Surface Matching and Chemical Scoring to Detect Unrelated Proteins Binding Similar Small Molecules

By

Jeffrey Ryan Van Voorst

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Computer Science
Biochemistry and Molecular Biology

2011

ABSTRACT

Surface Matching and Chemical Scoring to Detect Unrelated Proteins Binding Similar Small Molecules

By

Jeffrey Ryan Van Voorst

How can one deduce if two clefts or pockets in different protein structures bind the same small molecule if there is no significant sequence or structural similarity between the proteins? Human pattern recognition, based on extensive structural biology or ligand design experience, is the best choice when the number of sites is small. However, to be able to scale to the thousands of structures in structural databases requires implementing that experience as computational method. The primary advantage of such a computational tool is to be able to focus human expertise on a much smaller set of enriched binding sites.

Although a number of tools have been developed for this purpose by many groups [53, 63, 89, 91, 94], to our knowledge, a basic hypothesis remains untested: two proteins that bind the same small molecule have binding sites with similar chemical and shape features, even when the proteins do not share significant sequence or structural similarity. A computational method to compare protein small molecule binding sites based on surface and chemical complementarity is proposed and implemented as a software package named SimSite3D. This method is protein structure based, does not rely on explicit protein sequence or main chain similarities, and does not require the alignment of atomic centers. It has been engineered to provide a detailed search of one fragment site versus a dataset of $\sim$ 13,000 full ligand sites in 2-4 hours (on one processor core).

Several contributions are presented in this dissertation. First, several examples are

presented where SimSite3D is able to find significant matches between binding sites that have similar ligand fragments bound but are unrelated in sequence or structure. Second, including the complementarity of binding site molecular surfaces helps to distinguish between sites that share a similar chemical motif, but do not necessarily bind the same molecule. Third, a number of clear examples are provided to illustrate the challenges in comparing binding sites which should be addressed in order for a binding site comparision method to gain widespread acceptance similar to that enjoyed by BLAST [3, 4]. Finally, an optimization method for addressing protein (and small molecule) flexibility in the context of binding site comparisons is presented, prototyped, and tested.

Throughout the work, computational models were chosen to strike a delicate balance between achieving sufficient accuracy of alignments, discriminating between accurate and poor alignments, and discriminating between similar and dissimilar sites. Each of these criteria is important. Due to the nature of the binding site comparison problem, each criterion presents a separate challenge and may require compromises to balance performance to achieve acceptable performance in all three categories.

At the present, the problem of addressing flexibility when comparing binding site surfaces has not been presented or published by any other research group. In fact, the problem of modeling flexibility to determine correspondences between binding sites is an untouched problem of great importance. Therefore, the final goal of this dissertation is to prototype and evaluate a method that uses inverse kinematics and gradient based optimization to optimize a given objective function subject to allowed protein motions encoded as stereochemical constraints. In particular, we seek to simultaenously maximize the surface and chemical complementarity of two closely aligned sites subject to directed changes in side chain dihedral angles.

This dissertation is dedicated to my family:
To Melissa, Brendan, Eliana, Kyla, and Keegan

ACKNOWLEDGMENTS

First and foremost I acknowledge God for giving me the ability, strength, and resolve to carry out the research and write this dissertation. Because, without Him we can do nothing.

I thank my wonderful and loving family for their support. Melissa, you have done more than your share of raising our children. Brendan, Eliana, Kyla, and Keegan thank you for your love you have shown and putting up with the time I spent writing this dissertation. Without you it would have been easier to give up. I am grateful for the love and support you have shown me over the last six years. Many sacrifices were made so that I could complete my research.

Leslie, you are a wonderful mentor and an excellent scientist. I am grateful for your cheerful outlook, ability to keep me on task, and flexibility in my scheduling. It was a pleasure working in your lab and with your group. What many people cannot understand is that there was rarely a day in the five years in your lab that I did not want to continue our research projects.

George, I am thankful for your support and desire for me to write a computer science dissertation. I had many fruitful discussions with you. It has been very useful to remind me about presenting ideas that computer scientists can understand.

Yiying, it has been my pleasure to speak with you about the mathematical and technical aspects of my research. In particular, it is, at times, refreshing for me to talk to someone who understands math concepts and is more adept at formulating mathematical constructs than I am. I am especially indebted to you for your input and patience in talking through the optimization methods and the inverse kinematics approach used in ArtSurf.

Profs. Garavito and Esfahanian (and Leslie, George, and Yiying), I am thankful for you willingness to serve on my dissertation committee. Your input on the direction of my

research, and suggested edits for my dissertation were well received.

Matt, Leann, Maria, Chetan, Sandeep, and Anjali, you were all wonderful people to be around each day. Furthermore, you helped me learn most of the biochemistry that I know today, and you all were very helpful whenever I had questions.

Chelsea, Rachel, and Johnney, you helped to build the datasets used to develop and test SimSite3D. You were hard working, and without your work I would not have been able to present the results listed in this dissertation. Also, your input on making SimSite3D more user friendly is appreciated.

Barry, you were instrumental in getting SimSite3D (ASCbase) installed globally at Pfizer. Without your help, it would probably be languishing on an install disk under a heap of dust. Furthermore, your offering me a postdoc with a deadline helped to keep me from dragging my feet as much as I tend to.

Without the financial support from numerous sources I would have been unable to attend Michigan State University, perform my research, or write this dissertation. The majority of the funding was provide by a generous grant from Pfizer. A number of other sources includes the MSU Quantitative Biology Initiative, the MSU College of Engineering, the MSU Department of Computer Science, the National Science Foundation, and a Dissertation Completion Fellowship.

Last of all, I cannot forget about my experience working in a factory after obtaining a Masters degree in math. That experience reinforced my desire to pursue science and redouble my efforts to obtain a Ph.D. Therefore, I acknowledge your hand in this work, Access Business Group.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

The motivations for this dissertation include the ability to effectively mine protein structure datasets to discover similar binding sites in protein structures. This searching can be used to pose candidate small molecules or functional groups that are likely to bind protein structures or pockets with unknown function. In addition, the partial matching and flexible matching nature of comparing binding sites is of a general interest to scientists in the computer vision and computational geometry communities. Techniques for flexible surface matching exist in character animation, face recognition, and many other applications. In this dissertation, techniques from such fields have been adapted to be applicable to compare protein binding sites. It is expected that insights and knowledge gained from comparing proteins might be applicable in other areas such as full three dimensional matching and medical imaging.

1.1.1 The Need for a Binding Site Comparison Tool

Many proteins, and by extension protein networks and biological processes, are affected by interactions with specific small molecules. Understanding the basis and mechanism

of protein small-molecule interactions is crucial for drug discovery and design because most drugs work by enhancing or reducing the activity of one or more proteins. In order to gain a better understanding of proteins, structural genomics initiatives have been put forward to encourage the experimental solving of novel protein structures [59]. Because a relatively large number of novel structures are solved each year, automated methods to mine the datasets of existing protein structures for features that the novel proteins share with better studied proteins are important. The drug design community is especially interested in chemical and shape patterns across protein folds. However, in many instances, the binding sites and the biologically relevant small molecules that interact with proteins from structural genomics are unknown. Thus, a computational tool that compares potential binding sites against a dataset of proteins that have small molecules bound can be useful to propose candidate small molecules for proteins with unknown function.

As an example, suppose there exists a novel protein, called Protein A, that protein biochemists seek to understand. A commonly used technique is to search the known protein sequence space for a Protein B whose sequence is significantly similar to Protein A's sequence. The goal is to find that Protein B does exist, and that Protein B has been already studied. Thereby, one can infer features of Protein A based on conserved features between Protein A and Protein B. Other techniques to find proteins related to A include protein structure based search tools and experts looking at experimentally resolved protein structures in Protein A's structural fold. All of these tools are restricted to proteins with significantly similar sequence or structure. However, in many cases, there exist sets of proteins that can bind the same small molecules (e.g. ATP), such no two proteins in a set have significant pairwise sequence or structural similarity.

From a protein small-molecule interaction point of view, researchers are interested in all folds of proteins that can bind the same small molecule. For this reason, a number of tools have been developed that can compare the protein small-molecule binding sites of any two proteins. However, the journal articles and the previously existing tools have

shown little progress in addressing the problem of finding similar binding sites in otherwise unrelated proteins. Therefore, there exists a need for a non-sequence non-mainchain structure-based methodology to compare binding sites from any two proteins and to provide a ranking of a query binding site versus a dataset of binding sites reflecting their likelihood of binding the same or very similar molecules.

