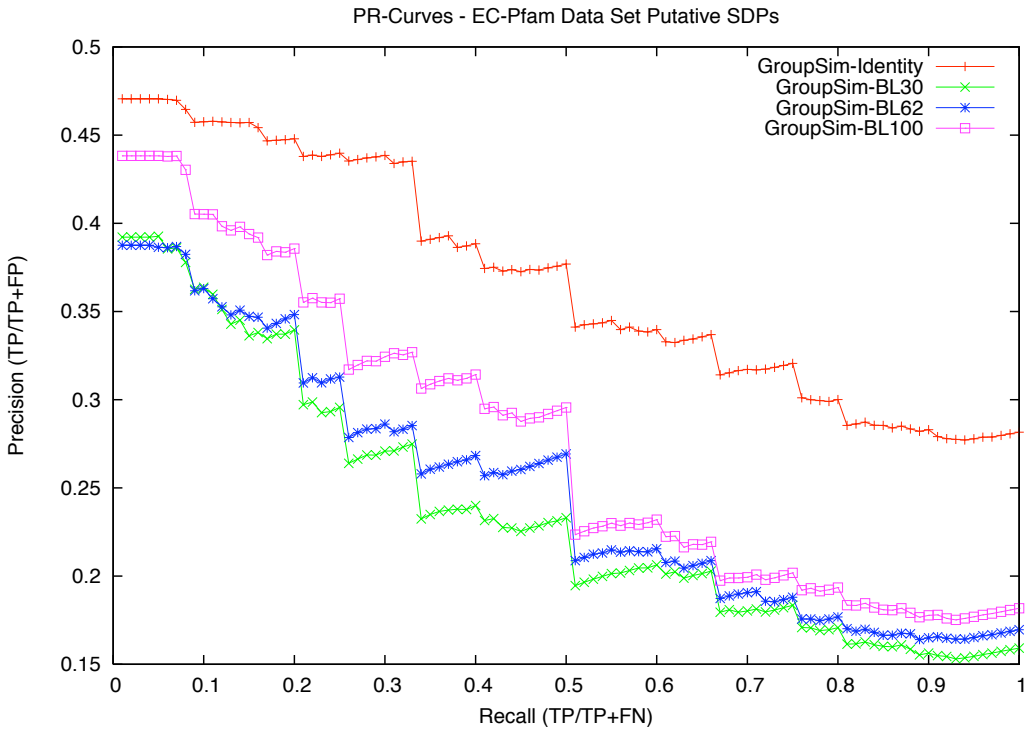

Figure 3.17: Precision-Recall curves for *GroupSim* using a range of BLOSUM [41] matrices and the identity matrix on the putative SDPs ($SDP_{\mathcal{O}}$). *GroupSim* with the identity matrix is superior in this PR analysis as well.

Column Shuffling provides some improvement to methods.

SDPpred [17] incorporates a shuffling procedure into its *mutual information (MI)* based score. As we noted in the main text, this shuffling appears to provide a performance boost over *MI* in the Precision-Recall evaluation, but not in the box plot evaluation. Figures 3.18 and 3.19 give box plots and PR-curves for several methods with and without a column shuffling normalization. The results are similar for *RE* and *GroupSim*. The PR curves show consistent improvement, but the box plots show that the improvement is not consistent across the range of rank statistics. This provides evidence that shuffling may improve some predictions, but overall all predictions do not become more accurate. These results are based on 1000 iterations of the shuffling procedure of Kalinina et al. [17] without their linear transformation.

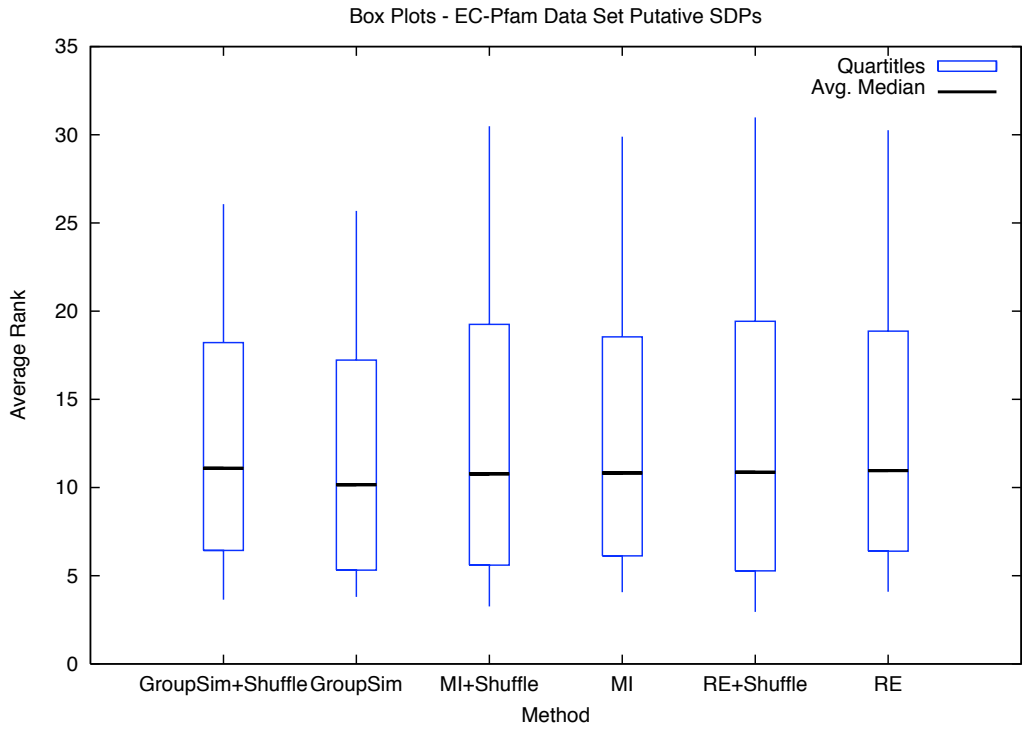

Figure 3.18: Box plots comparing *GroupSim*, *RE*, and *MI* with and without a column shuffling normalization on the set of putative SDPs, $SDP_{\mathcal{O}}$. Each box shows the average over all alignments of the five-number summary (the minimum, lower quartile, median, upper quartile, and maximum) for a method. Lower averages indicate better performance. Shuffling provides some improvement, but it is not consistent across statistics.

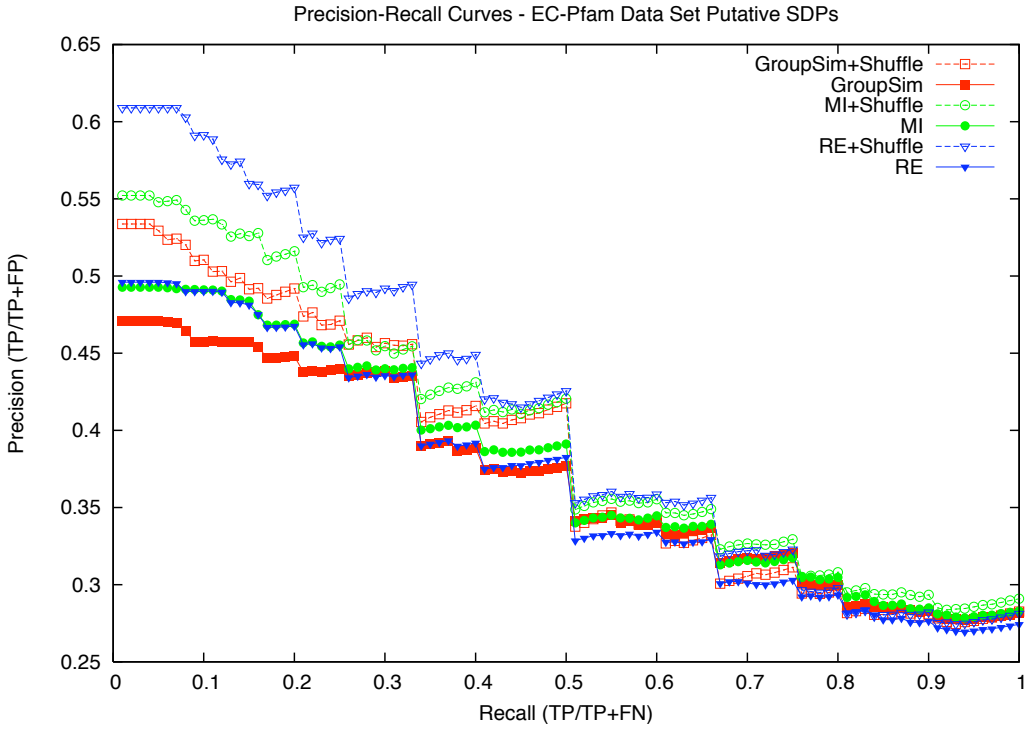

Figure 3.19: Precision-Recall curves comparing *GroupSim*, *RE*, and *MI* with and without a column shuffling normalization on the set of putative SDPs, $SDP_{\mathcal{O}}$. Shuffling provides improvement for the methods in this evaluation.

Chapter 4

Predicting Protein Ligand Binding Sites by Combining Evolutionary Sequence Conservation and 3D Structure

4.1 Introduction

Proteins' functions are determined to a large degree by their interactions with other molecules. Identifying which residues participate in these interactions is an important component of functionally characterizing a protein. Current experimental efforts to identify functional sites and determine protein function cannot keep pace with the rapid rate at which known sequences and structures are being deduced. To date, there are nearly 1000 sequenced genomes with several thousand more in progress [1, 2], and over 50,000 3D structures in the Protein Data Bank [3, 4]. As a result, computational methods play an increasingly important role in characterizing protein function. Many approaches, based on analysis of protein sequences or structures, have been developed to predict a variety of protein functional sites, including ligand binding sites [5, 82, 83], DNA-binding sites [84], catalytic sites [5, 85], protein-protein interaction interfaces (PPIs) [86, 87] and specificity determining positions [6, 17, 88].

In this chapter, we focus on the task of predicting small molecule binding sites from protein

sequences and structures. In addition to aiding in the functional characterization of proteins, knowledge of these binding sites can guide the design of inhibitors and antagonists and provide a scaffold for targeted mutations. Over the past 15 years, a large number of methods for predicting ligand binding sites have been developed. Structural approaches have used geometric and energetic criteria to find concave regions on the protein surface that likely bind ligands [82, 89–97]. Sequence-based approaches, on the other hand, have largely exploited sequence conservation, or the tendency of functionally or structurally important sites to accept fewer mutations relative to the rest of the protein [30].

Here, we introduce *ConCavity*, a new approach for predicting 3D ligand binding pockets and individual ligand binding residues. The *ConCavity* algorithm directly integrates evolutionary sequence conservation estimates with structure-based surface pocket prediction in a modular three step pipeline. In the first step, values are assigned to a regular grid of points surrounding the protein surface. These grid scores are a combination of the output of a structure-based pocket finding algorithm and the sequence conservation of nearby residues in a multiple sequence alignment of homologs. We perform this integration for three different pocket identification algorithms: *Ligsite* [92], *Surfnet* [90], and *PocketFinder* [98]. Next, we extract coherent pockets from the grid using 3D shape analysis algorithms to ensure that the predicted pockets have biologically reasonable shapes and volumes. In the final step, we map from the predicted pockets to nearby residues by assigning high scores to residues near high scoring pocket grid points. The *ConCavity* algorithm yields predictions of regions in space that are likely to contain ligand atoms and of protein residues likely to contact bound ligands.

We evaluate *ConCavity* and a representative set of methods that consider evolutionary conservation [5] or structural attributes [82, 90, 95, 98, 99] on the diverse, non-redundant LigASite database of biologically relevant binding sites [100]. As in previous evaluations, we test the ability of methods to predict pockets, as represented by regions in space or protein surface residues, that contact bound ligands. However, we designed our methods to make more specific predictions, e.g., which residues in a pocket bind the ligand and which spatial regions of a pocket are most likely to contain ligand atoms. We also evaluate the methods' performance on these more difficult tasks.

Though previous methods have used sequence or structural data in predicting ligand binding sites, the two approaches have rarely been directly compared and evaluated, and even the basic question of whether sequence or structure is more informative for a given prediction task has not been adequately considered. We find that the structural approaches outperform conservation, and *ConCavity*, which combines conservation and structure, provides significant improvement over both

types of existing algorithms in ligand binding pocket and residue predictions. *ConCavity* achieves maximum precision of greater than 76% when predicting ligand binding residues, while previous methods never obtain precision greater than 67%. These results suggest that there is significant added benefit to considering structural information when it is available. We also show that several methodological improvements in pocket extraction and residue mapping give our implementations of existing methods a significant gain in performance over the previous versions.

Finally, we perform a detailed analysis of our predictions that reveals much about the relationship between sequence conservation, structure, and function. Consistent with earlier findings, there is a drop in the performance of all methods that consider structure when predicting on apo (unbound) structures as compared to holo (bound) structures. Nevertheless, *ConCavity* still performs very well, and the ranking of methods does not change. *ConCavity* is also found to provide significant improvement over previous methods in the context of predicting drug binding sites. Since ligand binding sites are not the only type of functional site of interest, we also evaluate the methods on the prediction of catalytic sites. As with ligand binding site prediction, *ConCavity* performs the best of all methods. However, sequence conservation outperforms the tested structure-based methods in this context, and overall the methods have much more difficulty identifying catalytic sites than ligand binding sites. The contrasting performance of structure and sequence methods in these scenarios emphasizes the need to consider the context as well as the type of functional site sought when developing and applying these approaches.

Overall, sequence conservation and structure-based attributes provide complementary information about functional importance. For example, ligand binding site prediction methods based on structural information alone are often unable to distinguish binding pockets from non-binding inter-chain cavities. However, the residues in ligand binding sites usually exhibit greater evolutionary conservation than those comprising the non-binding cavities. *ConCavity* takes advantage of such complementarity to improve on previous approaches.

Related Work

Sequence-based functional site prediction has been dominated by the search for residue positions that show evidence of evolutionary constraint. Amino acid conservation in the columns of a multiple sequence alignment of homologs is the most common source of such estimates (see [30] for a review). Recent approaches that compare alignment column amino acid distributions to a background amino acid distribution outperform many existing conservation measures [5, 31]. However, the success of conservation-based prediction varies based on the type of functional residue sought; sequence

conservation has been shown to be strongly correlated with ligand binding and catalytic sites, but less so with residues in protein-protein interfaces (PPIs) [5]. A variety of techniques have been used to incorporate phylogenetic information into sequence-based functional site prediction, e.g., traversing phylogenetic trees [60, 101], statistical rate inference [36], analysis of functional subfamilies [6, 88], and phylogenetic motifs [102]. Recently, evolutionary conservation has been combined with other properties predicted from sequence, e.g. secondary structure and relative solvent accessibility, to identify functional sites [80].

Structure-based methods for functional site prediction seek to identify protein surface regions favorable for interactions. Ligand binding pockets and residues have been a major focus of these methods [82, 89–97]. *Ligsite* [92] and *Surfnet* [90] identify pockets by seeking points near the protein surface that are surrounded in most directions by the protein. *CASTp* [93, 95] applies alpha shape theory from computational geometry to detect and measure cavities. In contrast to these geometric approaches, other methods use models of energetics to identify potential binding sites [25, 98, 99, 103]. Recent algorithms have focused on van der Waals energetics to create grid potential maps around the surface of the protein. *PocketFinder* [98] uses an aliphatic carbon as the probe, and *Q-SiteFinder* [99] uses a methyl group. Our work builds upon geometry and energetics based approaches to ligand binding pocket prediction, but it should be noted that there are other structure-based approaches that do no fit in these categories (e.g., Theoretical Microscopic Titration Curves (THEMATICS) [104], binding site similarity [105], phage display libraries [106], and residue interaction graphs [107]). In contrast to sequence-based predictions, some structure-based methods are able to make predictions both at the level of residues and regions in space that are likely to contain ligands.

Several previous binding site prediction algorithms have considered both sequence and structure. ConSurf [19] provides a visualization of sequence conservation values on the surface of a protein structure, but it does not make explicit binding site predictions. Spatially clustered residues with high Evolutionary Trace values were found to overlap with functional sites [108], and Panchenko *et al.* [18] found that averaging sequence conservation across spatially clustered positions provides improvement in functional site identification in certain settings. Several groups have attempted to identify and separate structural and functional constraints on residues [109, 110]. Wang *et al.* [111] perform logistic regression on three sequence-based properties and an estimate of the effect on structural stability of mutations at each position to find functional sites. Though these approaches make use of protein structures, they do not explicitly consider the surface geometry of the protein in prediction. Geometric, chemical, and evolutionary criteria have been used together to define motifs that

represent known binding sites for use in protein function prediction [112]. Machine learning algorithms have been applied to features based on sequence and structure [113, 114] to predict catalytic sites [26, 29, 85, 115] and recently to predict drug targets [116] and a limited set of ligand and ion binding sites [117–119]. Sequence conservation has been found to be a dominant predictor in these contexts. Two recent approaches to ligand binding site identification have used evolutionary conservation in a post-processing step to rerank [82] or refine [120] geometry based pocket predictions. In contrast, *ConCavity* integrates conservation directly into the search for pockets. This allows it to identify pockets that are not found when considering structure alone, and enables straightforward analysis of the relationship between sequence conservation, structural patterns, and functional importance.

4.2 Results

In this section, we compare the ability of *ConCavity*, which combines evolutionary sequence conservation with structure-based pocket prediction, and a representative set of previous methods to identify ligand binding sites.

For simplicity, we use *Ligsite*$^+$, our implementation (as indicated by superscript "+") of a popular geometry based surface pocket identification algorithm, as representative of the performance of structure-based methods, and refer to it as "*Structure*". (We demonstrate in the Methods section that *Ligsite*$^+$ provides a fair representation of these methods.) The *Jensen-Shannon divergence* (*JSD*) has recently been shown to provide state-of-the-art functional site identification performance among sequence conservation estimation algorithms [5]. We use *JSD* to represent these approaches and refer to it as "*Conservation*". We have developed three versions of *ConCavity* that integrate evolutionary conservation into different surface pocket prediction algorithms (*Ligsite* [92], *Surfnet* [90], or *PocketFinder* [98]). When the underlying algorithm is relevant, we refer to these versions as *ConCavity*$_L$, *ConCavity*$_S$, and *ConCavity*$_P$. However, for simplicity, we will use *ConCavity*$_L$ as representative of these approaches and call it "*ConCavity*".

ConCavity and *Struture* produce predictions of ligand binding pockets and residues. The pocket predictions are given as non-zero values on a regular 3D grid that surrounds the protein; the score associated with each grid point represents an estimated likelihood that it overlaps a bound ligand atom. Similarly, each residue in the protein sequence is assigned a score that represents its likelihood of contacting a bound ligand. *Conservation* only makes residue-level predictions, because it does not consider protein structure. All methods are evaluated on the 331 proteins of the non-redundant

LigASite 4.0 dataset. When evaluating pocket predictions, the ligands found in the holo version of the dataset are considered positives; when evaluating residue predictions, the residues annotated as ligand binding in the apo version of the dataset are used.

We now present our main findings.

Integrating evolutionary sequence conservation and structure-based pocket finding in ligand-binding site prediction improves on either approach alone.

We quantify the overall performance of each method's predictions in two ways. For both pocket and residue prediction, we generate Precision-Recall (PR) curves that reflect the ability of each method's grid and residue scores to identify ligand atoms and ligand binding residues, respectively. (Just as residues are assigned a range of ligand binding scores, grid points in predicted pockets get a range scores, since there may be more evidence that a ligand is bound in one part of a pocket than another.) These curves are given in Figure 4.1. In addition, for predicted pockets (non-zero values in the 3D grid), it is natural to consider how well they overlap known ligands. However, we also must consider the size of the predicted pockets, because it is easy to cover more of the ligand by predicting larger pockets. The Jaccard coefficient captures this tradeoff between precision and recall by taking the ratio of the intersection of the predicted pocket and ligand volumes over their union. It ranges between zero and one, and high values imply that the predictions cover the ligands well and have volumes similar to the ligand volumes. Pocket and ligand volumes, Jaccard coefficients, and related statistics are presented in Table 4.1.

Method	Fraction with Ligand Overlap	Prediction Vol. (Å^3)	Ligand Vol. (Å^3)	Prediction ∩ Ligand (Å^3)	Prediction ∪ Ligand (Å^3)	Jaccard coefficient
Structure	0.92	1167.3	1164.5	333.6	2080.5	0.179
Concavity	0.95	1185.8	1164.5	425.3	1993.8	0.255

Table 4.1: Comparison of the overlap between pockets predicted by each method and bound ligands in holo protein structures from the LigASite database. The first column gives the fraction of proteins for which a method's predictions overlap a ligand. The second column (Prediction Vol.) lists the median volume of the predicted pockets for each protein, while the third column (Ligand Vol.) lists the median volume of ligands observed in the PQS file. The next columns give the median volumes of the Intersection and Union of the predictions and ligands and the Jaccard coefficient (Intersection / Union). *ConCavity* and *Structure* predict pockets of similar sizes—both use a similar pocket threshold—but *ConCavity*'s predictions overlap more of the bound ligands. The higher Jaccard coefficient for *ConCavity* implies that it better manages the tradeoff between precision and recall.

The PR curves presented in Figure 4.1 provide support for our *ConCavity* approach of predicting binding sites by considering structural attributes weighted by sequence conservation. The curves in

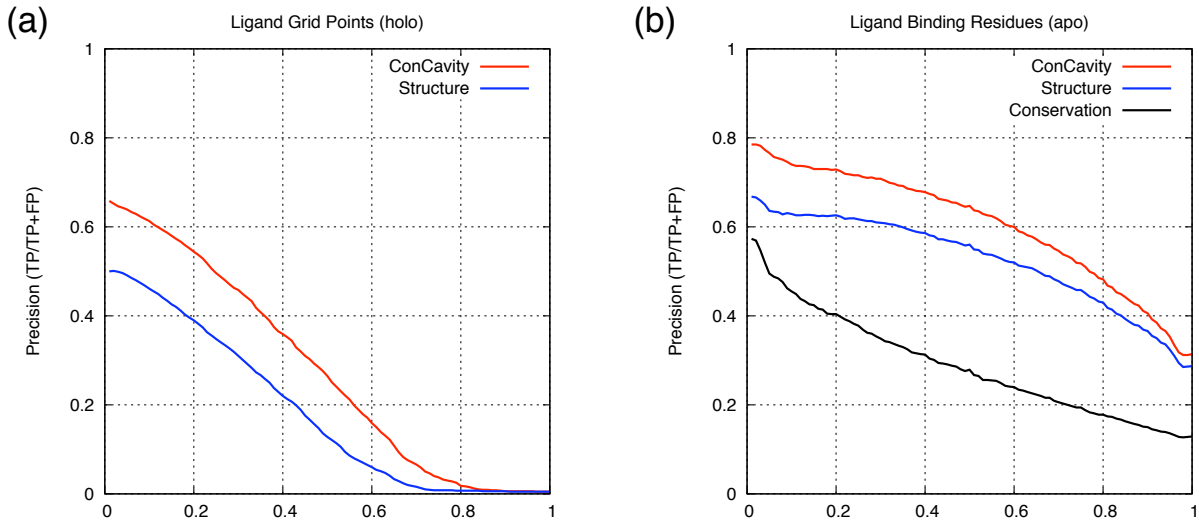

Figure 4.1: Ligand Binding Site Prediction Performance. **(a)** PR curves for prediction of the spatial location of biologically relevant bound ligands. **(b)** PR curves for ligand binding residue prediction. Our *ConCavity* algorithm, which combines sequence conservation with structure-based predictors, significantly outperforms other methods at both tasks. The approach based on structural information alone outperforms considering sequence conservation alone. Comparing the two panes, we see that accurately predicting the location of all ligand atoms is harder for the methods than finding all the contacting residues. *Conservation* does not make predictions of this type, so could not be included in (a). The curves are based on binding sites in the 331 proteins of the non-redundant LigASite 4.0 dataset.

Figure 4.1a show that, within predicted pockets, grid points with higher scores are more likely to overlap the ligand, and that the improvement of *ConCavity* over *Structure* exists across the range of score thresholds. Figure 4.1b demonstrates that the superior performance of *ConCavity* holds when predicting ligand binding residues as well as ligand binding pockets. *ConCavity*'s ability to identify ligand binding residues is striking; across this diverse dataset, the first residue prediction of *ConCavity* will be in contact with a ligand in nearly 80% of proteins. *ConCavity* also maintains high precision across the full recall range: precision of 65% at 50% recall and better than 30% when all ligand-binding residues have been identified. As mentioned above, this large improvement exists when predicting ligand locations as well; however, the PR curves illustrate that fully identifying a ligand's position is more difficult for each of the methods than finding contacting residues.

The ligand overlap statistics presented in Table 4.1 also demonstrate the superior performance of *ConCavity*. In nearly 95% of structures, *ConCavity*'s predictions overlap with a bound ligand. *Structure*'s predictions overlap ligands in nearly 92% of the proteins considered. The differences between the methods become even more stark when we examine the magnitude of these overlaps. Both *ConCavity* and *Structure* predict pockets with total volume (Prediction Vol.) similar to that

of all relevant ligands (Ligand Vol.), but *ConCavity*'s pockets overlap a significantly larger fraction of the ligand volume. Thus *ConCavity* has a higher Jaccard coefficient. This suggests that the integration of sequence conservation with structural pocket identification results in more accurate pockets than when using structural features alone.

Figure 4.1b also provides one of the first direct comparisons of ligand binding site prediction methods based on sequence conservation with those based on structural features. *Structure* outperforms *Conservation*, a state-of-the-art method for estimating sequence conservation. Protein residues can be evolutionarily conserved for a number of reasons, so it is not surprising that *Conservation* identifies many non-ligand-binding residues, and thus, does not perform as well as *Structure*. Considering the conservation of sequentially adjacent residues provides some improvement to *Conservation*, but beyond the very low recall region, it is not competitive with *Structure* (data not shown).

***ConCavity*'s improvement comes from integrating complementary information from evolutionary sequence conservation and structure-based pocket identification.**

Figure 4.2 presents pocket and residue predictions of *Conservation*, *Structure*, and *ConCavity* on three example proteins. In general, different types of positions are predicted by *Conservation* and *Structure*—only about 25% of their top predicted residues overlap (e.g. Figure 4.2b), where the number of residues considered for each structure is the number of known ligand binding residues. The residues predicted by sequence conservation are spread throughout the protein; ligand-binding residues are often very conserved, but many other positions are highly conserved as well due to other functional constraints. In contrast, the structure-based predictions are strongly clustered around surface pockets, and many of these residues near pockets are not evolutionarily conserved. However, these features provide largely complementary information about importance for ligand binding. Nearly 70% of residues predicted by both *Conservation* and *Structure* are in contact with ligands, while only 16% and 43% of those predicted using only conservation or structure respectively are ligand binding. *ConCavity* takes advantage of this complementarity to achieve its dramatic improvement; it gives high scores to positions that show evidence of both being in a well-formed pocket and being evolutionarily conserved.

The examples of Figure 4.2 illustrate this and highlight several common patterns in *ConCavity*'s improved predictions. For 1FK4, a small lipid transfer protein (Figure 4.2a), *Structure*'s residue predictions center on the main ligand binding pocket, while *Conservation* gives high scores to some positions in the binding site, but also some unrelated residues. Looking at the ligand location predictions, *Structure* and *ConCavity* both find the pocket, but the signal from conservation enables