1.1.2 Addressing the 3D Partial Matching Problem

A relatively common problem in object recognition is to find the best match between a partial scan or object representation and each larger or full object in a dataset. Examples of partial matching include finding the best match for a partial fingerprint, finding the best match for a partial face scan, and finding the best match for a small molecule fragment binding site. Partial matching tends to be more challenging and computationally expensive than full matching since, in general, the heuristics used for object matching fail for partial matching. The reason for this failing is heuristics for full matching generally exploit global topological features of objects, but in the case of partial matching, a number of the features may be missing. The missing global features, in the partial matching case, can make it difficult to consistently avoid false negatives and false positives when using such features.

The partial matching problem is, in general, approached in two ways. One method is to compute a number of candidate alignments between the partial object and a full object (one such method is RANSAC [34]). This method is helpful because the relative positioning of feature points might be conserved between the compared objects. On the other hand, since a number of candidate alignments are considered, the runtime of such partial matching techniques can be longer than the comparison of two complete objects.

Another popular object recognition method is to compute transformation invariant features such as points of maximal local curvature [84]. One then considers the distances between the feature points in both objects to determine if the partial object is consistent

with the larger object. Transformation invariant features do not necessarily perform well in the partial matching setting as features of a partial object may be consistent with those of the larger object, but the relative placement of the features as a whole may differ.

Unfortunately, many of the published partial matching methods are verified on distinct object parts that are rigid and closed objects such as animal legs, human heads, and plane wings [71]. To our knowledge, protein small molecule binding sites do not have such global features as a ”leg” or ”wing”. In addition, many such methods take care to not allow intersections, but protein molecular surfaces are akin to metaballs [15]. That is: if two protein atoms are sufficiently close, they are modeled as having their surfaces joined as though they were two cohesive objects that are blended together in a distance dependent manner. Therefore, although curvature has been used to align protein surfaces, the stability of points of maximal local curvature is unknown in the context of partial matching.

1.2 Overview: Contributions to Science

A major objective of this thesis is to test the hypothesis that the binding sites of proteins that bind similar small molecules exhibit sufficiently similar features such that an automated method can recognize and group them according to their surface and chemical similarities. This objective is addressed by contributions to the state of the art in computational methods to compare protein small-molecule [1] binding sites. Since protein structures are 3D objects with their shape and function defined by the packing of a number of small, flexible and linked building blocks, applicable computer vision and computational geometry techniques are adjusted and applied to address the binding site comparison problem. The techniques used for binding site comparisons differ in details and implementation from many traditional computer vision methods because protein structures

1 In protein small-molecule settings, the small molecule is many times called a ligand.

are fully 3D objects and change of scale is not an issue with protein structures. The initial method to compare binding sites is enhanced by including surface and chemistry matching to address the problem of similar binding sites. Because proteins are flexible, due to relative motions within and between the building blocks, a flexible refinement method is developed and implemented to more consistently compare and contrast binding sites. I have implemented the binding site comparison methods as a software package named SimSite3D and extensively tested the methods on a number of challenging datasets. The results show that binding site chemical and shape features are necessary to compare binding sites from proteins that do not have significantly similar sequence or structural features.

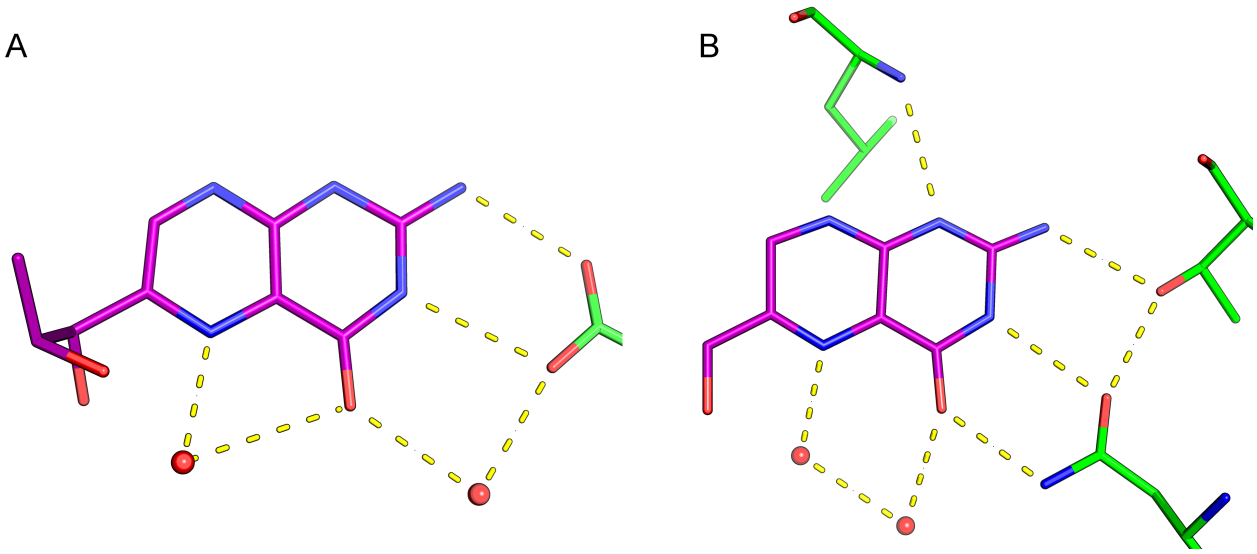


Figure 1: An example of different hydrogen bonding patterns of very similar ligands bound to two different protein folds. The molecules are drawn using tubes (edges) which represent the covalent bonds between the non-hydrogen atoms. The vertices represent the centers of the atoms in the molecules with purple and green denoting carbon atoms from the ligands and proteins, respectively. The blue and red vertices represent nitrogen and oxygen atoms, respectively. The red balls represent the center of the oxygen atoms the of water molecules. The dashed yellow lines denote the pairs of non-hydrogen atoms that are participating in hydrogen bonds. The protein in panel A is a G. gallus dihydrofolate reductase (DHFR) (PDB 1DR1). The protein in panel B is a Y. pestis 6-hydroxymethyl-7,8-dihydropterin pyrophosphokinase (HPPK) (PDB 2QX0). These two sites are difficult to recognize as similar because only two protein atoms form similar hydrogen bonds with the pterin and the match between the hydroxyl group and carboxylate oxygen does not provide a strong signal (this is difficult to present in 2-dimensional images). Notice: for interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

In the process of considering the binding site comparison problem, we have discovered a number of challenges which cannot be addressed directly via methods similar to those presented in this dissertation. We provide two examples where simple similarity heuristics fail. The first difficulty is water molecules are very important for protein-ligand recognition. If the water molecules present in the structures are ignored, it can be challenging to recognize that two binding sites, from otherwise unrelated proteins, can bind similar small molecules (Figure 1). The second challenge is that two proteins, unrelated by sequence or protein structure, may bind the same small molecule in opposing orienta-

tions with respect to the shape of the binding sites (Figure 2). Therefore, maximizing the overlay of binding site surfaces need not result in a good superposition of bound small molecules, even if the small molecules are the same.

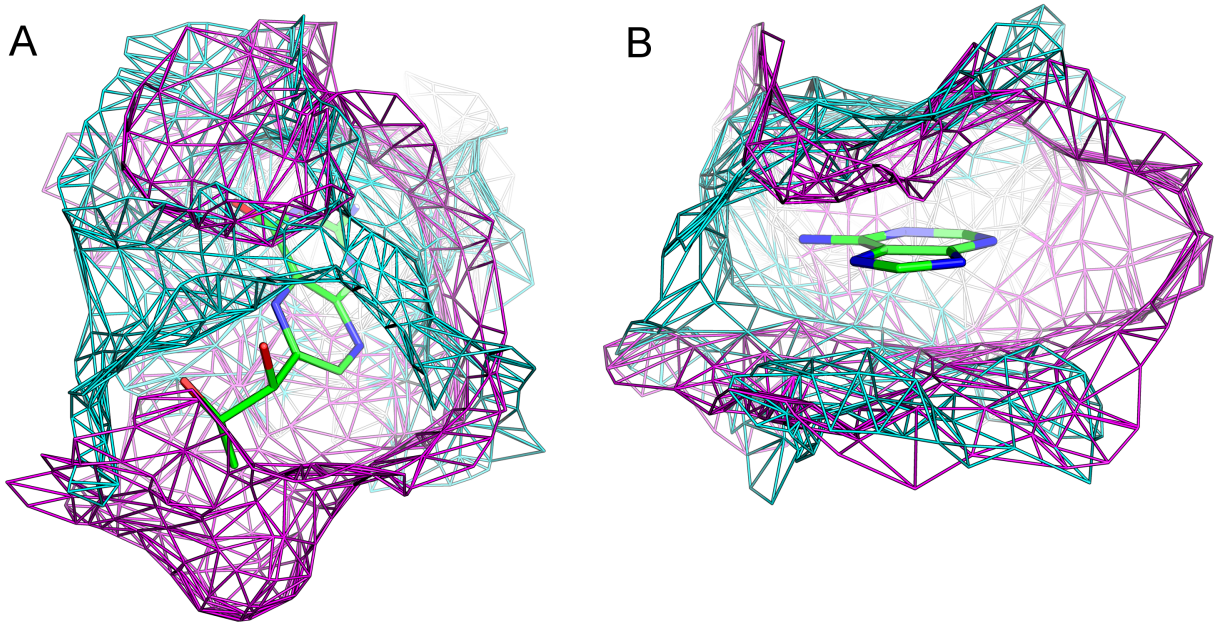


Figure 2: Examples of very different protein surfaces near the binding sites of the same small molecules. The mesh surfaces represent the boundary between the protein and other molecules in the solution. One might expect that if two proteins bind the same small molecule that their mesh surfaces would be similar. However, this expectation does not necessarily hold for proteins from different folds. In this figure, the proteins were aligned using the bound small molecules as the reference frame. In panel A, one can see that the molecular surface of G. gallus DHFR (magenta mesh) is quite different from that of the molecular surface of Y. pestis HPPK (cyan mesh). In panel B, the molecule adenine is shown with the molecular surfaces patches from an N6-out protein (magenta mesh) and from an N6-in protein (cyan mesh)–the position of N6 is at the end of the blue tube that is not part of the two rings. Notice that in both panels the small molecule alignments do not maximize the amount of overlapping surface area. Therefore, the site alignment with maximum surface complementarity need not be close, in pose space, to the better alignment with respect to the position of corresponding ligand atoms.

Chapter 2

Background

Because this dissertation builds upon from techniques and science from two scientific fields, relevant background topics from Computer Science and Biochemistry will be illustrated and explained. In particular, key points are presented for the partial matching of objects and the process of training and testing machine learning models. A brief introduction to protein structure and molecular forces is provided to explain protein terminology and chemical characteristics of biomolecules.

2.1 Protein Biochemistry

Biochemistry is the study of the chemistry used by living organisms to carry out the tasks associated with life. Some features of life include growth, using energy sources (food), and reproduction. These tasks are performed using a large number of molecular constructs that vary greatly in size and complexity. The molecules are typically classified by their functions and chemical composition. Some of the classes of molecules are biopolymers built from a relatively small set of small molecules, and they include proteins and genetic material (DNA and RNA). Given the breadth of the field, a brief overview to biochemistry is not possible here and a good reference book that explains the current views of the field based on experimental evidence is *Biochemistry* by Voet & Voet [100].

Protein biochemistry can be characterized as the field of chemistry that studies the unique features of proteins. Proteins are used by all known living organisms to accomplish specialized tasks. At a high level, proteins seem deceptively simple in that they are biopolymers comprised of five elements (H, C, O, N, S). Proteins are built from 20 basic building blocks, called amino acids or residues. The residues are linked in a chain, by covalent bonds. This chain is many times called the backbone, and it consists of alternating peptide bonds and amino acid side chains. At some point shortly after its peptide bonds are formed, a protein's chain "folds" to give the protein its unique 3D shape [7].

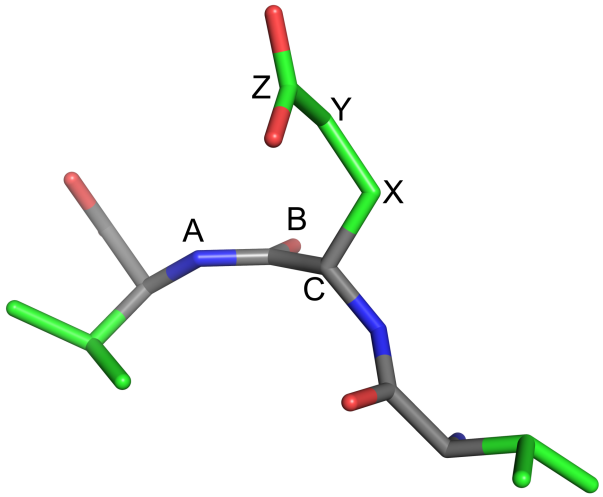


Figure 3: An example of a three amino acid section of a protein. The backbone carbon atoms are colored gray, and the sidechain carbon atoms are colored green. **A**, **B**, and **C** are backbone nitrogen, oxygen, and alpha carbon atoms. Notice that **A**, **B**, and **C** and the carbon atom in their center are in a plane; this is a peptide plane and is an important feature of protein backbone structure. **X**, **Y**, and **C** are participating in covalent single bonds. An example of a dihedral angle is given by **CXYZ** where the convention is to hold **CXY** fixed and to vary the position of **Z**. In particular, **CXYZ** is the angle between the vectors **CX** and **YZ** when they are projected in a plane that has its normal in the direction of **XY**.

The flexibility of proteins is mainly due to the fact that many of the covalent bonds in proteins are single bonds. One characteristic of a single bond is that the two sets of atoms on either side of the bond can rotate relative to each other (with the bond as the axis of rotation), and these rotations can be used to describe most protein motions.

Definition. A **dihedral angle** is the relative rotation of two sets of atoms that are connected by a single bond where the angle is computed with respect to the connecting single bond.

The large number of dihedral angles is one of the main reasons why computationally modeling proteins is a major challenge. Modeling all of the joints' degrees of freedom in proteins as a discrete search space is intractable for most proteins. Furthermore, modeling protein flexibility as a discrete search space is restrictive because many preferred angles are better modeled as distributions with relatively large variance.

Another challenge is that proteins are very small. Molecules at the scale of proteins cannot be observed using even the most powerful microscopes and cannot be accurately modeled using Newtonian mechanics. The main tool used to observe the 3D structure of proteins is x-ray crystallography. At the present, the use of x-ray crystallography to resolve the structure of a ”new” protein that has not yet been resolved is a long and challenging research and engineering process. The successful application of x-ray crystallography yields a snapshot of a fully 3D model of the relative atomic coordinates of a protein (many times called a protein structure) and any small molecules that were bound to the protein. By carefully analyzing the geometric and chemical properties observed in protein structures and the small molecules they bind, theoretical chemists have developed quantum mechanical and statistical models to describe the forces relevant to protein small-molecule interactions and binding.

2.1.1 Molecular Forces

One fascinating feature of biomolecules is that their unique 3D structures strongly depend on the ”weak” molecular forces within the molecules and between biomolecules and their environment (solvent, etc.). Because the ”weak” molecular forces have a much smaller magnitude than molecular bonds the constraints imposed by the forces are less rigid and the forces take less energy to overcome. The ”weak” forces and dihedral rotations of

single bonds allow the molecules to be flexible. A biomolecule's flexibility has a large impact on its overall characteristics, and the "weak" molecular forces can be characterized by their dominant features. Therefore, the modeling of proteins and small molecules requires an understanding of the "weak" molecular forces.

Two types of "weak" polar interactions are due to molecules having charges, with opposite signs, that are brought in to close proximity. Ionic forces are between oppositely charged atoms or functional groups that have formal charges. Ionic forces are characterized as being relatively strong and having less of a directional dependence than hydrogen bonds. Examples of objects formed by ionic forces are salt crystals and salt bridges in biomolecules. Ionic forces are not covalently bonded interactions as crystals formed by ionic forces generally separate into their separate ions in polar solvents (e.g. much of the Na and Cl in table salt crystals disassociates in water to form Na^+ and Cl^- ions).