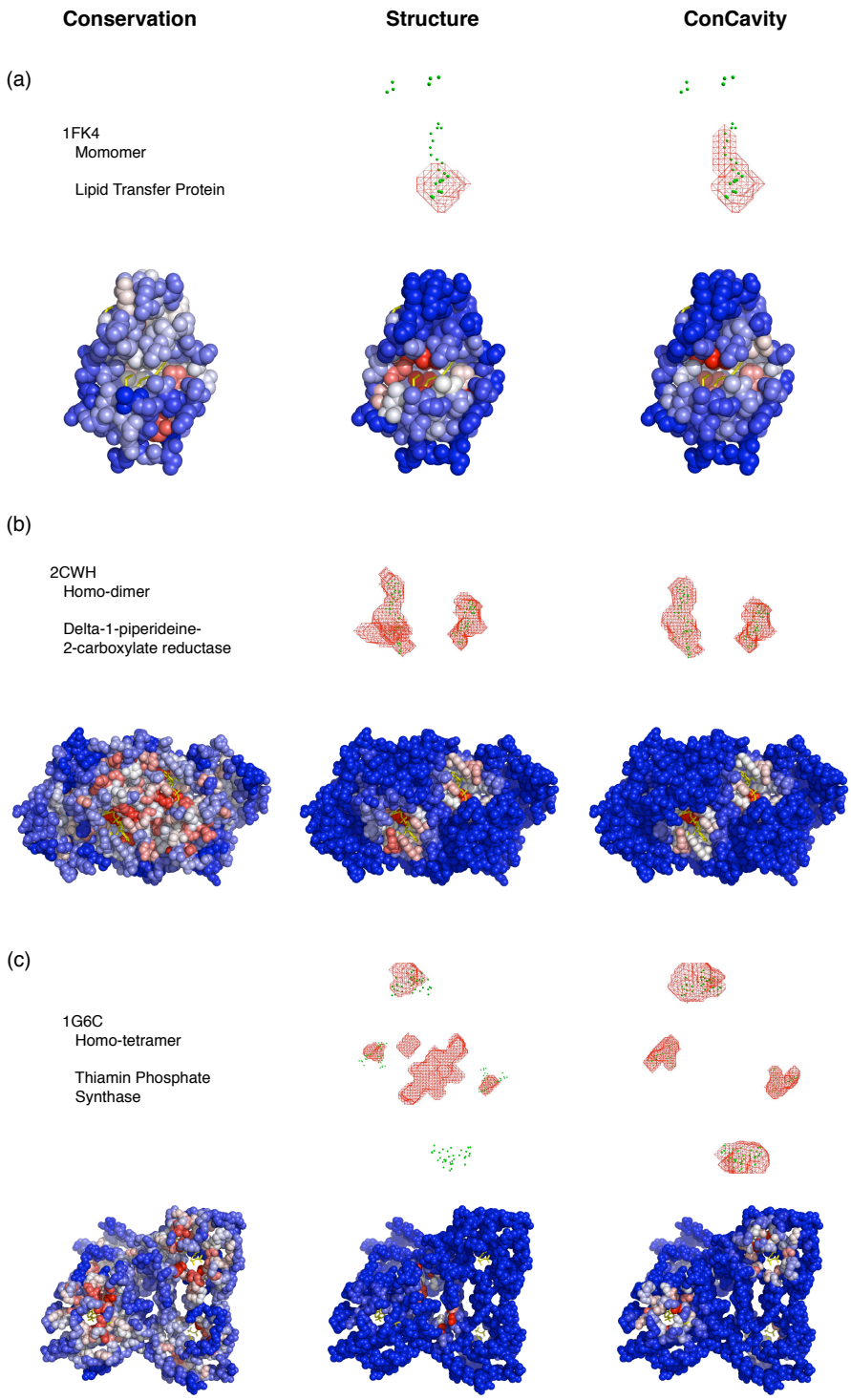

Figure 4.2: Example binding pocket and residue predictions of *Conservation*, *Structure*, and their combination in *ConCavity*. The top row in each pane gives isosurfaces that represent the pocket predictions of each algorithm. Ligand atoms are shown as green balls. Sequence conservation-based methods make only residue-level predictions and thus are not included in the pocket illustration. The bottom row of structures in each pane shows the residue predictions of each method; the highest scoring residues are colored dark red, and the lowest scoring are colored dark blue. *ConCavity* only gives high scores to regions containing ligands and accurately traces their shape.

ConCavity to more accurately trace the ligand's location. This illustrates how the pattern of functional conservation observed at the protein surface influences the shape of the predicted pocket. Ligands often do not completely fill surface pockets; if the contacting residues are conserved, our approach can suggest a more accurate shape.

The results for 2CWH (Figure 4.2b) and 1G6C (Figure 4.2c) demonstrate that *ConCavity* can predict dramatically different sets of pockets than are obtained when considering structure alone. In 2CWH, both methods identify the ligands, but *Structure* predicts an additional pocket that does not have a ligand bound. *ConCavity* does not predict this other pocket. *Structure* performs quite poorly on the tetramer 1G6C: it predicts several pockets that do not bind ligands; it fails to completely identify several ligands; and it misses one ligand entirely. In stark contrast, *ConCavity*'s four predicted pockets each accurately trace a ligand. The incorporation of conservation resulted in the accurate prediction of a pocket in a region where no pocket was predicted using structure alone.

***ConCavity* significantly outperforms available prediction servers.**

We now compare the performance of *ConCavity* to several existing ligand binding site identification methods with publicly available web servers: *LigsiteCS* [82][1], *LigsiteCSC*$^{+}$ [82], *QSiteFinder* [99][2], and *CASTp*[3] [95]. *LigsiteCS* and *LigsiteCSC* are updated versions of geometry-based *Ligsite*; *LigsiteCSC*$^{+}$ is our implementation of *LigsiteCSC* using JSD as its conservation scoring method. *Q-SiteFinder* estimates van der Waals interactions between the protein and a probe in a fashion similar to *PocketFinder*. *CASTp* is a geometry-based algorithm for finding pockets based on analysis of the protein's alpha shape. Each of these servers produces a list of predicted pockets represented by sets of residues; however, none of them provide a full representation of a predicted pocket. As a result, we compare their ability to predict ligand binding residues. See the Methods section for more information on the generation and processing of the servers' predictions.

Figure 4.3 presents the ligand binding residue PR-curves for each of these methods. *ConCavity* significantly outperforms *LigsiteCS*, *LigsiteCSC*$^{+}$, *Q-SiteFinder*, and *CASTp*. Several of the servers did not produce predictions for a small subset of the proteins in the database, e.g., the *Q-SiteFinder* server does not accept proteins with more than 10,000 atoms. Figure 4.3 is based on the 293 proteins for which all methods produced predictions. Thus the curve for *ConCavity* is slightly different than those found in the other figures, but its performance does not change significantly. *LigsiteCSC*$^{+}$ is the previous method most similar to *ConCavity*; it uses sequence conservation to rerank the pockets predicted by *LigsiteCS*. *LigsiteCSC*$^{+}$ provides slight improvement over *LigsiteCS*, but the

[1] http://gopubmed2.biotec.tu-dresden.de/pocket/
[2] http://www.modelling.leeds.ac.uk/qsitefinder/
[3] http://sts-fw.bioengr.uic.edu/castp/

improvement is dwarfed by that of *ConCavity* over *Structure* (Figure 4.1). This illustrates the benefit of incorporating conservation information directly into the search for pockets in contrast to using conservation information to post-process predicted pockets.

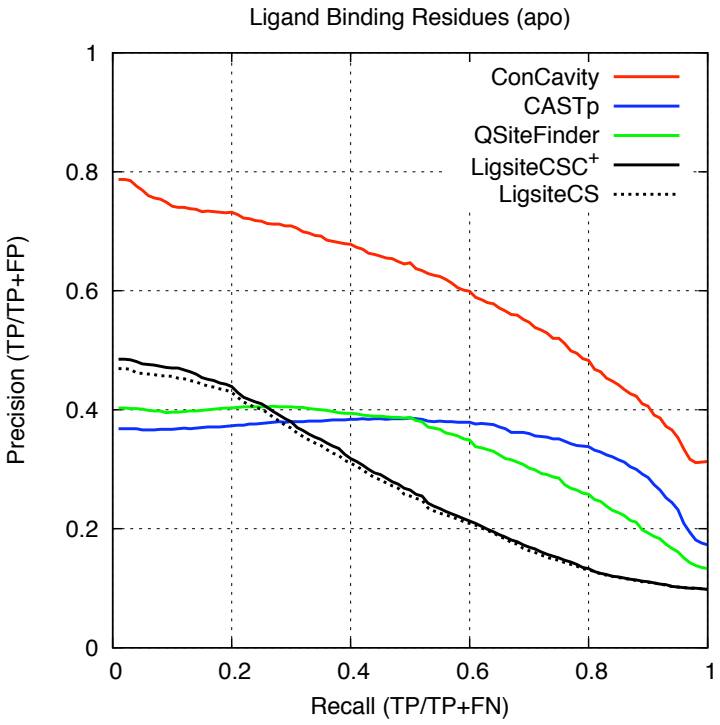

Figure 4.3: Comparison of *ConCavity* with publicly available ligand binding site prediction servers. *ConCavity* significantly outperforms each previous method at the prediction of ligand binding residues. In contrast to these methods, *ConCavity* assigns residues in the same pocket different likelihoods of binding ligands. This ability and the direct integration of sequence conservation are the major sources of *ConCavity*'s improvement. This figure is based on the 293 proteins in the LigASite apo dataset for which all methods were able to produce predictions.

The poor performance of these previous methods at identifying ligand binding residues is due in part to the fact that each of them gives all residues near a predicted binding pocket the same score. The entire pocket is a useful starting place for analysis, but knowledge of the specific ligand binding residues is of most interest to researchers. Many residues in a binding pocket will not actually contact the ligand. The predictions of our methods reflect this—residues within the same pocket can receive different ligand binding scores. The inability of previous methods to distinguish residues in a pocket is one reason why we elect to use our implementation of *Ligsite* (aka *Structure*) as representative of the performance of structure-based methods; see the methods section for more details.

***ConCavity* performs similarly for geometry and energetics based grid creation methods.**

In the previous sections, we used $ConCavity_L$, which integrates evolutionary sequence conserva-

tion and *Ligsite*$^+$, to represent the performance of the *ConCavity* approach. However, our strategy for combining sequence conservation with structural predictions is very general; it can be used with a variety of grid-based surface pocket identification algorithms.

Figure 4.4 gives PR-curves that demonstrate that *ConCavity* provides excellent performance whether the structural approaches are based on geometric properties (*Ligiste*$^+$, *Surfnet*$^+$) or energetics (*PocketFinder*$^+$). This holds for predicting both ligand locations in space (Figure 4.4a) and ligand binding residues (Figure 4.4b). The three *ConCavity* versions each outperform all structure only methods, and all perform similarly despite the variation in performance between *Ligiste*$^+$, *Surfnet*$^+$, and *Pocketfinder*$^+$. In the following sections we will include results for all three methods when space and clarity allow.

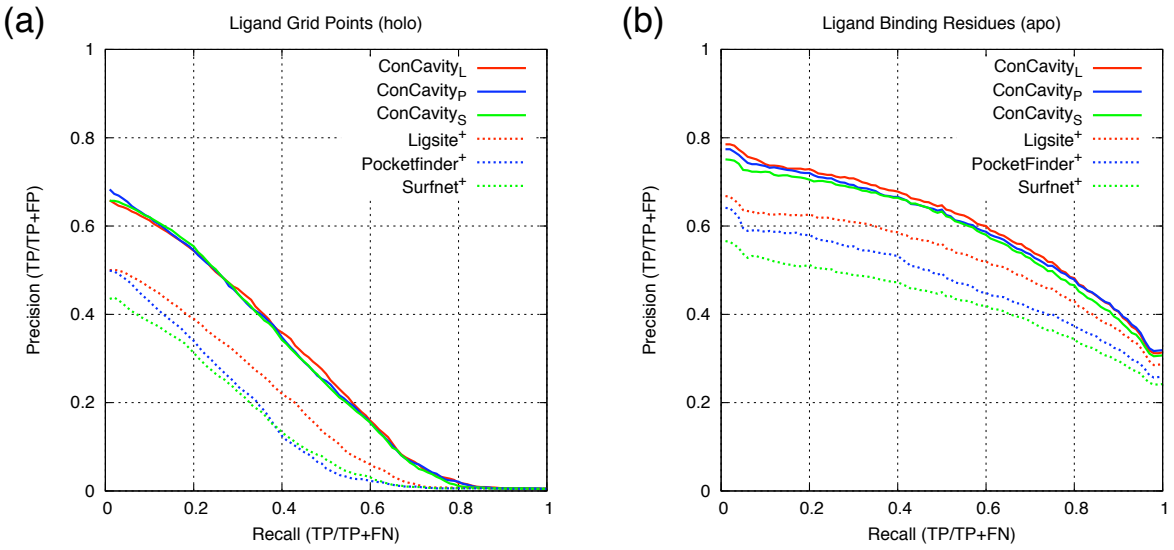

Figure 4.4: Ligand binding pocket **(a)** and residue **(b)** prediction performance of different versions of *ConCavity* on the LigASite database. The three versions of *ConCavity* (solid lines)—each based on integrating sequence conservation with a different grid creation strategy (*Ligsite*$^+$, *Surfnet*$^+$, and *PocketFinder*$^+$)—perform similarly. All significantly outperform the methods based on structure analysis alone (dashed lines). These conclusions hold for both ligand binding pocket (a) and ligand binding residue (b) prediction.

Structure-based methods have difficulty with multi-chain proteins.

Proteins consisting of multiple subunits generally have more pockets than single-chain proteins due to the gaps that often form between chains. To investigate the effect of structural complexity on performance, we partitioned the dataset according to the number of chains present in the PQS quaternary structure and performed our previous evaluations on the partitioned sets. Figure 4.5 gives these statistics for *ConCavity*, *Structure*, and *Conservation*; to enable side-by-side comparison, we

report the area under the PR curves (PR-AUC) rather than giving the full curves.

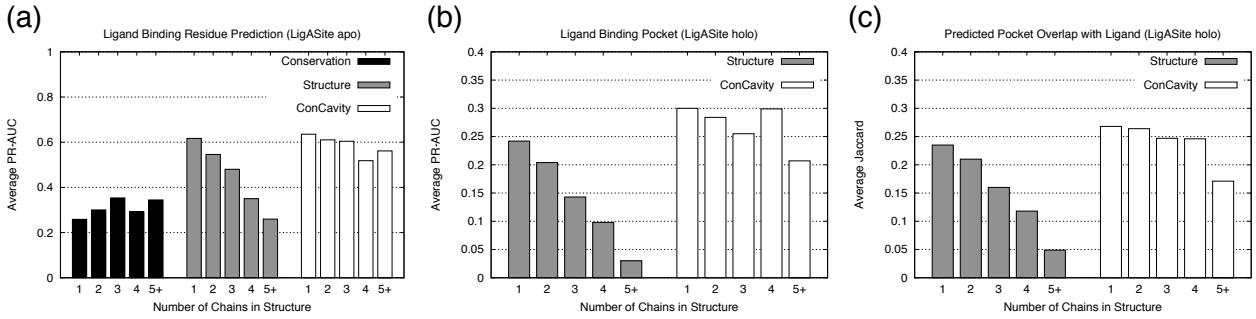

Figure 4.5: Ligand-binding site identification performance by number of chains in structure. **(a)** gives the average area under the precision-recall curve (PR-AUC) for predicting ligand binding residues on each set of structures; **(b)** gives the average PR-AUC for ligand binding pocket identification; and **(c)** gives the average Jaccard coefficient of the overlap of the predicted pockets with bound ligands. (See Table 4.1 and Figure 4.1 for more details on these statistics.) Methods based on structure alone have an increasingly difficult time distinguishing among ligand-binding pockets and non-ligand-binding gaps between chains as the number of chains in the protein increases. This trend is clear at each type of evaluation. Looking at (a), *Conservation*'s performance does not exhibit this effect. In fact, *Conservation* outperforms *Structure* on proteins with five or more chains. The integration of sequence conservation and pocket prediction in *ConCavity* improves performance in each chain based partition in each evaluation. *ConCavity* sees only a modest decrease in performance on proteins with multiple chains. *Conservation* could not be included in (b) and (c), because it does not make pocket predictions. Number of structures per chain group: 1 chain: 132, 2 chains: 112, 3 chains: 22, 4 chains: 36, 5+ chains: 29.

As the number of chains in the structure increases, there is a substantial decrease in the performance of the methods based on structural pocket prediction alone. The pattern is seen both when predicting ligand binding residues (Figure 4.5a) and pockets (Figure 4.5b and c). This effect is so strong that, for proteins with five or more chains, *Conservation* outperforms *Structure*. The number of chains in the protein has little effect on *Conservation*'s performance. These observations emphasize the importance of including multi-chain proteins in the evaluation.

The homo-tetramer 1G6C in Figure 4.2c provides an illustrative example of the failure of *Structure* on multi-chain proteins. There is a large gap between the chains in the center of the structure, and several additional pockets are formed at the interface of pairs of contacting chains. As seen in the figure, the large central cavity does not bind a ligand, however it is the largest pocket predicted by *Structure*. This is frequently observed among *Structure*'s predictions. While some pockets between protein chains are involved in ligand binding, many of them are not. As the number of chains increases, so does the number of such potentially misleading pockets.

By incorporating sequence conservation information, *ConCavity* accurately identifies ligand binding pockets in multi-chain proteins. The conservation profile on the surface of 1G6C provides a clear

example of this; the pockets that exhibit sequence conservation are those that bind ligands. 1G6C is not an exception. *ConCavity* provides significant performance improvement for each partition of the dataset in all three evaluations, and greatly reduces the effect of the large number of non-ligand-binding pockets in multi-chain proteins on performance. *ConCavity* also provides improvement over *Structure* on the set of one chain proteins. This is notable because these proteins do not have between-chain gaps, so the improvement comes from tracing ligands and selecting among intra-chain pockets more accurately than using structure alone (as in Figure 4.2a).

***ConCavity* performs well on both apo and holo structures.**

The binding of a ligand induces conformational changes to a protein [121]. As a result, the 3D structure of the binding site can differ between structures of the same protein with a ligand bound (holo) and not bound (apo). In the holo structures, the relevant side-chains are in conformations that contact the ligand, and this often defines the binding pocket more clearly than in apo structures. To investigate the effect of the additional information provided in holo structures on performance, we evaluated the methods' on both sets (Table 4.2).

	PR-AUC	
Method	apo	holo
$ConCavity_L$	0.606	0.649
$ConCavity_P$	0.598	0.629
$ConCavity_S$	0.587	0.634
$Ligsite^+$	0.525	0.550
$PocketFinder^+$	0.472	0.497
$Surfnet^+$	0.426	0.472

Table 4.2: Comparison of area under the Precision-Recall curve (PR-AUC) for ligand-binding residue prediction methods on apo (unbound) and holo (bound) versions of LigASite. All methods perform nearly as well on the apo structures as on the holo structures, and the relative ranking of the methods is the same across datasets.

As expected, all methods performed better on the holo (bound) structures than the apo (unbound) structures; however, our methods only experience a modest drop in performance. All previous conclusions hold whether considering apo structures or holo structures; the ranking of the methods is consistent, and the improvement provided by considering conservation is similarly large. We will continue to report residue prediction results on the apo structures in order to accurately assess the performance of the algorithms in common, real-world situations faced by ligand binding site prediction methods.

The methods better identify ligand binding sites in enzymes than non-enzymes.

The LigASite apo dataset contains protein molecules that carry out a range of different functions. Enzymes are by far the most common; they make up 245 of the 331 proteins in the dataset. The remaining 86 non-enzyme ligand binding proteins are involved in a wide variety of functions, e.g., transport, signaling, nucleic acid binding, and immune system response.

Table 4.3 compares the performance of the ligand binding site prediction methods on enzymes and non-enzymes. All methods perform significantly better on the enzymes. Active sites in enzymes are usually found in large clefts on the protein surface and consistently exhibit evolutionary sequence conservation [44, 122], so even though enzymes bind a wide array of substrates, these common features simplify prediction when compared to the variety of binding mechanisms found in other proteins.

	PR-AUC	
Method	Enzymes	Non-enzymes
ConCavity$_L$	0.663	0.441
ConCavity$_P$	0.654	0.436
ConCavity$_S$	0.647	0.419
Ligsite$^+$	0.555	0.439
PocketFinder$^+$	0.506	0.376
Surfnet$^+$	0.452	0.354
Conservation	0.320	0.202

Table 4.3: Ligand binding residue identification in enzymes and non-enzymes (LigASite apo). All methods are better at identifying binding residues in enzymes than in non-enzymes. The *ConCavity* methods achieve the best performance on both datasets, but considering conservation information provides less improvement in non-enzymes.

Despite the drop in performance on non-enzyme proteins, the main conclusions from the earlier sections still hold. However, the improvement provided by *ConCavity* is not as great on the non-enzymes—*Ligsite*$^+$ performs nearly as well as *ConCavity*$_L$. This could be the result of the more complex patterns of conservation found in non-enzyme proteins, and the poor performance of *Conservation* in this setting. It is also possible that *Ligsite*$^+$'s approach is particularly well suited to identifying binding sites in non-enzymes. Overall, these results highlight the importance of using a diverse dataset to evaluate functional site predictions.

***ConCavity* improves identification of drug binding sites.**

Knowledge of small molecule binding sites is of considerable use in drug discovery and design.

Many of the techniques used to screen potential targets, e.g., docking and virtual screening, are computationally intensive and feasible only when focused on a specific region of the protein surface. Structure based surface cavity identification algorithms are commonly used to guide analysis in such situations [116].

The superior performance provided by *ConCavity* over *Structure* on the diverse set of proteins considered above suggests that *ConCavity* would likely be useful in the drug screening pipeline. To test *ConCavity*'s ability to identify drug binding sites, we evaluated it on a set of 99 protein-drug complexes [123]. Table 4.4 compares the ligand overlap PR-AUC and Jaccard coefficient for the three versions of *ConCavity* and their structure-based analogs. Each *ConCavity* method significantly improves on the methods that only consider structural features. While significant, the improvement is not quite as large on this dataset as that seen on the more diverse LigASite dataset (Table 4.1). It is possible that this is due to the fact that drug compounds are not the proteins' natural ligands; the evolutionary conservation of the residues in binding pockets may reflect the pressures related to binding the actual ligands rather than the drugs.

Method	Grid Value PR-AUC	Jaccard coefficient
$ConCavity_L$	0.249	0.231
$ConCavity_P$	0.238	0.223
$ConCavity_S$	0.256	0.232
$Ligsite^+$	0.202	0.193
$PocketFinder^+$	0.180	0.164
$Surfnet^+$	0.157	0.164

Table 4.4: Drug binding site identification performance. This table compares the median Jaccard coefficient of prediction-ligand overlap and the average grid value precision-recall AUC for *ConCavity* and methods based on structural analysis alone on a set of 99 protein-drug complexes. (See Table 4.1 and Figure 4.1 for more details on these statistics.) Integrating sequence conservation and structure-based pocket finding improves the identification of drug binding sites. *Conservation* is not included in this evaluation, because it does not make pocket-level predictions.

Sequence alignment quality affects the improvement provided by ConCavity.

Estimates of evolutionary conservation, such as the *Jensen-Shannon divergence*, rely on multiple sequence alignments (MSAs) of homologous proteins. Alignment quality varies from protein to protein. For some proteins in the dataset, there are a large number of diverse homologous sequences, while others have only a few related sequences available.

To explore the effect of MSA quality on the performance of *ConCavity*, we partitioned the

dataset based on the fraction of gaps in the associated alignments. We consider the fraction of gaps because many of the lower quality alignments contain a large number of sequence fragments that only partially cover the structure's sequence, e.g., a single domain.

For the 245 structures in the LigASite apo dataset with alignments containing fewer than 40% gaps, the average PR-AUC improvement provided by the three *ConCavity* methods over their *Structure* counterparts is 0.132. For the 86 alignments with more than 40% gaps, the average improvement is 0.093. The 40% threshold was selected because there was a clear drop in average performance at this level, and performance is quite similar within finer partitions of these two groups.

This demonstrates that though alignment quality is important, considering evolutionary conservation can provide improvement over structure-alone, even when a high quality alignment is lacking. Fragmentary sequences provide useful information, but are not as reliable as fully aligned complete homologous proteins. There are cases where poor conservation estimates result in worse predictions than when considering structure alone. Since better estimates of the evolutionary constraints on residue positions lead to better ligand binding site predictions, *ConCavity* will likely become even more accurate as conservation estimation methods improve and more protein sequences become available.

Integrating conservation and structure improves prediction of catalytic sites, but the performance is worse than for ligand binding site prediction.

Ligand-binding sites are not the only type of functional site of interest to biologists. A large amount of attention has been given to the problem of identifying catalytic sites. As noted above, the majority of enzyme active sites are found in large clefts on the protein surface, so even though the structural methods considered in this chapter were not intended to identify catalytic sites, they could perform well at this task.

Table 4.5 gives the results of an evaluation of the methods' ability to predict catalytic sites (defined by the Catalytic Site Atlas [45]) in the LigASite apo dataset. Compared to ligand binding site prediction, the relative performance of the methods is different in this context. The *ConCavity* approach still performs the best, but sequence conservation alone performs nearly as well. Most surprisingly, *Conservation* significantly outperforms those based on structure alone. All the methods perform much worse when predicting catalytic sites than predicting ligand-binding residues, e.g., $ConCavity_L$ has PR-AUC of 0.273 versus 0.606.

These results imply that being very evolutionarily conserved is more indicative of a role in catalysis than being found in a surface pocket. Though catalytic sites are usually found in pockets

Method	PR-AUC
ConCavity$_L$	**0.273**
ConCavity$_P$	0.262
ConCavity$_S$	0.254
Conservation	0.233
Ligsite$^+$	0.168
PocketFinder$^+$	0.139
Surfnet$^+$	0.133

Table 4.5: Identifying catalytic residues in LigASite apo structures. *ConCavity* gives the best performance on this task as well, but in contrast to ligand binding residue prediction, *Conservation* outperforms the structure-based approaches. Overall, the methods perform worse at predicting catalytic sites than ligand binding sites.

near bound ligands, there are many fewer catalytic sites per protein than ligand-binding residues. As a result simply searching for residues in pockets identifies many non-catalytic residues. This is consistent with earlier machine learning studies that found conservation to be the dominant predictive feature, and it suggests that new structural patterns must be sought to improve the identification of catalytic sites.

Several previous methods have combined sequence conservation and structural properties in machine learning frameworks to predict catalytic sites [29, 85, 115]. Direct comparison with these methods is quite difficult because of unavailable datasets and algorithms. Tong et al. [115] compared the precision and recall of several machine learning methods on different datasets in an attempt to develop a qualitative understanding of their relative performance. While it is not prudent to draw conclusions based on cross-dataset comparisons, we note for completeness that *ConCavity*'s catalytic site predictions on a diverse dataset achieve higher precision (21%) at full recall than any of the methods reported in their comparisons at much lower recall.

4.3 Discussion

Evolutionary sequence conservation and protein 3D structures have commonly been used to identify functionally important sites; here, we integrate these two approaches in *ConCavity*, a new algorithm for ligand binding site prediction. By evaluating a range of conservation and structure-based prediction strategies on a large, diverse dataset of ligand binding sites, we establish that structural approaches generally outperform sequence conservation, and that by combining the two, *ConCavity*

outperforms conservation-alone and structure-alone on about 95% and 70% of structures respectively. Overall, *ConCavity*'s first predicted residue contacts a ligand in nearly 80% of the apo structures examined, and it maintains high precision across all recall levels. These results hold for the three variants of *ConCavity* we considered, each of which uses a different underlying structure-based component. In addition, *ConCavity*'s integrated approach provides significant improvement over conservation and structure-based approaches on the common task of identifying drug binding sites.

Combining sequence conservation-based methods with structure information is especially powerful in the case of multimeric proteins. Our analysis has shown that the performance of structural approaches for identifying ligand binding sites dramatically decreases as the number of chains in the structure increases; conservation alone outperforms structure-based approaches on proteins with five or more chains. It is difficult to determine from structural attributes alone if a pocket formed at a chain interface binds a ligand or not. However, ligand binding pockets usually exhibit high evolutionary sequence conservation. *ConCavity*, which takes advantage of this complementary information, performs very well on multi-chain proteins; the presence of many non-ligand binding pockets between chains has little effect on its performance.

While *ConCavity* outperforms previous approaches, we have found two main causes of inaccurate predictions: low-quality estimates of evolutionary conservation and ligand binding sites with limited conservation. Some proteins lack a large number of known complete homologous sequences; conservation estimates for these proteins may not be as reliable as those based on better alignments. We have shown that though estimates based on low quality sequence alignments may harm performance for some structures, they provide a net performance gain over all such structures. This issue will become less relevant as sequence data coverage and conservation estimation methods improve, but users should take note of the quality of the alignment used. The second problematic situation occurs when a ligand binding site lacks strong conservation. *ConCavity* is less likely to predict such a pocket due to its low conservation. It is possible that some of these sites are hypervariable [7] for functional reasons. Missing or incomplete ligands can also affect the apparent performance of the methods, but such issues are unavoidable due to the nature of the data. A small number of proteins bind biologically relevant ligands outside of protein surface pockets. These situations pose difficulties for our method, but are not common in the dataset. Additional predictors could be incorporated into our model to address these cases; alternatively, a wide range of predictors could be used within a machine learning framework to obtain further performance improvements.

In implementing and evaluating previous 3D grid-based ligand binding site prediction approaches,

we have found that the methods used both to aggregate grid values into coherent pockets as well as to map these pockets onto surface residues can have a large effect on performance. We describe the details of these new algorithms in the Methods section. On a high level, the new methodologies we propose for these tasks provide significant improvement by predicting a flexible number of well-formed pockets for each structure and by assigning each residue a likelihood of binding a ligand based on its local environment rather than on the rank of the entire pocket. We have used morphological properties of ligands to guide pocket creation, but the most appropriate algorithms for these steps depend strongly on the nature of the prediction task. These steps have received considerably less attention than computing grid values; our results suggest that they should be given careful consideration in the future.

We have focused on the prediction of ligand binding sites, but the direct synthesis of conservation and structure information is likely to be beneficial for predicting other types of functionally important sites. Our application of *ConCavity* to catalytic site prediction illustrates the promise and challenges of such an approach. Catalytic sites are usually found in surface pockets, but considering structural evidence alone performs quite poorly—worse than sequence conservation. Combining structure with evolutionary conservation provides only a very modest gain in performance over conservation alone. Protein-protein interface residues are another appealing target for prediction; much can be learned about a protein by characterizing its interactions with other proteins. However, protein-protein interaction sites provide additional challenges; they are usually large, flat, and often poorly conserved [9]. *ConCavity* is not appropriate for this task. Other types of functional sites also lack simple attributes that correlate strongly with functional importance. Analysis of these sites' geometries, physical properties, and functional roles might, in addition to producing accurate predictors, lead to new insights about the general mechanisms by which proteins accomplish their molecular functions.

In summary, this work significantly advances the state of the art in ligand binding site identification by improving the philosophy, methodology, and evaluation of prediction methods. It also increases our understanding of the relationship between evolutionary sequence conservation, structural attributes of proteins, and functional importance. By making our source code and predictions available online, we hope to establish a platform from which the prediction of functional sites and the integration of sequence and structure data can be investigated further.