The second polar force is the attraction between certain small electronegative atoms that can directionally "share" a hydrogen atom that is covalently bound to one of the two atoms [1]. The protein atoms that participate in a strong hydrogen bonds are nitrogen and oxygen. Because the non-hydrogen atoms in biological molecules are primarily carbon, oxygen, and nitrogen, hydrogen bonds are very important for life on earth and in the study of biochemistry. Hydrogen bonds are considered as a distinct category from ionic bonds because the atoms don't have full formal charges, and the experimental evidence (NMR) that hydrogen bonds have a partial covalent bond-like structure is not observed for ionic bonds. Hydrogen bonds are very important since they help to stabilize proteins and are a primary force for the formation of protein secondary structures.

Another interaction commonly described as an attractive force, that is not technically a force, is the hydrophobic effect. The most clear feature of the hydrophobic ("fear of water") effect can be observed by the very high resistance of oil and water to mix. The hydrophobic effect in biochemistry is characterized by the preference of non-polar atoms

1 Although it is useful draw a distinction between hydrogen bonds and covalent bonds, in nature, there is a continuum between no bond and the presence of a chemical bond

(typically carbon and sulfur) to group together and away from the polar solvent, and cause the orientation of nearby polar solvent molecules to be more constrained in order to satisfy their desire to form hydrogen bonds. Two commonly held hypotheses are that the hydrophobic effect is important in the packing of protein secondary structure elements to form folded protein structures, and that it is a strong component driving the binding of proteins and small molecules.

The force that directly affects all atoms (even non polar atoms) is the van der Waals forces that occur when a pair of atoms are in close proximity. The van der Waals force contains both a repulsive and attractive component. The attractive forces are called the London dispersive forces, and are thought to be due to induced dipole-dipole interactions. The repulsive force is due to the Pauli exclusion principle for the overlap of atoms. The van der Waals force between any pair of nearby atoms is very weak, but due to the very large number of pairs of nearby atoms, the sum of the van der Waals forces is important for the cohesion of protein structures.

2.1.2 Structural Features

We now provide an introduction to important points of larger-scale protein structure. A more thorough introduction to protein structure is given by Brandon & Tooze [16].

As mentioned previously, proteins are comprised of one or more amino acids connected via peptide bonds. Proteins are translated from messenger RNA by a ribosome using the 20 amino acids. Each amino acid has two parts: the main chain and side chain. The amino acid main chain atoms and bond structure are the same for the 20 amino acids. When a number of main chain groups are covalently bonded together end to end, they form a peptide chain (many times called the protein's backbone, or main chain). An amino acid's side chain atoms are those atoms that are not part of the main chain, are the part of the amino acids that differ, and are the reason why proteins are so challenging to model.

Proteins are characterized as having four levels of structure. The first or primary structure is the protein sequence, that is, the listing of the amino acid names from the beginning of the protein's peptide chain to its end (i.e. N to C terminus). Computationally analyzing protein sequences is relatively straightforward since all protein sequences are linear and have no branches. Protein sequences have been studied quite successfully as beads on a string and as character strings with gaps. Protein sequence comparisons are typically computed by dynamic programming [77, 93] or space-efficient approximate methods (i.e. need not find the global maximum) [75, 103]. However, protein sequences are 1 dimensional, and do not indicate which portions of the residues in a small molecule binding site interact with each other or with other small molecules. Also, sequence methods cannot adequately address binding site comparisons between two proteins that have low sequence similarity (typically $<$20% similarity). The reason is at low sequence similarity the relative position of the binding site residues need not be similar in both sequences; in other words, their backbones can and do fold differently.

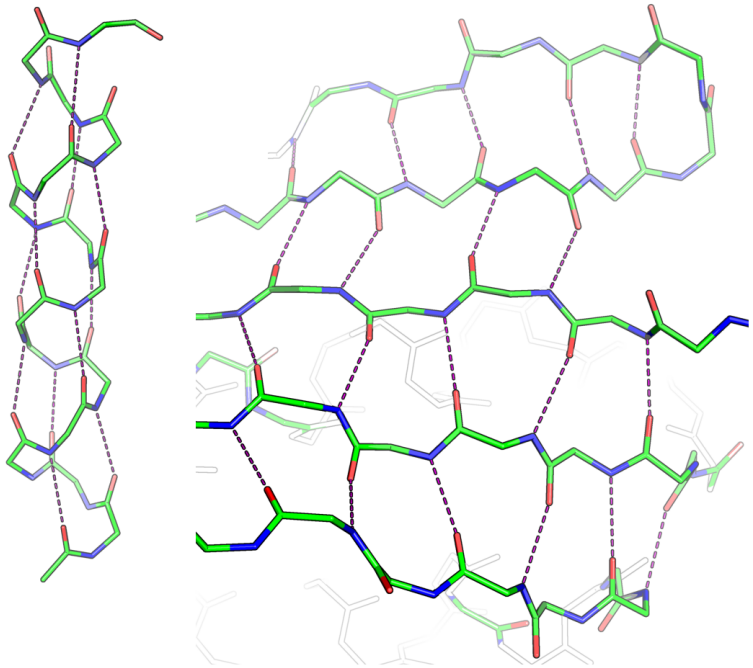


Figure 4: An example of protein regular secondary structure. The bonds between the protein main chain atoms are shown as tubes with red, green, and blue representing oxygen, carbon, and nitrogen atoms, respectively. The amino acid side chain atoms are not shown. The purple dotted lines denote the pairs of atoms which are participating in hydrogen bonds. On the left is an example of an α-helix. On the right is an example of large β-sheet; notice that in the top right corner, there is an example of a β hairpin which is forming part of the sheet. These particular secondary elements can be found in a crystal structure of an E. coli RNA nuclease (PDB: 3AA3).

Protein secondary structure can be classified into three categories: α helices, β sheets, and loops or disordered parts of proteins. An α helix is a local conformation of the protein backbone such that the ith residue's main chain oxygen atom forms a hydrogen bond with the $i+4$th residue's main chain nitrogen atom. The protein's main chain looks like a spiral or helix (Figure 4). A β sheet is a portion of the protein where two or more lengths of protein main chains run parallel or anti-parallel [2]. to each other and form hydrogen bonds between main chain atoms. In that region, the resulting main chain structure looks like a hairpin or, with more strands added along the edge, like a sheet (Figure 4). Protein

2 The chains themselves are parallel in both cases, but if one draws a vector in the direction of increasing residue numbers the vectors can be either pointing in the same or opposite directions

main chain hydrogen bonds dominate secondary structure, and there are two categories of regular, ordered secondary structure elements: α helices [80] and β sheets [79].

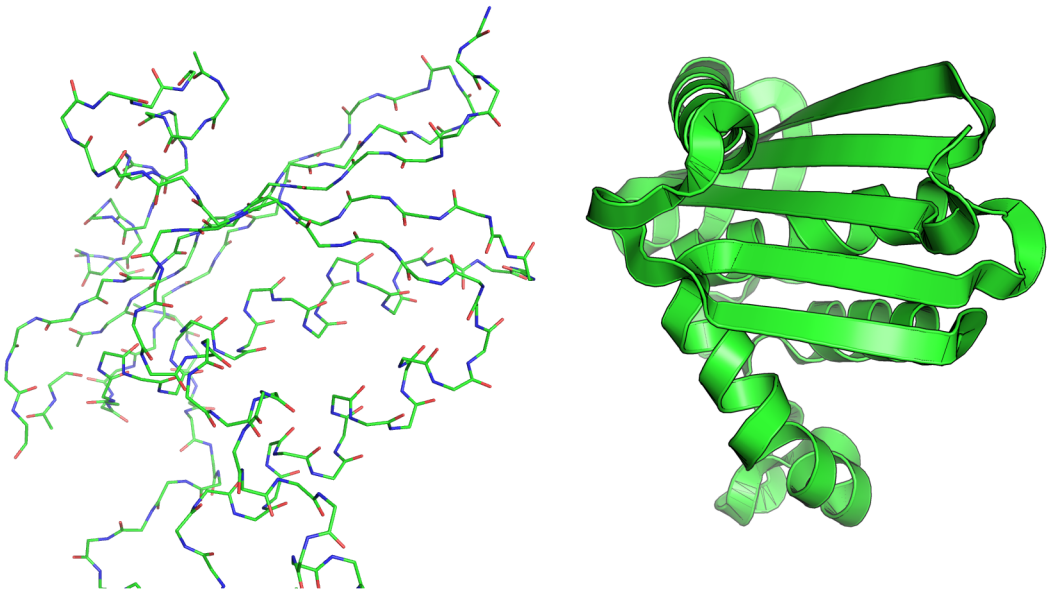


Figure 5: An example of the packing of protein secondary structure elements to form a folded protein. On the left is the main chain of the protein shown as tubes; a β-sheet can be seen in the upper center and right of the protein, and 2 α-helices can be seen in lower right of the protein. On the right is a cartoon drawing of same protein with the secondary structure elements rendered so that they are easily recognized. Cartoon drawings can be very illustrative of the packing of the secondary elements and the overall structure of proteins. These particular secondary elements can be found in a crystal structure of an E. coli RNA nuclease (PDB: 3AA3).