4.4 Methods and Algorithms

4.4.1 *ConCavity*

This section describes the components of the *ConCavity* algorithm for predicting ligand binding residues from protein 3D structures and evolutionary sequence conservation.

ConCavity proceeds in three conceptual steps: grid creation, pocket extraction, and residue mapping (Figure 4.6). First, the structural and evolutionary properties of a given protein are used to create a regular 3D grid surrounding the protein in which the score associated with each grid point represents an estimated likelihood that it overlaps a bound ligand atom (Figure 4.6a). Second, groups of contiguous, high-scoring grid points are clustered to extract pockets that adhere to given shape and size constraints (Figure 4.6b). Finally, every protein residue is scored with an estimate of how likely it is to bind to a ligand based on its proximity to extracted pockets (Figure 4.6c).

On a high-level, this grid-based strategy has been employed by several previous systems for ligand binding site prediction. However, our adaptations to the three steps significantly affect the quality of predictions. First, we demonstrate how to integrate evolutionary information directly into the grid creation step for three different grid-based pocket prediction algorithms. Second, we introduce a method that employs mathematical morphology operators to extract well-shaped pockets from a grid. Third, we provide a robust method for mapping grid-based ligand binding predictions to protein residues based on Gaussian blurring. The details of these three methods and an evaluation of their impacts on ligand-binding predictions are described in the following subsections.

Grid Creation

The first step of our process is to construct a 3D regular grid covering the free-space surrounding a protein. The goal is to produce grid values that correlate with the likelihoods of finding a bound ligand at every grid point.

Several methods have been previously been proposed to produce grids of this type. For example, *Ligsite* [92] produces a grid with values between 0 and 7 by scanning for the protein surface along the three axes and the four cubic diagonals. For each grid point outside of the protein, the number of scans that hit the protein surface in both directions—so-called protein-solvent-protein (PSP) events—is the value given to that point. A large number of PSP events indicate that the grid point is surrounded by protein in many directions and thus likely to be in a pocket.

Surfnet [90] assigns values to the grid by constructing spheres that fill the space between pairs of protein atoms without overlapping any other atoms. These sets of spheres are constructed for all

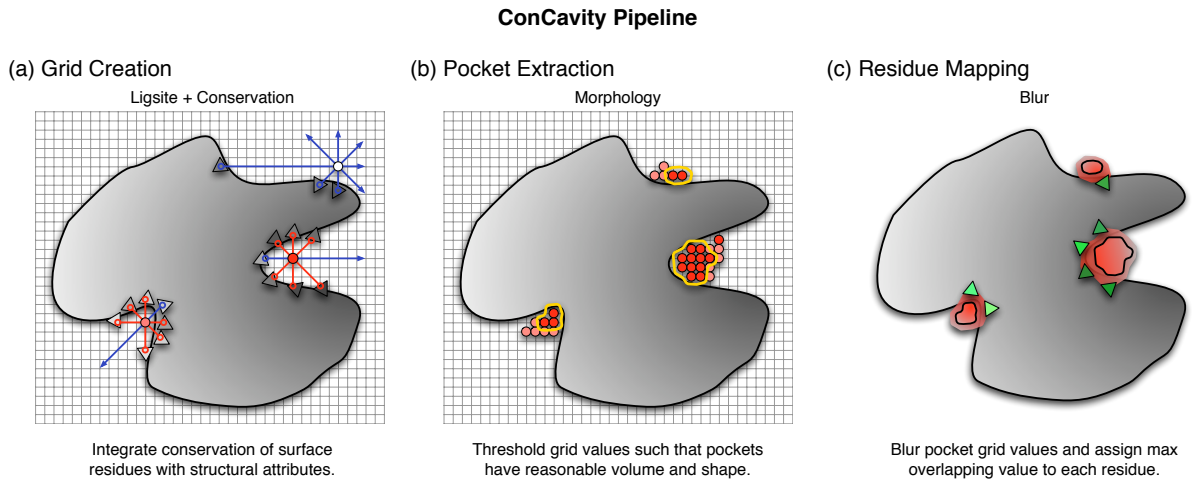

Figure 4.6: *ConCavity* prediction pipeline. The blob represents a protein 3D structure; the triangles represent surface residues; and the gray gradient symbolizes the varying sequence conservation values in the protein. Darker shades of each color indicate higher values. **(a)** The initial grid values come from the combination of sequence conservation information and a structural predictor, in this example *Ligsite*. The algorithm proceeds similarly for *PocketFinder* and *Surfnet*. **(b)** The grid generated in (a) is thresholded based on morphological criteria so that only well-formed pockets have non-zero values. For simplicity, only grid values near the pockets are shown. **(c)** Finally, the grid representing the pocket predictions is mapped to the surface of the protein. We perform a 3D Gaussian blur ($\sigma = 4$Å) of the pockets, and assign each residue the highest overlapping grid value. Residues near regions of space with very high grid values receive the highest scores.

pairs of protein surface atoms within 10Å of each other. Spheres with a radius smaller than 1.5Å are ignored, and spheres are allowed to have a maximum radius of 4Å. This procedure results in a set of overlapping spheres that fill cavities and clefts in the protein. Extending the original algorithm slightly, we assign the value for each grid point to be the number of spheres that overlap it (rather than simply 1 for overlap and 0 for no overlap as in the original algorithm). Thus, higher values are generally associated with the positions in the "center" of a pocket.

PocketFinder [98] assigns values to grid points by calculating the van der Waals interaction potential of an atomic probe with the protein. The Lennard-Jones function is used to estimate the interaction potential between the protein and a carbon atom placed at each grid point. The potential at a grid point p is:

$$V(p) = \sum_{\alpha \in \textbf{protein}} \left(\frac{C_{12}^{\alpha}}{r^{12}} - \frac{C_{6}^{\alpha}}{r^{6}}\right) \tag{4.1}$$

where C_{12}^{α} and C_{6}^{α} are constants (taken from AutoDock [124]) that shape the Lennard-Jones function according to the interaction energy between the carbon probe atom and protein atom α, and r is the distance between the grid point p and α (interactions over distances greater than 10Å are ignored).

Other grid creation methods have been proposed as well, but these three (*Ligsite*, *Surfnet*, and

PocketFinder) provide a representative set for our study.

We augment these algorithms by integrating evolutionary information into the grid creation process. Our methodology is based on the observation that these (and other) grid creation algorithms operate by accumulating evidence ("votes") for ligand binding at grid points according to spatial relationships to nearby protein atoms. For *PocketFinder*, each protein atom casts a "vote" for nearby grid points with magnitude equal to the (opposite of the) van der Waals potential. In *Ligsite*, every pair of protein atoms "votes" for solvent-accessible grid points on line segments between them. In our implementation of *Surfnet*, pairs of atoms "vote" for all the grid points overlapping a sphere covering the solvent accessible region between them.

Based on this observation, we weight the "votes" as the grid is created by an estimate of sequence conservation of the residue(s) associated with the atom(s) that generate the votes. We tested several schemes for scaling votes. If c_1 and c_2 are estimated conservation scores associated with the relevant atoms (e.g., derived from their residues' conservation in multiple sequence alignments), we scaled the structure-based component by the product ($c_1 c_2$), the arithmetic mean ($\frac{c_1+c_2}{2}$), the geometric mean ($\sqrt{c_1 c_2}$), the product of exponentials ($2^{c_1} 2^{c_2}$), and the product of exponentials of transformed conservation values ($2^{2c_1-1} 2^{2c_2-1}$). Each of these schemes provides improvement for all methods, but due to method specific differences, no single weighting scheme is best for all methods. Specifically, for *PocketFinder*, which has only one atom associated with each vote, we scale the vote (van der Waal's potential) of each atom linearly by c_1. For *Ligsite* we scale the votes by arithmetic mean of the conservation values and for *Surfnet* by the product of the exponentials of the transformed conservation values.

In our study, conservation scores are calculated by the Jensen-Shannon divergence (JSD) with sequence weighting and a gap penalty [5]; however, any sequence conservation measure that produces residue scores (which are then mapped to atoms within the residues) could be incorporated.

Performance

The superior performance of our *ConCavity* grid creation method at predicting ligand binding pockets and residues is demonstrated in Figure 4.4 of the Results section. The only difference between the *ConCavity* methods (*ConCavity*$_L$, *ConCavity*$_S$, *ConCavity*$_P$) and their counterparts based on structure alone (*Ligsite*$^+$, *Surfnet*$^+$, *PocketFinder*$^+$) is the use of sequence conservation in the grid creation step. For each grid creation strategy considering evolutionary conservation yields significant improvement.

Pocket Extraction

The second step of our process is to cluster groups of contiguous, high-scoring grid points into *pockets* that most likely contain bound ligands.

Several methods have been previously proposed to address this problem. The simplest is to apply a fixed threshold to the grid, i.e., eliminate all grid points below some given value. Then, the remaining grid points can be clustered into pockets (e.g., connected components), and small pockets can be discarded. This method, which we call "*Threshold*", is used in *Ligsite*, for example, where only grid values greater than or equal to two are included in [92] and greater than or equal to six in [82]. A problem with this approach is that the threshold is set to the same value for all proteins, which provides no control over the total number and size of pockets predicted by the algorithm. In the worst case, when every grid value is below the threshold, then the algorithm will predict no pockets. On the other hand, if the threshold is too low, then there will be many large pockets. Different proteins have different types of pockets, so no one threshold can extract appropriately sized and shaped pockets for all of them.

A slightly more adaptive method is used in *PocketFinder* [98]. In "*StdDev*" the mean and standard deviation of values in the grid are used to determine a different threshold for every protein. Specifically, the grid is blurred with $\sigma = 2.6$Å, and then the threshold is set to be 4.6 standard deviations above the mean of the grid values. This approach is problematic because the threshold depends on the parameters of the grid; any change to how the protein is embedded in the grid (e.g., orienting the protein differently, changing the distance between the protein and the grid boundary, etc.) will affect the mean and standard deviation of the grid values, which in turn will affect the threshold chosen to extract pockets. For example, simply making the extent of the grid 10% greater will include a large number of near-zero values in the grid, which will bring the threshold down and make the extracted pockets larger. Also, no control is provided over the number and size of pockets; it is possible that for some proteins no grid values are 4.6 standard deviations above the mean, in which case no pockets will be predicted.

It is difficult to control the number, sizes, and shapes of extracted pockets with *Threshold* and *StdDev*. In both methods a threshold is applied to every grid point independently and clusters are formed only on the basis of geometric proximity between grid points, so it is possible to extract a set of pockets that have biologically implausible shapes. For example, there is no way to guarantee that the algorithm won't extract one very large pocket that covers a significant fraction of the protein surface, or many small pockets distributed across the protein surface, and/or pockets that contain

long, thin regions where the cross-sectional diameter is too small to fit a bound ligand. Of course, it is possible to trim/discard such pockets after they have been extracted according to geometric criteria using post-processing algorithms [82, 98, 120]. However, unless there is feedback between the method used to select a grid threshold and the method used to cull pockets, then there is no way to guarantee that a biologicaly plausible set of pockets is output, i.e., it is possible that none of the pockets extracted with the chosen grid threshold meet the culling criteria.

In *ConCavity*, we integrate extraction and culling of pockets into a single framework. We perform a binary search for the grid threshold that produces a culled set of pockets that have specified properties (maximum number of pockets, total volume of all pockets, minimum volume for any pocket, minimum cross-sectional radius for any pocket, and maximum distance from protein surface). Specifically, for each step of the binary search, we select a grid threshold, extract a set of pockets (connected components of grid points having values above the threshold), and then apply a sequence of culling algorithms to trim/discard pockets based their sizes and shapes. The algorithm iterates, adjusting the threshold up or down, if the set of pockets resulting from the culling operations does not meet the specified global properties. The binary search terminates when it has found a set of pockets meeting all of the specified properties or determines that none is possible. We call this method "*Search*".

Specifically, the culling steps are implemented with a series of grid-based filters, each of which runs in compute time that grows linearly with the size of the grid. Given a current guess for the grid threshold, the first filter simply zeroes all grid points whose value is below the threshold value.

The second filter zeroes grid points whose distance from the van der Waal's surface of the protein exceeds a given threshold, *max_protein_offset*. This filter is computed by first rasterizing a sphere for all atoms of the protein into a grid, setting every grid point within the van der Waal's radius of any protein atom to one and the others to zero. Then, the square of the distance from each grid point to the closest point on the van der Waal's surface is computed with three linear-time sweeps, and the resulting squared distances are used to zero grid points of the original grid if the squared distance is greater than $max\_protein\_offset^2$.

The third filter ensures that no part of a pocket has cross-sectional radius less than a given threshold, *min_pocket_radius*. This filter is implemented with an "opening" operator from mathematical morphology . Intuitively, the boundary of every pocket (non-zero values of the grid) is "eroded" by *min_pocket_radius* and then "dilated" by the same amount, causing regions with cross-sectional radius less than the threshold to be removed, while the others are unchanged. This operator is implemented with two computations of the squared distances from pocket boundaries, each of which

takes linear time in the size of the grid.

The fourth filter constructs connected components of the grid and then zeros out grid points within components whose volume is less than a given threshold, *min_pocket_volume*. Connected components are computed with a series of depth-first traversals of neighboring non-zero grid points, which take linear time all together, and pockets are sorted by volume using quicksort, which takes $O(plogp)$ time for p pockets.

After these filters are executed for each iteration, the total volume of all remaining pockets is computed and compared to a given target volume, *total_pocket_volume*. If the total volume is greater (less) than the target, the grid threshold is increased (decreased) to a value half-way between the current threshold and the maximum (minimum) possible threshold—initially the largest (smallest) value in the grid—and the minimum (maximum) is set to the current threshold. The process is repeated with the new threshold until the total volume of all pockets is within *epsilon* of the given *total_pocket_volume*. Note that we perform a 1Å Gaussian blur on the *Ligsite* grid before beginning this search to provide finer control over the predicted pockets than is provided by the *Ligsite* integer grid values.

We set the parameters for these filters empirically. In previous studies, it has been observed that the vast majority of bound ligand atoms reside within 5Å of the protein's van der Waal's surface, thus we set *max_protein_offset* to 5Å. In order to target binding sites for biologically relevant ligands, we set *min_pocket_radius* to 1Å and *min_pocket_volume* to 100Å^3. Based on the observation that the total volume of all bound ligands is roughly proportional to the total volume of the protein [93], we set *total_pocket_volume* to a given fraction of the total protein volume—2% in our studies. Finally, we set the grid resolution to 1Å and *epsilon* to 1Å^3.