Protein tertiary structure is the 3-dimensional structure that exists after an amino acid peptide chain is "folded". Proteins are called folded since the resulting structure is compact. Two copies of a translated protein sequence will result in two identically folded proteins because the sequence of amino acids specifies a protein's fold [7]. In this dissertation, when the terms protein or protein structure are used, they refers to the protein's tertiary or 3-dimensional structure.

Protein quaternary structure is the 3-dimensional structure of proteins that is formed when two or more tertiary structures, formed by separate polypeptide chains, come together and form a protein or a complex of proteins. The interactions may be permanent (one protein resulting from two or more chains) or transient (protein chains can come to-

gether and separate again). Not all proteins form complexes, but many proteins are not biologically active unless they are in complexes. The computational field of predicting the interactions and relative orientations of protein chains in quaternary structures is called protein-protein docking.

2.1.3 Protein-Small Molecule Binding Sites

Protein-small molecule interactions occur when a protein and corresponding small molecule come into close proximity and the two molecules form a complex that is more energetically favorable than being separate. The portion of the protein that interacts with the small molecule (ligand) due to hydrogen bonds, the hydrophobic effect, etc. is called the binding site. Here, we focus on small organic molecules that are the natural chemical partners (substrates) of proteins, such as ATP, rather than small ions (e.g. sulfate or water).

2.2 Object Recognition

Several definitions are helpful when discussing object recognition as a field or method. Rigid objects can be described by their position and orientation.

> **Definition.** The **center** of an object may be its center of mass, geometric center, etc. as long as the method of measuring the center is consistent for all objects considered
>
> **Definition.** The **position** of an object is generally approximated by the location of its center with respect to a given reference frame (typically a local or global origin).

Many objects have distinct features such that unit vectors can be used to represent the location of the features with respect to the objects' centers (e.g. center of an animal's mass to the tip of its nose or the end of its tail).

Definition. The **orientation** of a given object is its relative heading with respect to a given coordinate system. The heading of an object can be represented by a unit vector.

The position and orientation of a rigid object can be described by six degrees of freedom: three degrees for its position and three for its orientation. A rigid object and be moved from one position and orientation to another by applying a rigid rotation to its position and orientation and adding a translation vector to its position.

Definition. A **rigid transformation** is any rotation and/or translation that can be applied to an object that does not change its shape or volume.

Definition. A **pose** of an object is its position and orientation respect to a particular reference frame.

Definition. An **alignment** is a particular rigid transformation applied to one object that brings its center close to another object's center and provides the first object with approximately the same orientation as the second.

A challenging problem that encompasses much of computer vision and is relevant to computational geometry is: given an example (or model) object, find all copies of that object in a given environment. This class of problems is denoted as the general object recognition problem. In the general case, this problem is very challenging because we seek to find the object even if it is partially occluded or its representation is somewhat distorted (i.e. cluttered environments, significant sensor noise, deformable features, etc.). One approach commonly applied to object recognition is divide-and-conquer. The major steps include: segmentation of the search space and/or locating candidate matches to the model, determining the best alignment between the model and each candidate object (registration problem), and a ranking of each candidate with respect to its similarity to the model by use of a mathematical/statistical model (many times called a scoring function). The divide-and-conquer approach to object recognition has been used with considerable success in many application areas including military applications and handwritten character recognition.

An example of object recognition, is to use a divide-and-conquer approach to search a dataset of images for matches to a given human face [108]. On the surface, this search problem may appear to be an easy task because humans excel at solving this problem when the number of images is small (people tend to get bored or tired if the number of images is too large). However, face recognition is computationally difficult because the research community does not have a complete understanding of how humans process the information in an image, and humans seem to be hardwired to recognize faces [108]. Solving the segmentation problem requires computing features such as the approximate scale of a face (e.g. average number of pixels per face), the colors that represent sensed human flesh tones, lighting conditions etc. Determining the orientation of a person's head is very important for recognition; as an example, a human face has very different characteristics when viewed from the side or viewed from the front. Accurate face recognition requires that the alignment of each face in the image be as close as possible to that of the model face. The fact that each of these steps is computationally challenging for a general image, highlights the fact that humans excel at many complex pattern recognition tasks that are open computational problems.

As with the general face recognition problem, the idea of searching through a dataset of protein-ligand binding sites for those sites that are similar to a query binding site can be attacked using the general object recognition framework.

Definition. A **query object** is that particular object used to search a dataset of objects and have returned those object that are similar to that particular object

Because locating protein-ligand bindings sites is a very difficult problem in itself [69], we assume, that the location of the binding sites is known and do not consider the problem of locating binding sites on a protein. This reduction in scope is similar to the scope of the problem of face recognition for identity where one typically starts with a frontal face scan and compares it to a dataset of frontal face scans [21]. However, unlike human faces, protein-ligand binding sites exist in many different shapes and there is no known

set of landmarks that can be used to align each binding site to a common reference frame. Secondly, many of the query sites are significantly smaller than the sites in the screening dataset, and we seek the best partial match between the query site and each dataset site. Therefore, the binding site search problem is more computationally demanding than human face recognition for identity because one must search for candidate alignments.

Our goal is determining the best partial alignment between a given query site and each binding site in a dataset of sites. This goal can be achieved by computing a number of candidate alignments and, then, ranking the candidate alignments with respect to their accuracy of alignment. Thus, as with many object recognition solutions, we separate the problem of finding the best partial alignment into two subproblems.

2.2.1 Searching for Candidate Alignments Between two Labeled 3D Point Clouds

Determining candidate alignments for the best partial match between two 3D point clouds is a common and challenging problem. Two of the more common solutions are at opposite extremes and are to use the maximum or minimum number of point correspondences and a least squares error fit to enumerate the probable 3D alignments.

Consider maximizing the number of point correspondences used to determine a candidate alignment between the two point clouds. Such a method seems like a good idea since the candidates will use most of the available information. However, using all of the points is problematic since one or two poor point correspondences can greatly influence the least squares error solution because it is a minimizer of the average error over all the corresponding points. In addition, if a number of the points have a significant amount of measurement error, it is difficult to determine the quality of point correspondences and the quality of candidate alignments will suffer. Adjusting the fit by successively removing the point correspondence with the largest residual, recomputing the fit on the remaining correspondences, and terminating when the average residual error is less than an accept-

able tolerance fails in the case of "poison" points [34]. Thus, a straightforward use of a large number of point correspondences, in the presence of significant errors, to determine candidate alignments is generally error prone.

Another approach is to use the minimum number of point correspondences required to have a unique transformation. In three dimensions three unique correspondences and noncollinear points are required for a unique transformation. The beauty of this approach is only three point correspondences need to have low error to get a good candidate alignment. The disadvantage is that a potentially large number of candidate alignments will need to be reviewed. If all possible correspondences are considered and the first and second point clouds have N and M points respectively, the number of alignments to consider is given by the number of ways one can choose three points from N points times the number of ways of choosing three points from M points. The number of three point correspondences is $O(N^3M^3)$. Given the very large number of candidate alignments generated by considering all the three point correspondences, one typically resorts to a sampling method or pruning method to reduce the number of candidate alignments.

Random sampling methods have been used with reasonable success for many object recognition problems in computer vision. One of the earlier such methods is RANdom SAmple Consensus (RANSAC) [34]. RANSAC uses a computational model M of the query object, and a point set of each object in the dataset that is being queried. The RANSAC algorithm is best presented in a pseudocode form as presented in Algorithm 1. Since RANSAC-like methods build up from a minimal number of correspondences to a larger set, they can cope with common issues in computer vision such as partial matching due to occlusion, etc. For this reason, RANSAC methods are quite popular in computer vision applications.

In many application areas, the points can be labeled or contain additional data and the edges between points can be assigned domain specific characteristics. One can use the additional data at the points or edges to drastically reduce the number of 3 point matches to

Algorithm 1 RANSAC meta-algorithm [34]

Require: Model object M (point set, mesh surface, CAD object, linear model, etc.)
Require: Set S of sample points from object to compare to model M
Require: Minimum number of points required for the model (say m)
Require: Error tolerance T for accepting sample points fit the model M
Require: Minimum number of point correspondences desired for final model m_{final}
Require: Maximum iterations N
  for n in range(N) **do**
    $s_n := m$ randomly chosen points from S
    Fit M to s_n to get model instance M_n
    Determine the subset s_n^* of S that is in reasonable agreement with M_n
    Fit M to s_n^* to update the model instance M_n^*
    if s_n^* and M_n^* is the current best **and** $|s_n^*| \geq m_{\text{final}}$ **then**
      save M_n^* and s_n^* as best found
    end if
    if error$(M_n^*, s_n^*) \leq T$ **and** $|s_n^*| \geq m_{\text{final}}$ **then**
      break
    end if
  end for

consider and to increase the number of good alignments. A common technique is to use colored or labelled points and require corresponding points to have compatible features. An example of adding information to sample points is to have a common reference frame for fingerprints, and at each minutia denote the angle that the ridge tangent line makes with the horizontal axis, and require corresponding points to have similar minutia angles [36]. Associating features with data points requires an investment in preprocessing, but in many applications it greatly reduces the search space.

The binding site partial matching problem makes it difficult to use a straightforward application of Probability-Based Matching (PBM) techniques. The diversity of the sites and the partial matching nature of this problem implies that there is not a common reference frame from which to measure features such as angles, etc. (as opposed to fingerprint matching and face recognition). In general, there are no landmarks (e.g. tip of nose in face recognition or wheels in car recognition) that can be used to quickly align two randomly chosen binding sites. The reasons include the fact that protein interaction sites

are very diverse in their sizes, their shapes, and the chemistry they present; and protein binding sites can exhibit significant conformational change with respect to their scale. [3] Because one cannot define a common reference, vectors associated with corresponding points cannot be directly compared as in the finger print matching case, but require at least a 3D rigid transformation before comparison.

2.2.2 Scoring Candidate Alignments

The existence of candidate alignments is rarely sufficient evidence of a match between a model object and the objects to which the model was aligned. The reason is that alignment methods typically trade quality of alignment for a decrease in the search runtime. In fact, in the case of RANSAC [34] or similar methods based on using the minimum number of point correspondences to determine candidate alignments, the candidate alignments require additional scrutiny and filtering to determine which candidate represents the best alignment. Typically, a scoring function or ranking method is used to determine the quality of candidate alignments and provide an ordering of the alignments with respect to their quality of alignment.

As an example, in human-face recognition, the fact that a method is able to align a model face to a face in an image does not imply that the person's model face was a good match to the face in the image. The reason is initial alignment methods typically focus on getting the probable face in an image at the same scale and orientation as the query image, and additional scrutiny is needed to determine if the two faces match. In the case of face recognition for identification with high resolution range scans, one feature that works reasonable well is if the root mean squared error (typically called RMS error or RMSD, see Appendix A) of the points in the two face scans are within ~ 1 mm the faces

3 In protein biochemistry, the existence of significant differences in the relative atomic positions of two of the same or similar proteins is termed conformational change. The reason is almost all of the relative differences can be explained by differences of dihedral angles of single bonds.

are considered a match [21]. Unfortunately, such a stringent tolerance means that different facial expressions can cause the method to err on the side of false negatives. Therefore, the method relies on the assumption that the person being scanned wants to have a positive identification. Thus, face recognition requires a tolerance of match to distinguish between true positives and imposters.

There are four prediction categories that are used to assess the performance of scoring function with respect to an object and a particular class. Suppose we have a set (class) A of objects such that x is in A and y is not. In addition, we have a scoring function $S()$ (classifier) to predict whether a given object is in A.

Definition. A **true positive** is an object that a scoring function correctly classifies as being part a given class ($S(x)$ is A) and $x \in A$.

Definition. A **true negative** is an object that a scoring function correctly classifies as not being part of a given class ($S(y)$ is not A and $x \notin A$).

Definition. A **false positive** is an object that is not part of the class, but the scoring function incorrectly classifies as being in the set ($S(y)$ is A, but $y \notin A$).

Definition. A **false negative** is an object that is part of the class, but the scoring function incorrectly classifies as not being in the set ($S(x)$ is not A, but $x \in A$).

These categories are widely used to estimate the performance of classifiers with respect to given classes. Since most classifiers make classification errors on occasion, a clear understanding of these categories can be instrumental in choosing among classifiers and/or settings thresholds for classes based on errors one seeks to avoid.

In many cases, we prefer to select the best of the candidate alignments and not one that is "close enough". For that reason regression or approximation methods are preferred over classification methods. In addition, when the dependent variable(s) are continuous, classification methods require arbitrary boundaries or thresholds to be set during the training process (i.e. classification requires converting a continuous variable to an

integer variable). It is straightforward and relatively inexpensive to set arbitrary thresholds for classification given a regression solution, but if a classification method was built and the thresholds change, the classification training and testing must be redone. This does not mean that regression is superior to classification, but rather that regression is preferred in the case of approximating continuous values. Classification methods are generally used in the case where the number of classes are finite and the boundaries are meaningful. Since alignment error is represented by a continuous variable, we will focus on regression techniques.

The general framework used to build a ranking method (the regression problem) has been consistent for many years [14, 27, 42]. This framework is as follows:

1. Get a dataset containing the independent variables (measured features) and the dependent variable (measured feature we seek to predict).

2. Use feature selection and extraction; that includes analyzing the raw data to determine which features and combinations of features to use for prediction.

3. Determine the goals of the ranking method and choose one or more approximation or machine learning techniques that fit well with the goals.

4. Fit the models and methods from the third step to dependent features from the second step to predict the independent feature.

5. Evaluate the models and methods on an independent dataset to gauge the generalizability of each model and method.

6. Choose the best model from the fifth step and provide it to the customers or users.

This framework is straight forward, and has been successfully applied in many different applications [14, 27, 42].

Although the framework itself is straightforward, each step has a number of problem specific and significant details that need to be addressed in order to make accurate and

useful predictions. It is precisely for this reason that machine learning, data mining, and statistical inference continue to be active areas of research. In particular, in the regression/approximation step, it has been shown that without prior knowledge (bias) there is no dominant approximation method that outperforms all others on all data distributions (this is known as the "No free lunch" theorem [104]). Since each step represents a significant amount of work with respect to the binding site comparison problem, we will briefly touch on relevant techniques that were used at each step.

The proper collection of informative data is essential for statistical learning methods to be used to analyze the data and make predictions. In many cases, one does not have the luxury of obtaining more data or asking for additional features as the cost of additional data is prohibitive. However, if there is a coordinated effort before the data is collected, the types of data gathered (experimental measurements, etc.) and the statistical analysis techniques should be considered by those engineering the study to provide the maximum impact for the cost of gathering the data. As an example, if experiments are expensive but certain potentially useful measurements can be taken at the time the experiments are performed with relatively little additional cost, then the experimental design should be modified to collect the additional features. Thus sufficient and accurate data collection can be very helpful in setting a sound basis for accurate analysis and predictions, but in many cases it is either not feasible or cost effective.

Given the problems in data collection, it is to be expected that in many cases data analysts are given noisy and/or partial data. The data analyst's job is to determine which features to use in the analysis and prediction phases. Supposing numerous features were given, then one needs to reduce the set of features to a manageable number of features that contain those features that are thought to be or that are statistically shown to be the most predictive. A major reason for feature selection is to avoid the curse of dimensionality. In simple terms, the additional information that an added feature provides to a model decreases with each added feature and because samples are used to estimate the true

population there is a point at which the added information is less than the measurement and sampling errors [27]. A similar idea is that one can fit an overly complex model to a large number of features so that the training data is exquisitely modeled, but predictions on examples not included in the model can easily fail since the model focused too heavily on closely interpolating the training data rather than learning the data features. Thus, feature selection is generally required so that a reasonable number of predictive features are used and the resulting model has good generalizability.