Performance

To assess the impact of different pocket extraction strategies on the precison and accuracy of binding site detection, we implemented several alternative methods and compared how well the pockets they predict overlap with ligands in holo structures from the LigASite dataset. Table 4.6 shows the results for three different types of grids (left column). The second column lists the pocket extraction algorithm. In addition to *Thresh* and *StdDev*, *Largest(N)* refers to zeroing all grid entries not in the largest N pockets (connected components). The third column third column ("Prediction Vol.") lists the median volume of all predicted pockets over each protein. For reference, the median volume of all ligands observed in the PQS files ("Ligand") is 1164.5Å^3. The next two columns list the median volumes of the Intersection (Ligand $\cap$ Prediction) and Union (Ligand $\cup$ Prediction) of

the Prediction and Ligand grids. Finally, the rightmost three columns list the median, precision (Intersection / Prediction), recall (Intersection / Ligand), and Jaccard coefficient (Intersection / Union). For the last three columns, higher values (between 0 and 1) represent better results.

Grid Generation	Pocket Extraction	Prediction Vol. (Å^3)	∩ w/ Lig. (Å^3)	∪ w/ Lig. (Å^3)	∩ / Prediction	∩ / Ligand	Jaccard coeff.
Ligsite$^+$	Thresh(6)	1828.1	331.5	2669.1	1.657	0.331	0.111
Ligsite$^+$	Thresh(6), Largest(3)	1545.1	313.2	2491.4	1.373	0.296	0.108
Ligsite$^+$	Search	1167.3	333.6	2080.5	0.286	0.332	**0.179**
Surfnet$^+$	-	27770.3	1023.1	27809.2	0.037	0.951	0.036
Surfnet$^+$	Search	1153.3	300.1	2099.2	0.251	0.286	**0.153**
Pocketfinder$^+$	Thresh($mean$+1σ)	49624.7	1065.7	49649.3	0.022	0.927	0.022
Pocketfinder$^+$	Thresh($mean$+2σ)	9874.2	704.8	10581.6	0.072	0.670	0.069
Pocketfinder$^+$	Thresh($mean$+2.5σ)	3663.9	494.1	4247.4	0.139	0.485	0.119
Pocketfinder$^+$	Thresh($mean$+3σ)	1312.6	313.1	2328.7	0.217	0.321	0.138
Pocketfinder$^+$	Search	1182.4	306.2	2138.5	0.246	0.291	**0.155**

Table 4.6: Comparison of pocket extraction methods. For three types of grids (left column), we ran different pocket extraction algorithms (second column) and compared how well the pockets overlap bound ligands in holo PQS structures. Each value presented is the median over all structures in the dataset; see the text for a description of each statistic. The median volume of all ligands observed over the structures ("Ligand") is 1164.5Å^3. Comparing to the median volume of the pockets predicted by each method, we see that *Search*'s pockets are closest to the actual ligand volumes. Moreover, *Search*'s high Jaccard coefficient for each grid type indicates that it provides the best tradeoff between recall and precision among the methods tested.

The statistics presented in Table 4.6 reflect various attributes of the pockets predicted by each extraction technique. The Jaccard coefficient (Intersection / Union) takes into account the natural tradeoff between recall and precision by rewarding predictions that overlap the known ligands (large Intersection) and penalizing methods that predict very large pockets (large Union). Thus, it is a suitable measure for comparing the overall performance of the pocket extraction methods. For example, though the pockets of *PocketFinder*$^+$ with the *Threshold*($mean$+1σ) extraction method have very high recall (0.927), its Jaccard coefficient is very low, because the predicted pockets have a very large average volume (42.6x more than the ligands). For each grid type, our *Search* pocket extraction method predicts pockets with volumes close to the actual ligand volume and obtains the best Jaccard coefficient. As a result, we use *Search* in *ConCavity* and our implementations of previous grid based methods.

Residue Mapping

The third step of our process uses the extracted set of pockets to generate ligand-binding predictions for residues. Our goal is to score every residue based on its relationship to the extracted pockets

such that residues with higher scores are more likely to bind ligands. This goal is more ambitious than that of previous residue mapping approaches which have sought only to identify the residues associated with predicted pockets.

Perhaps the simplest and most common previous method is to mark all residues within some distance threshold, d, of any pocket as binding (e.g., score = 1) and the rest as not binding (e.g., score = 0) [99]. We call this method "*Dist-01*." Both the pocket surface and geometric center have been taken as the reference point previously; we use the pocket surface in *Dist-01*. This approach ignores all local information about the predicted pockets. Two related methods have incorporated attributes of predicted pockets into *Dist-01*. The first assigns near-pocket residues scores that reflect the size of the closest pocket ("*Dist-Size*") [82]; residues near the largest pocket receive the highest score and so on. A similar approach uses the average conservation of all residues near the pocket ("*Dist-Cons*") [82] to rank the pockets and assign rank-based scores to residues.

In *ConCavity*, our goal is to assign scores to residues based on their likelihood of binding a ligand. We use the original grid values (which reflect the predicted likelihood of a ligand at every point in space) to weight the scores assigned to nearby residues. Starting with the grid values within the set of extracted pockets, we blur that grid with a Gaussian filter (σ = 4Å), and then assign to every residue the maximum grid value evaluated at the location of any of its atoms. This method, which we call "*Blur*," assigns residues in the same pocket different scores, since some residues are in the middle of a binding site, next to the part of a pocket with highest grid values, while others are at the fringe of a site, near marginal grid values. The score assigned by *Blur* reflects these differences in the likelihood that an individual residue is ligand binding.

In contrast to *Blur*, none of the previous residue extraction methods give different scores to residues in the same pocket. For comparison, we developed a version of the *Dist* strategy that (like *Blur*) considers the original grid values. *Dist-Raw* simply assigns to each residue within d of a pocket, the value of the nearest pocket grid point.

Performance

We analyze the performance of these residue mapping approaches by comparing their PR-AUC on the task of predicting ligand binding residues as defined in the LigASite apo dataset. In each case we start with the same grid of extracted pockets and apply a different residue mapping algorithm. We consider all residue mapping strategies on three different pocket grids: *ConCavity*$_L$, *ConCavity*$_S$, and *ConCavity*$_P$. For all *Dist* approaches, we set d to 5Å, and for *Dist-Cons* we consider the conservation of all residues within 8Å of the pocket (as in [82]).

Mapping Method	Pocket Grid Source $ConCavity_L$	$ConCavity_P$	$ConCavity_S$
Blur	**0.606**	**0.598**	**0.587**
Dist-Raw	0.453	0.547	0.468
Dist-Size	0.434	0.483	0.461
Dist-Cons	0.411	0.465	0.423
Dist-01	0.392	0.452	0.393

Table 4.7: Comparison of residue mapping strategies. For three grids of predicted pockets, we applied each residue mapping algorithm and report here the PR-AUC for identifying ligand binding residues in the LigASite apo dataset. *Blur* achieves the best performance for each grid.

The results presented in Table 4.7 demonstrate that *Blur* provides better performance for each grid type than all versions of previous residue mapping approaches. Thus, we use *Blur* in *ConCavity* and our implementation of previous ligand binding site prediction algorithms. The two methods that assign residues scores based on the values of nearby grid points (*Blur* and *Dist-Raw*) provide better performance in each case than those that assign all residues in a pocket the same score based on a global property of the pocket (*Dist-Size* and *Dist-Cons*). This suggests that the local environment around residues should be considered when predicting binding sites.

4.4.2 Previous Methods

We have compared *ConCavity* to several methods for ligand binding site prediction. Many of these methods lack publicly accessible implementations, and those that are available output different representations of predicted pockets and residues. In this section, we describe of how we generate predictions for all previous methods considered in our evaluations. In some cases we have completely reimplemented strategies and in others we have post-processed the output of existing implementations. As mentioned earlier, a "+" appended to the method name indicates that it is (at least in part) our implementation, e.g., *Ligsite*$^+$.

Ligsite$^+$, *Surfnet*$^+$, and *Pocketfinder*$^+$

We developed new implementations of the *Ligsite*, *Surfnet*, and *Pocketfinder* methods for grid generation. This was necessary to allow us to fully integrate sequence conservation with these methods. However, it also enabled us to investigate the the effect of different pocket extraction and residue mapping algorithms on overall performance.

By default, we use *Search* to extract pockets and *Blur* to map to residues for *Ligsite*$^+$, *Surfnet*$^+$,

Name	Prediction Algorithm Steps			Post-processing
	Grid Creation	Pocket Ext.	Res. Mapping	
ConCavity$_L$	*Ligsite+Cons*	*Search*	*Blur*	-
ConCavity$_P$	*PocketFinder+Cons*	*Search*	*Blur*	-
ConCavity$_S$	*Surfnet+Cons*	*Search*	*Blur*	-
Ligsite$^+$	*Ligsite*	*Search*	*Blur*	-
PocketFinder$^+$	*PocketFinder*	*Search*	*Blur*	-
Surfnet$^+$	*Surfnet*	*Search*	*Blur*	-
LigsiteCS	http://gopubmed2.biotec.tu-dresden.de/pocket/			Residues Ranked by Pocket Rank
Q-SiteFinder	http://www.modelling.leeds.ac.uk/qsitefinder/			
CASTp	http://sts-fw.bioengr.uic.edu/castp/			
LigsiteCSC$^+$	http://gopubmed2.biotec.tu-dresden.de/pocket/			Residues Ranked by Pocket Conservation

Table 4.8: Implementation Details for Evaluated Methods. This table summarizes the details of each step of the ligand binding site prediction process for the methods we evaluate. The new *ConCavity* methods are based entirely on our code. We also developed new implementations of three previous methods (*Ligsite*$^+$, *PocketFinder*$^+$, and *Surfnet*$^+$). Predictions for the other previous methods were obtained from the listed publicly accessible web servers. These servers output sets of residues associated with predicted binding pockets. For inclusion in the residue prediction evaluation, the output of these servers was post-processed as specified. This step is not necessary for our methods, because *Blur* outputs ranked residue predictions. A "$^+$" appended to the method name indicates that it is based (at least in part) on our code. Implementation details of each algorithm are given in the text.

and *Pocketfinder*$^+$, because as was shown above, these approaches yield the best performance. Our implementations output representations of the predicted ligand binding pockets and ranked lists of contacting residues, so they can be included in both pocket and residue-based evaluations.

LigsiteCS, *Q-SiteFinder*, and *CASTp*

For our experiments, we generate binding site predictions using three publicly available web servers: *LigsiteCS* [82], *QSiteFinder* [99], and *CASTp* [95]. Each of these servers produces a list of predicted pockets represented by sets of residues. In each case, the residues do not have scores associated with them. Thus to include these methods in the ligand binding residue prediction evaluation, we must assign scores to the residues. We tried two approaches. The first assigns all predicted residues a score of one and all others a score of zero. The second ranks the residues by the highest ranking pocket to which they are assigned, i.e., all residues from the first predicted pocket are given a higher score than those from the second and so on. These approaches are similar to the residue mapping algorithms discussed in the *ConCavity* section above; however, those exact algorithms could not be applied here because the web servers do not provide representations of the full extent of predicted pockets. We found that residue ranking produces better results (data not shown), so we use this approach. We consider the default number of pockets predicted by each method: *LigsiteCS* returns three pockets; *Q-SiteFinder* returns ten pockets; and *CASTp* returns a variable number. The *Q-SiteFinder* web server would not accept proteins with more than 10,000 atoms, so 36 proteins from our dataset lack *Q-SiteFinder* scores.

LigsiteCS, *Q-SiteFinder*, and *CASTp* do not provide a representation of each predicted pocket's full extent, so they could not be included in the ligand location prediction evaluation.

LigsiteCSC$^+$

The *LigsiteCSC* method is an extension of *LigsiteCS* that uses the evolutionary sequence conservation of residues surrounding predicted pockets to reorder the pocket predictions. This feature on the *LigsiteCS* prediction server did not work for many proteins in our dataset, so we implemented our own version on top of the *LigsiteCS* results. For each pocket, we calculate the average conservation of all residues within 8Å of the pocket center. The JSD method on the HSSP alignments is used to produce the conservation scores. The top three pockets in terms of size are then ranked in terms of average conservation. This implementation follows the published description of *LigsiteCSC*, except for the use of JSD for conservation instead of ConSurf.

Jensen-Shannon Divergence

The Jensen-Shannon divergence (JSD) is used to represent the performance of evolutionary sequence conservation; it was recently shown to provide state-of-the-art performance on a range of functional site prediction tasks [5]. It compares the amino acid distribution observed in columns of a multiple sequence alignment of homologs to a background distribution. JSD scores range between zero and one. The code provided in Capra and Singh [5] with the default sequence weighting and gap penalty was used to score all alignments.

4.4.3 Data

The prediction methods described in this chapter take 3D structures and/or multiple sequence alignments as input. Protein structures were downloaded from the Protein Quaternary Structure (PQS) server [125]. Predicted quaternary structures were used (rather than the tertiary structures provided in PDB files) so as to consider pockets and protein-ligand contacts for proteins in their biologically active states. All alignments come from HSSP [46].

Ligand binding sites as defined by the non-redundant version of the LigASite dataset (v4.0) [100] were used to evaluate method predictions. Ligasite consists of 331 proteins with apo (unbound) structures, each having less than 25% sequence identity with any other protein in the set. Each apo structure has at least one associated holo (bound) structure in which biologically relevant ligands are identified in order to define ligand binding residues and map them to the apo structure. If multiple holo structures are available for the protein, the sets of contacting residues are combined. The average number of holo structures for each apo structure is 2.665, and the maximum for any single structure is 30. The average chain length is 274 residues with a minimum of 59 and a maximum of 1023. The average number of positives—sites contacting a biologically relevant ligand—per chain is 25 residues (about 11% of the chain). The apo dataset includes many proteins with multiple chains; the average number of chains per protein is 2.49. The chain distribution is: 1 chain: 132 proteins, 2 chains: 112 proteins, 3 chains: 22 proteins, 4 chains: 36 proteins, 5+ chains: 29 proteins.

The drug dataset comes from a set of 100 non-redundant 3D structures selected by [123]. This set contains a diverse set of high-quality structures (resolution < 3Å) with drug or drug-like molecules (molecular weight between 200 and 600, and 1-12 rotatable bonds) bound. Structure 1LY7 has been removed from the PDB, so we consider the 99 remaining structures.

The catalytic site annotations were taken from version 2.2.9 of the Catalytic Site Atlas [45]. There are 130 proteins in the LigASite apo dataset with entries in the Catalytic Site Atlas. These

proteins have an average of 2.9 catalytic sites per chain (just over 1% of all residues in the chain).

4.4.4 Evaluation

Ligand binding pocket predictions are represented by non-zero values in a regular 3D grid around the protein. These predictions are evaluated on the pocket level by computing their volume and overlap with known ligands, and on the grid level by analyzing how well the grid scores rank grid points that overlap ligand atoms. We use a grid with the rasterized ligand atoms from the PQS structure to find the intersection and union of the ligands and predictions. From these, we compute the over-prediction factor (Prediction Volume / Ligand Volume), precision (Intersection Volume / Prediction Volume), recall (Intersection Volume / Ligand Volume), and Jaccard coefficient (Intersection Volume / Union Volume). We use precision-recall (PR) curves, which compare precision (TP / TP + FP) on the y-axis with recall (TP / TP + FN) on the x-axis, to evaluate the ability of each method to predict whether a ligand atom is present at a grid point. We consider grid points that overlap a ligand atom as positives. To construct the PR curve, we calculate the precision and recall at each cutoff of the grid values in the pocket prediction grid.