Another common problem is the analysts are usually given raw data and, the data must be processed to obtain useful features. This is termed the feature extraction problem and is becoming even more common as the amount of available raw data is increasing at a much greater rate relative to the quantity of annotated data. An example of feature extraction applied to raw data is automatic annotation of video clips uploaded to websites. A current problem is that video clips on websites such as Youtube generally have very little useful annotation and there are far too many clips to be robustly annotated by humans. Automatic annotation is a type of feature extraction that uses computer vision techniques to define objects in the video frames and uses object recognition to assign the types of objects present in the scene. The assigned features can then be used to classify the videos so that text strings such as ”chair” or ”fire” could be used to search for videos that contain a chair or fire. A more novel approach is to have users provide an image of an object and have the system return the videos that contain objects similar to the user's object.

The choice of which model types to fit depends heavily on the prediction goals and the assumptions about the features. If one seeks to show how much of the relationship can be explained by a linear relation between the dependent variables and a known function of the independent variable, using linear regression is the first tool of choice. If a good approximation is desired and understanding the underlying connection between the features and response is less important, tools such as K-nearest neighbors, neural networks,

and support vector machines have been shown to perform well in practice. On the other hand, if the data is noisy, the underlying relationship is unknown, and one seeks the general trend rather than a highly accurate reproduction of the training data at all points, smoothing methods including thin-plate splines are preferred.

After the model types are chosen, each model must be fit to the training data to build a predictor. Because many models have adjustable parameters that control model features, these parameters need to be given appropriate values with respect to the data. A poor method of choosing the parameter values is to use those parameters that allow the model to best fit the training data, because this method does not have an estimate of the model accuracy for new data. A better method is to fit the model with a wide range of parameter values and choose the best parameter value based on the model that best predicts on a validation dataset. In many cases, the cost of having a separate validation set is prohibitive. Two better methods to use to estimate the model parameters when a validation set is not available or is cost prohibitive are cross validation and generalized cross validation [23, 38].

Next, the models must be compared on a testing dataset that is separate from the training and validation sets to gauge the generalization abilities of the models. This step is crucial, since more complex models tend to have better predictions on the training sets than the less complex models. However, due to the curse of dimensionality and/or overfitting, complex models need not outperform the simpler models on new examples. As examples, the model that the stock market will always be higher at the end of each successive year on average outperforms all other existing models when the question is "will a given stock market index have a greater value than today after exactly one year?" Similarly, in the computational drug design field virtually all of the methods designed to predict the change in free energy upon protein-ligand binding perform, on average, no better than using the ligands' molecular weight to predict the change in free energy [44].

One could argue that the more complex models are a waste of resources. However,

the advantage of more complex models is they can be used to analyze the data and ask more specific questions than can be asked of the very simple model. In fact, one of the better uses for models is to filter large quantity of inputs to an amount that experts can adequately handle and focus human expertise on those examples that tend to have the most interesting characteristics.

2.3 Computational Geometry Techniques

Proteins have a number of constraints that can be classified as distance or angle constraints. The current models of protein-protein and protein-small molecule interactions are based on relative distances between atoms and angles between sets of bonds. Therefore, many existing computational geometric methods are well suited for studying protein-small molecule interactions.

2.3.1 Addressing the Partial Matching Problem

In this dissertation, the partial matching problem is to find the best match between a given part of an object and each full object in a given dataset. This search is called partial because there exist features in the full objects that do not have correspondences in the partial object. Partial matching of fully 3D objects is particularly challenging since many methods and heuristics used for object matching are only feasible for two dimensions (i.e. images) or are not applicable for partial matches. In addition, in the binding site matching problem we seek the best partial match between the query site and each dataset site (not just those sites that are already known to be similar to the query site).

Examples of commonly used techniques that do not perform well for 3D partial matching include aligning objects via their major and minor axes, Hausdorff distance, and distance or gradient based probabilistic matching methods [78]. Partial matching using major and minor axis or Hausdorff based distances will tend to place the partial object near

the center of mass of the larger object which need not be the best match. Techniques such as histogram of oriented gradients [24] that perform well for 2D images tend to not scale to 3 dimensions. Probabilistic matching methods such as spin images or histograms of point-to-point distances cannot be used to heavily prune the search space since it is both challenging to determine if a histogram of a partial object cannot be contained in the histogram of a larger object and the partial objects need not have the large distances present in the full objects which is where many of the differences in two full objects tend to be observed. In work for this thesis, the partial matching problem has been addressed using a variety of methods including brute force and generalizing techniques from object recognition.

2.3.2 Applying Inverse Kinematics

In the later portion of this dissertation, we investigate the contribution of flexibility of proteins to bind the same small molecule. Protein flexibility is known to play an important role in the process of protein-ligand binding [6, 22, 61, 106]. One way to model protein flexibility is to consider each atom as a joint and each covalent bond as a rigid link between the joints. By modeling proteins as joints and links, one can use the method of inverse kinematics to pull atoms directly or indirectly via features to new locations while obeying atomic and bond constraints.

2.4 Comparing Protein-Small Molecule Binding Sites

There are many tools that have been designed to align proteins using the relative positioning of key features [66]. The features may include the relative positions and orientations of α-carbons [33], secondary structure features [47, 60], protein residues [9], or more abstract features such as hydrogen bond donor atoms (this dissertation, etc.). As noted in the introduction, a key requirement of 3D binding site comparison methods is that one must

know the relative 3D positions of the protein atoms for both the query and dataset sites. Similarly, the methods presented in this section require accurate 3D atomic coordinates for the proteins considered.

Because of the emphasis of protein comparison methods on the relative positions of atoms or derived features, the methods can be considered as point cloud comparison methods subject to biochemical constraints. [4]

Definition. A **set** is a collection of objects such as integers, decimal numbers, faces, proteins, etc. with a membership operation $\in$. Such that, given a particular set S and an object s we write $s \in S$ if s is in the set S and, write $s \notin S$ if s is not in the set S.

Definition. A **finite set** is a set such that the number of elements in the set is a positive integer n that is less than infinity.

Definition. An **unordered set** is a set that does not have a defined order for the set elements.

Definition. A **point cloud** is an unordered finite set of 3D points that have finite coordinates (i.e. the set is bounded). In set notation, a point cloud P with N points is a point set and may be written as $\{p_i = (x_i) \mid i \in [0, N) \text{ and } x_i \in \mathbb{R}^3\}$.

In order to reduce the time complexity of searches and to match complementary points, many point based methods extend the point cloud definition to include point labels.

Definition. A **labeled point cloud** is a point cloud such that each point has one label from a finite set of labels L. That is, a point cloud P with N points such that $\{p_i = (x_i, l_i) \mid i \in [0, N) \text{ and } x_i \in \mathbb{R}^3 \text{ and } l_i \in L\}$.

In some cases it is advantageous to assign a direction (unit vector) to each point.

Definition. A **labeled point cloud with directions** is a labeled point cloud such that each point has a label and a unit vector. In set notation: $\{p_i = (x_i, l_i, v_i) \mid i \in [0, N) \text{ and } x_i, v_i \in \mathbb{R}^3 \text{ and } l_i \in L \text{ and } \|v_i\| = 1\}$.

4 If one wishes to be more precise about point sets and index sets, an introduction to point set topology is a good place to start.

Comparing parts of proteins as point clouds has a strong advantage in that point clouds have been and continue to be extensively studied in computer science and application areas [12, 31, 52, 95].

2.4.1 Protein Structure Alignments

One way to align two binding sites is to align their respective protein structures by aligning secondary structure features and 3D coordinates of their α-carbon atoms. Two commonly used automatic structural alignment tools are Dali [47] and Secondary Structure Matching (SSM) [60]. The proteins that carry out the same or highly similar tasks in different species tend to have a similar protein structure, conserved residues in their binding sites, and binding sites in the same location relative to the full structure. Therefore, structural alignments are useful, but not necessarily sufficient, to engineer small molecules that are specific for a particular species. As an example, structural alignments of two proteins necessary for cell life (e.g. dihydrofolate reductases, etc.), one from a bacteria and one from a human, can be used to design a molecule that prefers to bind to the bacterial protein and not to the human protein (provided significant differences do exist in the binding site). Such a preference of binding can be used to design potent antifungals or antibiotics to treat particular infections with hopefully few side effects in humans. However, structural alignments cannot rule out the possibility that a similar binding site exists in a protein that is structurally distinct from the target protein.