For the residue-based evaluation, we consider how well each method's residue scores identify ligand binding residues. Positives are those residues in contact with a ligand as defined by LigASite database. PR curves were made by calculating, for each chain, the precision and recall at each position on the ranked list of residue scores. The PR curves were constructed similarly for the catalytic site analysis, but positives were defined as those residues listed in the Catalytic Site Atlas.

To construct the summary PR curves for each prediction method, curves were created for each chain in each structure, and all the PR curves for a method were averaged first within the structure and then across all structures. We used the method and code of Davis and Goadrich [78] for calculating the area under the PR curve (PR-AUC).

Chapter 5

Conclusion

Analyzing protein sequence and structure data holds considerable promise for increasing our understanding of how proteins use simple amino acid building blocks to accomplish their vast array of functions. In this dissertation, we have described three projects (including prediction tools and datasets) that greatly improve our ability to identify and characterize the residues involved in a range of functional processes.

In the first, we developed a new, fast approach to the estimation of evolutionary sequence conservation and evaluated it against a representative set of existing methods on three large datasets of functional sites. Our simple information theoretic approach, based on the insight that sites under pressure will have amino acid distributions further from an estimated background distribution, outperforms many more sophisticated methods in each of three real-world prediction tasks. We also found considerable differences in the ability of all conservation methods to predict different types of functional sites. Catalytic and ligand binding sites are usually highly conserved, while identifying protein-protein interfaces causes considerable problems for all methods.

In the second project, we considered a type of functional site that evolutionary sequence conservation provides little help in identifying: residues that determine molecular substrate specificity (SDPs). Prediction of these residues has been crippled by a lack of reliable data. We combined several bioinformatics resources, including protein sequence, structure, and experimental data, to construct a large database of likely SDPs in enzymes. This database enabled us to characterize SDPs in terms of their physicochemical and evolutionary properties with respect to those of other sets of functional sites. It also equipped us to evaluate the performance of existing sequence-based SDP prediction methods and to demonstrate the superior results of our *GroupSim* method. The

techniques used to create the dataset can themselves be seen as an SDP prediction algorithm that combine several lines of evidence.

Finally, we directly integrated sequence-level information about evolutionary conservation with structure-based predictions of protein surface cavities to accurately identify ligand binding pockets and residues. Prior to our work, sequence and structure based methods for this problem had been largely separate. In contrast to previous efforts to combine them, our method, *ConCavity*, considers both levels of information simultaneously rather than combining the results of conservation analysis with structure analysis after performing each step in isolation. Not only does our method yield significantly better predictions than either approach alone, it also provides insights into the relationship of sequence conservation and structural features. It is difficult to distinguish surface pockets in which ligands bind from those that do not based on structural criteria alone, and it is similarly difficult to determine whether or not an amino acid residue is conserved because it is involved in ligand binding or some other important aspect of the protein's function. However, we find that residues that are in pockets with a high degree of sequence conservation are likely involved in ligand binding.

In addition to producing useful prediction tools and datasets, the work presented in this thesis highlights the importance and power of comprehensive, large-scale evaluation frameworks that are grounded in the real-world settings in which prediction methods are applied. Our evaluation of algorithms for predicting evolutionary conservation provide a dramatic example of this. Over the past 20 years, at least 20 different methods for estimating evolutionary conservation or rate of evolution have been developed [30]. Previous evaluation of these methods was usually small scale, ad hoc, and often tied to the developers' intuitions about what the results of a conservation measure "should" look like on the data of interest. By grounding our evaluation in the prediction of three types of functionally important residues for which we could build large datasets, we were able to make sound statements about what methods worked and in what settings. Surprisingly, a number of sophisticated methods intended to improve on the simple Shannon Entropy approach failed to perform at its level. Similar patterns were observed in our evaluation of SDP prediction methods. Simple, fast methods often did as well as those based on more complex models. This suggests that more sophisticated methods often fail to capture much about the complexities of the biological systems under study beyond relatively basic intuitive relationships, e.g., conservation within groups that bind the same ligand and difference between them indicates a role in specificity. We believe that this is largely the result of the lack of reliable, quantitative specifications of the objects being predicted. Complex methods and models have often been based on assumptions and intuitions of

domain experts. These experience-based hypothesis are useful, but they must be rigorously evaluated before being incorporated into published prediction methods. Our datasets provide an important step towards this goal. In contrast to many previous approaches, the algorithms introduced in this dissertation were found to perform well in realistic settings.

As suggested by our results using evolutionary sequence conservation, we suspect that identifying some types of functional sites will prove to be more difficult than others. Different data sources may be better suited to certain contexts. Our work has demonstrated the power of directly combining protein sequence and structure data in a variety of prediction tasks. It enabled us to build a large dataset of positions likely involved in determining substrate specificity and resulted in significant improvement in the prediction of ligand binding pockets and residues.

Future advances in the prediction of protein functional sites will likely be driven by the integration of additional types of data about protein sequence, structure, evolution, and cellular context. Several recent approaches have combined a large number of sequence, structure, and evolutionary features to predict catalytic sites using machine learning algorithms [26, 29, 85, 115]. Similar attempts have been made recently for PPI sites [126] and predicting residues that bind particular ligands [117–119]. These studies have all found that, out of the hundreds considered, a very small number of features provide the vast majority of the predictive power. Evolutionary sequence conservation and a few structural attributes like pocket membership are usually the most informative predictors. The improvement provided by these machine learning approaches over direct approaches is modest at best. We find the principled prediction of functional sites based on direct biologically observed relationships between features, e.g., the observation that ligands are much more likely to bind in evolutionarily conserved pockets than non-conserved pockets, to be both more satisfying and useful in practice. Going forward, integrating the assignment of functions to protein residues into a comprehensive probabilistic framework is an appealing goal, but we believe insights based on biological relationships should provide the foundation rather than current "kitchen sink" machine learning approaches.

Probabilistic approaches to integrating heterogeneous data to predict protein function and interactions [127–130] could provide a model for incorporating the many disparate evidence types and functional roles a residue can have into a coherent predictive system. However, there are several challenges to achieving this goal. We lack a consistent language or ontology (like the Gene Ontology [131] for protein function) with which to describe the functional roles of specific protein residues. Attempts to characterize and classify catalytic sites [44, 45] provide a step in this direction. The new Sequence Ontology of the Protein Feature Ontology project [132] may ultimately provide the

necessary framework, but there is still significant work to be done to define the scope and boundaries of possible residue functions. In addition, even with a coherent framework in which to discuss residue function, many functional roles would lack a large number of experimentally determined positive examples, thus making reliable training and inference difficult.

In conclusion, this dissertation describes three projects that improve our ability to identify functional sites in proteins from sequence and structural data. They are steps toward the ultimate goal of completely characterizing a protein's molecular functions and the residues involved in carrying them out. Because of both the rapid pace at which the amount of informative sequence and structure data are being produced and the ingenuity of the computational biologists who store and process this information, we are optimistic about continued progress towards this goal.

Bibliography

[1] K Liolios, N Tavernarakis, P Hugenholtz, and NC Kyrpides. The genomes on line database (GOLD) v.2: a monitor of genome projects worldwide. *Nucleic Acids Res*, 34:D332–D334, 2006.

[2] EW Sayers, T Barrett, DA Benson, SH Bryant, K Canese, V Chetvernin, DM Church, M DiCuccio, R Edgar, S Federhen, M Feolo, LY Geer, W Helmberg, Y Kapustin, D Landsman, DJ Lipman, TL Madden, DR Maglott, V Miller, I Mizrachi, J Ostell, KD Pruitt, GD Schuler, E Sequeira, ST Sherry, M Shumway, K Sirotkin, A Souvorov, G Starchenko, TA Tatusova, L Wagner, E Yaschenko, and J Ye. Database resources of the national center for biotechnology information. *Nucleic Acids Res*, 37:D5–D15, 2009.

[3] H Berman, J Westbrook, Z Feng, G Gilliland, T Bhat, H Weissig, I Shindyalov, and P Bourne. The protein data bank. *Nucleic Acids Res.*, 28:235–242, 2000.

[4] M Levitt. Growth of novel protein structural data. *PNAS*, 104(9):3183–3188, 2007.

[5] JA Capra and M Singh. Predicting functionally important residues from sequence conservation. *Bioinf*, 23:1875–1882, 2007.

[6] JA Capra and M Singh. Characterization and prediction of residues determining protein functional specificity. *Bioinf*, 24:1473–1480, 2008.

[7] TJ Magliery and L Regan. Sequence variation in ligand binding sites in proteins. *BMC Bioinf*, 6:240, 2005.

[8] S Liang, C Zhang, S Liu, and Y Zhou. Protein binding site prediction using and empirical scoring function. *Nucleic Acids Res.*, 34(13):3698–3707, 2006.

[9] DR Caffrey, S Somaroo, JD Hughes, J Mintseris, and ES Huang. Are protein-protein interfaces more conserved in sequence than the rest of the protein surface? *Prot Sci*, 13:190–202, 2004.

[10] M Guharoy and P Chakrabarti. Conservation and relative importance of residues across protein-protein interfaces. *PNAS*, 102(43):15447–15452, 2005.

[11] J Mintseris and Z Weng. Structure, function, and evolution of transient and obligate protein-protein interactions. *PNAS*, 102(31):10930–10935, 2005.

[12] S Karlin and L Brocchieri. Evolutionary conservation of RecA genes in relation to protein structure and function. *J. Bacteriology*, 178:1881–1894, 1996.

[13] W Valdar and J Thornton. Conservation helps to identify biologically relevant crystal contacts. *J. Mol. Biol.*, 313:399–416, 2001.

[14] O Schueler-Furman and D Baker. Conserved residue clustering and protein structure prediction. *Proteins*, 52:225–235, 2003.

[15] O Lichtarge, HR Bourne, and FE Cohen. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*, 257:342–358, 1996.

[16] S Hannenhalli and R Russell. Analaysis and prediction of functional sub-types from protein sequence alignments. *J. Mol. Biol.*, 303:61–76, 2000.

[17] OV Kalinina, AA Mironov, MS Gelfand, and AB Rakhmaninova. Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. *Prot Sci*, 13:443–456, 2004.

[18] A Panchenko, F Konrashov, and S Bryant. Prediction of functional sites by analysis of sequence and structure conservation. *Protein Sci.*, 13:884–892, 2004.

[19] M Landau, I Mayrose, Y Rosenberg, Y Glaser, E Martz, T Pupko, and N Ben-Tal. ConSurf 2005: the projection of evolutionary conservation scores of residues on protein structures. *Nucleic Acids Res*, 33:W299–W302, 2005.

[20] S Jones and JM Thornton. Searching for functional sites in protein structures. *Curr. Opin. Chem. Biol.*, 8(1):3–7, 2004.

[21] A Wallace, N Borkakoti, and J Thornton. TESS: A geometric hasing algorithm for deriving 3D coordinate templates for searching structural databases. *Protein Sci.*, 6:2308–2323, 1997.

[22] J Fetrow and J Skolnick. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. *J. Mol. Biol.*, 281:949–968, 1998.

[23] A Stark and R Russell. Annotation in three dimensions. PINTS: patterns in non-homologous tertiary structures. *Nucleic Acids Res.*, 31:3314–3344, 2003.

[24] MJ Ondrechen, JG Clifton, and D Ringe. THEMATICS: a simple computational predictor of enzyme function from structure. *PNAS*, 98(22):12473–12478, 2001.

[25] AH Elcock. Prediction of functionally important residues based solely on the computed energetics of protein structure. *J Mol Bio*, 312:885–896, 2001.

[26] A Gutteridge, G Bartlett, and J Thornton. Using a neural network and spatial clustering to predict the location of active sites in enzymes. *Journal of Molecular Biology*, 330:719:734, 2003.

[27] A Bordner and R Abagyan. Statistical analysis and prediction of protein-protein interfaces. *Proteins*, 60:353–366, 2005.

[28] J Chung, W Wang, and P Bourne. Exploiting sequence and structure homologs to identify protein-protein binding sites. *Proteins*, 62:630–640, 2006.

[29] N. Petrova and C. Wu. Prediction of catalytic residues using support vector machine with selected protein sequence and structural properties. *BMC Bioinf*, 7:312, 2006.

[30] W. Valdar. Scoring residue conservation. *Proteins*, 48:227–241, 2002.

[31] K Wang and R Samudrala. Incorporating background frequency improves entropy-based residue conservation measures. *BMC Bioinformatics*, 7(385), 2006.

[32] R Durbin, S Eddy, A Krogh, and G Mitchison. *Biological Sequence Analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[33] R Williamson. Information theory analysis of the relationship between primary sequence structure and ligand recognition among a class of facilitated transporters. *J. Theor. Biol.*, 174:179–188, 1995.

[34] L Mirny and E Shakhnovich. Universally conserved positions in protein folds: reading evolutionary signals about stability, folding, kinetics, and function. *J. Mol. Biol.*, 291:177–196, 1999.

[35] T Cover and J Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

[36] I Mayrose, D Graur, N Ben-Tal, and T Pupko. Comparison of site-specific rate-inference methods: Bayesian methods are superior. *Mol Biol Evol*, 21:1781–1791, 2004.

[37] S Henikoff and JG Henikoff. Position-based sequence weights. *J. Mol. Biol.*, 243:574–578, 1994.

[38] S Sander and R Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9:56–68, 1991.

[39] P Shenkin, B Erman, and Mastrandrea L. Information-theoretical entropy as a measure of sequence variability. *Proteins*, 11:297–313, 1991.

[40] M Nielsen and I Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[41] S Henikoff and JG Henikoff. Amino acid substitution matrices from protein blocks. *PNAS*, 89:10915–10919, 1992.

[42] J Lin. Divergence measures based on the shannon entropy. *IEEE Trans. on Inf. Theory*, 37 (1):145–151, 1991.

[43] G Yona and M Levitt. Within the twilight zone: A sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.*, 315:1257–1275, 2002.

[44] G Bartlett, C Porter, N Borkakoti, and J Thornton. Analysis of catalytic residues in enzyme active sites. *J. Mol. Biol.*, 324:105–121, 2002.

[45] C Porter, G Bartlett, and J Thornton. The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res.*, 32:D129–D133, 2004.

[46] C Dodge, R Schneider, and C Sander. The HSSP database of protein structure-sequence alignments and family profiles. *Nucleic Acids Res*, 26(1):313–315, 1998.

[47] E Webb. *Enzyme Nomenclature. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology.* Academic Press, New York, 1992.

[48] A Bairoch. The ENYZME database in 2000. *Nucleic Acids Res.*, 28:304–305, 2000.

[49] R Laskowski, V Chistyakov, and J Thornton. PDBsum more: new summaries and analyses of the known 3d structures of proteins and nucleic acids. *Nucleic Acids Res.*, 33:D266–D268, 2005.

[50] B Lee and F Richards. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.*, 55:379–400, 1971.

[51] S Hubbard and J Thornton. NACCESS. Computer Program, 1993.

[52] L Mirny and M Gelfand. Using orthologous and paralogous proteins to identify specificity-determining residues in bacterial transcription factors. *J. Mol. Biol.*, 321:7–20, 2002.

[53] CD Livingstone and GJ Barton. Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. *Comput. Appl. Biosci*, 9(6):745–756, 1993.

[54] G Casari, C Sander, and A Valencia. A method to predict functional residues in proteins. *Nat. Struct. Biol.*, 2:171–178, 1995.

[55] J Donald and E Shakhnovich. Determining functional specificity from protein sequences. *Bioinformatics*, 21(11):2629–35, 2005.

[56] K Mayer, S McCorkle, and J Shanklin. Linking enzyme sequence to function using conserved property difference locator to identify and annotate positions likely to control specific functionality. *BMC Bioinformatics*, 6:284, 2005.

[57] K Ye, EW Lameijer, M Beukers, and A Ijzerman. A two-entropies analysis to identify functional positions in the transmembrane region of class a G-protein-coupled receptors. *Proteins: Strut., Func., and Bioinf.*, 63:1018–1030, 2006.

[58] W Pirovano, KA Feenstra, and J Heringa. Sequence comparison by sequence harmony identifies subtype-specific functional sites. *Nucleic Acids Research*, 34(22):6540–6548, 2006.

[59] A del Sol Mesa, F Pazos, and A Valencia. Automatic methods for predicting functionally important residues. *Journal of Molecular Biology*, 326:1289–1302, 2003.

[60] I Mihalek, I Res, and O Lichtarge. A family of evolution–entropy hybrid methods for ranking protein residues by importance. *J Mol Bio*, 336(5):1265–1282, 2004.

[61] I Wallace and D Higgins. Supervised multivariate analysis of sequence groups to identify specificity determining residues. *BMC Bioinformatics*, 8:135, 2007.

[62] F Pazos, A Rausell, and A Valencia. Phylogeny-independent detection of functional residues. *Bioinformatics*, 22(12):1440–1448, 2006.

[63] G Yu, B Park, P Chandramohan, R Munavalli, A Geist, and N Samatova. In silico discovery of enzyme-substrate specificity-determining residue clusters. *J. Mol. Biol.*, 352:1105–1117, 2005.

[64] S Chakrabarti, S Bryant, and A Panchenko. Functional specificity lies within the properties and evolutionary changes of amino acids. *J. Mol. Biol.*, 373:801–10, 2007.

[65] DM Kristensen, RM Ward, AM Lisewski, S Erdin, BY Chen, VY Fofanov, M Kimmel, LE Kavraki, and O Lichtarge. Prediction of enzyme function based on 3D templates of evolutionarily important amino acids. *BMC Bioinformatics*, 9(17), 2008.

[66] J Manning, E Jefferson, and G Barton. The contrasting properties of conservation and correlated phylogeny in protein functional residue prediction. *BMC Bioinformatics*, 9:51, 2008.

[67] J Pei, W Cai, L Kinch, and N Grishin. Prediction of functional specificity determinants from protein sequences using log-likelihood ratios. *Bioinformatics*, 22(2):164–171, 2006.

[68] P Marttinen, J Corander, P Törönen, and L Holm. Bayesian search of functionally divergent protein subgroups and their function specific residues. *Bioinformatics*, 22(20):2466–2474, 2006.

[69] B Reva, Y Antipin, and C Sander. Determinants of protein function revealed by combinatorial entropy optimization. *Genome Biology*, 8:R232, 2007.

[70] D Brown, N Krishnamurthy, and K Sjolander. Automated protein subfamily identification and classification. *PLoS Computional Biology*, 3(8):e160, 2007.

[71] A Bairoch, R Apweiler, C Wu, W Barker, B Boeckmann, S Ferro, E Gasteiger, H Huang, R Lopez, M Magrane, M Martin, D Natale, C O'Donovan, N Redaschi, and L Yeh. The universal protein resource (UniProt). *Nucleic Acids Res.*, 33:D154–59, 2005.

[72] RD Finn, J Mistry, B Schuster-Bockler, S Griffiths-Jones, V Hollich, T Lassman, S Moxon, M Marshall, A Khanna, R Durbin, S Eddy, E Sonnhammer, and A Bateman. Pfam: Clans, web tools, and services. *Nucleic Acids Res.*, 34(D247-D251), 2006.

[73] S Brown, J Gerlt, J Seffernick, and P Babbitt. A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biology*, 7:R8, 2006.

[74] SF Altschul, W Gish, W Miller, EW Myers, and DJ Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–10, 1990.

[75] CB Do, MSP Mahabhashyam, M Brudno, and S Batzoglou. Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15:330–40, 2005.

[76] W Kabasch and C Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–637, 1983.

[77] J Kyte and RF Doolittle. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, 157:105–32, 1982.

[78] J Davis and M Goadrich. The relationship between precision-recall and ROC curves. *Proceedings of 23rd International Conference on Machine Learning*, 23:233–240, 2006.

[79] R George, R Spriggs, G Bartlett, A Gutteridge, M MacArthur, C Porter, B Al-Lazikani, J Thornton, and M Swindells. Effective function annotation through catalytic residue conservation. *PNAS*, 102(35):12299–304, 2005.

[80] JD Fischer, CE Mayer, and J Soeding. Prediction of protein functional residues from sequence by probability density estimation. *Bioinf*, 24(5):613–620, 2008.

[81] AD McLachlan. Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551. *J. Mol. Biol.*, 61(2):409–24, 1971.

[82] B. Huang and M. Schroeder. LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct Bio*, 6:19, 2006.

[83] G Lopez, A Valencia, and ML Tress. firestar—prediction of functionally important residues using structural templates and alignment reliability. *Nucleic Acids Res*, 35:W573–W577, 2007.

[84] IB Kuznetsov, Z Gou, R Li, and S Hwang. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins: Stuct, Func, and Bioinf*, 64: 19–27, 2006.

[85] E. Youn, B. Peters, P. Radivojac, and S. Mooney. Evaluation of features for catalytic residue prediction in novel folds. *Prot Sci*, 16:216–226, 2007.

[86] Y Ofran and B Rost. Protein-protein interaction hotspots carved into sequences. *PLoS Comput Biol*, 3(7):e119, 2007.

[87] HX Zhou and S Qin. Interaction-site prediction for protein complexes: a critical assessment. *Bioinf*, 23(17):2203–2209, 2007.

[88] A del Sol Mesa, F Pazos, and A Valencia. Automatic methods for predicting functionally important residues. *J Mol Bio*, 326:1289–1302, 2003.

[89] D Levitt and L Banaszak. POCKET: A computer graphics method for identifying and displaying protein cavities and their surrounding amino acids. *J. Mol. Graphics*, 10:229–234, 1992.

[90] R Laskowski. SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. *J Mol Graph*, 12:323–330, 1995.

[91] KP Peters, J Fauck, and C Frömmel. The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. *J Mol Bio*, 256(1):201–213, 1996.

[92] M Hendlich, F Rippman, and G Barnickel. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *J Mol Graph Model*, 15(6):359–363, 1997.

[93] J Liang, H Edelsbrunner, and C Woodward. Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science*, 7: 1884–1897, 1998.

[94] GP Brady Jr and PFW Stouten. Fast prediction and visualization of protein binding pockets with PASS. *J Comp-Aided Mol Design*, 14:383–401, 2000.

[95] J Dundas, Z Ouyang, J Tseng, A Binkowski, Y Turpaz, and J Liang. CASTp: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic Acids Res*, 34:W116–W118, 2006.

[96] L. Xie and P. Bourne. A robust and efficient algorithm for the shape description of protein structures and its application in predicting ligand binding sites. *BMC Bioinf*, 8(Suppl. 4):S9, 2007.

[97] M. Weisel, E. Proschak, and G. Schneider. PocketPicker: analysis of ligand binding-sites with shape descriptors. *Chem Cen J*, 1:7, 2007.

[98] J An, M Totrov, and R Abagyan. Pocketome via comprehensive identification and classification of ligand binding envelopes. *Mol Cell Prot*, 4:752–761, 2005.

[99] A. Laurie and R. Jackson. Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. *Bioinf*, 21:1908–1916, 2005.

[100] BH Dessailly, MF Lensink, and SJ Wodak. LigASite: a database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Res*, Database:1–7, 2007.

[101] S Sankararaman and K Sjolander. INTREPID–INformation-theoretic TREe traversal for Protein functional site IDentification. *Bioinf*, 24(21):2445–2452, 2008.

[102] DB KC and DR Livesay. Improving position specific predictions of protein functional sites using phylogenetic motifs. *Bioinf*, 24:2308–2316, 2008.

[103] P Bate and J Warwicker. Enzyme/non-enzyme discrimination and prediction of enzyme active site location using charge-based methods. *J Mol Bio*, 340:263–276, 2004.

[104] J Ko, LF Murga, P Andre, H Yang, MJ Ondrechen, RJ Williams, A Agunwamba, and DE Budil. Statistical criteria for the identification of protein active sites using theoretical microscopic titration curves. *Proteins: Stuct, Func, and Bioinf*, 59:193–195, 2005.

[105] M. Brylinski and J. Skolnick. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *PNAS*, 105:129–134, 2008.

[106] I Halperin, H Wolfson, and R Nussinov. SiteLight: binding-site prediction using phage display libraries. *Prot Sci*, 12:1344–1359, 2003.

[107] G Amitai, A Shemesh, E Sitbon, M Shklar, D Netanely, I Venger, and S Pietrokovski. Network analysis of protein structures identifies functional residues. *J Mol Biol*, 344:1135–1146, 2004.

[108] H Yao, DM Kristensen, I Mihalek, ME Sowa, C Shaw, M Kimmel, L Kavraki, and O Lichtarge. An accurate, sensitive, and scalable method to identify functional sites in protein structures. *J Mol Bio*, 326:255–261, 2003.

[109] V. Chelliah, L. Chen, TL Blundell, and SC Lovell. Distinguishing structural and functional restraints in evolution in order to identify interaction sites. *J Mol Bio*, 342:1487–1504, 2004.

[110] G Cheng, B Qian, R Samudrala, and D Baker. Improvement in protein functional site prediction by distinguishing structural and functional constraints on protein family evolution using computational design. *Nucleic Acids Res*, 33:5861–5867, 2005.

[111] K Wang, JA Horst, G Cheng, DC Nickle, and R Samudrala. Protein meta-functional signatures from combining sequence, structure, evolution, and amino acid property information. *PLoS Comput Biol*, 4:9, 2008.

[112] BY Chen, VY Fofanov, DH Bryant, BD Dodson, DM Kristensen, AM Lisewski, M Kimmel, O Lichtarge, and LE Kavraki. The MASH pipeline for protein function prediction and an algorithm for the geometric refinement of 3d motifs. *J. Comp. Biol.*, 14(6):791–816, 2007.

[113] NJ Burgoyne and RM Jackson. Predicting protein interaction sites: binding hot-spots in protein-protein and protein-ligand interfaces. *Bioinf*, 22(11):1335–1342, 2006.

[114] S Yoon, JC Ebert, EY Chung, D DeMicheli, and RB Altman. Clustering protein environments for function prediction: finding PROSITE motifs in 3D. *BMC Bioinf*, 8(Suppl 4):S10, 2007.

[115] W Tong, Y Wei, LF Murga, MJ Ondrechen, and RJ Williams. Partial order optiumum likelihood (POOL): Maximum likelihood prediction of protein active site residues using 3D structure and sequence properties. *PLoS Comput Biol*, 5:1, 2009.

[116] M Nayal and B Honig. On the nature of cavities on protein surfaces: Application to the identification of drug-binding sites. *Proteins: Stuct, Func, and Bioinf*, 63:892–906, 2006.

[117] L Wei and RB Altman. Recognizing complex, asymmetric functional sites in protein structures using a bayesian scoring function. *J Bioinform Comput Biol.*, 1(1):119–138, 2003.

[118] AJ Bordner. Predicting small ligand binding sites in proteins using backbone structure. *Bioinf*, 24(24):2865–2871, 2008.

[119] JC Ebert and RB Altman. Robust recognition of zinc binding sites in proteins. *Prot Sci*, 17: 54–65, 2008.

[120] F. Glaser, R. Morris, R. Najmanovich, R. Laskowski, and J. Thornton. A method for localizing ligand binding pockets in protein structures. *Proteins: Stuct, Func, and Bioinf*, 62:479–488, 2006.

[121] R Najmanovich, J Kuttner, V Sobolev, and M Edelman. Side-chain flexibility in proteins upon ligand binding. *Proteins: Structure, Function, and Genetics*, 39:261–268, 2000.

[122] RA Laskowski, NM Luscombe, MB Swindells, and JM Thornton. Protein clefts in molecular recognition and function. *Prot Sci*, 5:2438–2452, 1996.

[123] E Perola, WP Walters, and PS Charifson. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. *Proteins: Stuct, Func, and Bioinf*, 56:235–249, 2004.

[124] GM Morris, DS Goodsell, RS Halliday, R Huey, WE Hart, RK Belew, and AJ Olson. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J Comp Chem*, 19(14):1639–1662, 1998.

[125] K Henrick and J Thornton. PQS: a protein quaternary structure file server. *Trends in Biochemical Sciences*, 23(9):358–361, 1998.

[126] Vlahoviček K Šikić M, Tomić S. Prediction of protein–protein interaction sites in sequences and 3d structures by random forests. *PLoS Comput Biol*, 5:1, 2009.

[127] OG Troyanskaya, K Dolinski, AB Owen, RB Altman, and D Botstein. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *PNAS*, 100(14):8348–8353, 2003.

[128] Ronald Jansen, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J. Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F. Greenblatt, and Mark Gersteein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302 (5644):449–453, 2003.

[129] Insuk Lee, Shailesh V. Date, Alex T. Adai, and Edward M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306(5701):1555–1558, 2004.

[130] CL Myers and OG Troyanskaya. Context-sensitive data integration and prediction of biological networks. *Bioinf*, 23(17):2322–2330, 2008.

[131] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.

[132] GA Reeves, K Eilbeck, M Magrane, C O'Donovan, L Montecchi-Palazzi, MA. Harris, S Orchard, RC Jimenez, A Prlic, TJP. Hubbard, H Hermjakob, and JM Thornton. The protein feature ontology: a tool for the unification of protein feature annotations. *Bioinf*, 24(2767-2772), 2008.