The primary goal of protein structural alignments is to have the best superposition of entire protein structures. The alignment methods typically present the quality of backbone superposition and the differences in protein sequence at each residue's position [47, 60]. In addition, because of the focus on backbone superposition, in practice, structural alignment methods require many more residues than those that typically form a small molecule binding site. Because the relative orientation and packing of protein residues determines the shape and chemistry of small molecule binding sites, protein structural

alignment methods themselves do not give a detailed report of the similarities and differences present in the binding sites. For this reason, experts must look at the aligned structures in molecular graphics and draw on domain knowledge and experience to design potential drug molecules that prefer to bind to the target structure instead of other proteins. The reliance of structural superposition methods on the positions of protein backbone atoms implies that such methods can rarely find any alignment between two structurally unrelated proteins. In addition, if two binding sites have different relative positioning with respect to their protein backbones, the sites will not be well aligned using structural alignments. In conclusion, automated protein structural alignment tools are very useful in drug design, but are restricted to proteins within the same protein family and do not give detailed comparisons of binding sites.

2.4.2 Comparing Patterns of Binding Site Residues

One way to remove the strong structural bias of structural alignment tools, is to search a protein structure dataset for proteins with patterns of the same or similar residues as those that form the query binding site. The residues in a given binding site may be represented as a labeled point cloud with directions, such that:

- x_i is the 3D coordinates of the α-carbon for the ith binding site residue
- v_i is a 3D unit vector that represents the orientation of the ith binding site residue (e.g. v_i could be given by the vector from the α-carbon to the β-carbon for the ith residue)
- l_i is the label associated with the ith residue; in many cases l_i is the name of the residue, and there are 20 standard residues

Two binding sites A and B, represented as labeled point clouds with directions, can be compared by searching for the best correspondence between the two sites.

Definition. A **pair of corresponding points** is a tuple (a_i, b_j) such that $a_i = (x_{a,i}, l_{a,i}, v_{a,i}) \in A$ and $b_j = (x_{b,j}, l_{b,j}, v_{b,j}) \in B$ and $l_{a,i} \sim l_{b,j}$.

The methods to search for and the determinations of the maximal set of point correspondences differ among the existing methods. However, a general progression is to compute the superposition of the two sets of correspondences using a least squares error fit and require that the average error be less than some tolerance and the vectors of the corresponding points have a dot product greater than some tolerance. These residue based methods are time and space efficient because of the large number of labels (usually 20 amino acid types) and the relatively small number of points (usually far fewer than 100 residues)

Two tools designed to compare binding sites based on residues are JESS [9] and PINTS [94]. An advantage of both JESS and PINTS over similar tools is they use statistical models to estimate the significance of match scores by giving a probability estimate for a random alignment to have the same score (p-value). Unfortunately, residue based methods have difficulty in aligning a query binding site with a similar binding site that has significant mutations or with a binding site from an unrelated protein since such sites will have a small number of residues in common. A specific example that may prove difficult for residue methods would be aligning the adenine pocket of a kinase ATP binding site with the adenine pocket of a nicotinamide adenine dinucleotide phosphate (NADP) dependent alcohol dehydrogenase as the residues binding adenine are distinctly different in both pockets.

2.4.3 Comparing Labeled Sets of Chemical Points

A logical progression from residue based methods is to abstract the residue features and concentrate on the common chemical interactions and shape complementarity of protein-ligand binding sites. The reasons for this abstraction include the fact that molecules interacting with proteins do so based on chemical properties and not specific residue names.

By removing the dependence on matching residues, one can compare and contrast the chemical features that are known to be important when describing the interactions between proteins and small molecules. There are a number of existing methods that are not comparing residues based on their names. A few of these methods are described briefly since a particular class of these methods is described in detail in Chapter 3.

Site comparison methods such as SIFt [26] and CompSite require users to correctly align the sites prior to running the comparison software. SIFt is a hybrid approach that does not entirely discard the notion of residues and by assuming all of the considered binding sites have residues in approximately the same relative location it reduces the 3D representation of each binding site to a vector. For each residue, SIFt uses a bit string to encode whether that particular residue is making a certain interaction with the bound ligand. Thus, SIFt uses both the protein and ligand information, and SIFt trades off the relative 3D orientation and position of residue features for speed and ease of applying "off-the-shelf" machine learning techniques. The assumptions of SIFt are the binding sites come from structurally related proteins, and under that assumption the encoding used is highly effective. However, SIFt is dependent on user provide alignments. User provided alignments can be a source of significant error, and most users are unable to provide alignments among proteins from different families.

CompSite uses the 3D binding site representation developed for SLIDE [107]. As with SIFt, CompSite requires users to correctly align the sites prior to running CompSite. This representation completely abstracts away the protein residues and is a chemistry labeled point set in the ligand binding volume [107]. The main work-flow is as follows:

1. Build the representation for each site.

2. Use complete link clustering of the points from all of the sites.

3. Use majority vote to find the regions of the binding sites that have the same chemistry in more than 50% of the sites.

4. Label substantial clusters where the majority of the points agree as a similarity points.

Given the level of abstraction of the site representation, CompSite is less dependent on the structural similarity of the proteins than SIFt. However, as with SIFt, the performance of CompSite is greatly affected by the user provided alignments of the sites.

Methods that require user-provided alignments of binding sites suffer from a few drawbacks. Having users align binding sites requires additional tools and can be labor intensive. The alignments can be problematic since there can be substantial error in both small-molecule alignments and protein structural alignments, and the alignments require substantial similarities in the structures or ligands. Also, such methods rely on the user's knowledge of the protein space, and are unlikely to useful for data mining as the user already has some prior knowledge and bias about the sites.

To remove these restrictions, a growing number of site comparison methods use full 3D alignments of the query site to each site in a dataset. While there is a number of variations on the general method to compare binding sites using labeled point clouds, all of the existing methods use additional features at each chemical point to increase the accuracy of matching and scoring. In particular, the point clouds are very much like the sets presented in Section 2.4.2, but the more abstract methods tend to have 4 or 5 types of labels (rather than 20). A less obvious difference, that does not necessarily affect the computational characteristics of point cloud matching algorithms, is that the position of the points and their associated directions differ greatly between residue methods and chemical point methods. Most of the existing chemical point methods use the binding site atom centers as the points. Others, such as SimSite3D and MED-SuMo, compute the relative position of the points based on the local geometry of the binding site atoms and residues.

Besides comparing the labeled point clouds, some of the methods also compare the sites' molecular surfaces. The advantage of comparing surfaces is the sites' shapes are

used, in addition to the chemical points, to gauge the similarity of sites. However, computing the degree of surface similarity is a relatively costly process when compared with computing the similarity of chemically labeled point clouds. As presented in the general object recognition framework, these methods all require a scoring mechanism to determine the quality of alignments and the similarity between two aligned sites. These more general methods include SimSite3D, SiteEngine [91], SuMo [53], Cavbase [89], and SitesBase [37].

Given the limits of protein sequence and structure based methods, it is likely that the focus on chemical features has the potential to yield more fruit when applied to comparing binding sites. At the present, the hypothesis, ”binding sites that binding similar ligands exhibit similar chemistry and shape features such that they can be detected by computational methods”, has not been adequately addressed. Therefore, in the next chapter, a method using chemically labeled point clouds with directions is presented as a basis to explore the hypothesis.

Chapter 3

Comparing Binding Sites as Chemically Labelled Point Clouds

Fully characterizing the processes of protein-ligand interactions is a challenging problem and is an active area of research. There are several major challenges:

- Proteins and ligands are flexible molecules.
- Some of the internal degrees of freedom of interacting molecules may change significantly over the course of the interaction (e.g. conformational change due to coordinated movement of residues).
- Proteins and ligands that interact have been shown to coordinate corresponding motions.
- Current theoretical and experimental evidence implies that protein-ligand interactions can only be truly characterized by quantum mechanics.

A review of computational methods that model protein-ligand interactions to predict the favorableness of such interactions (called binding affinity) may be found in [74]. At the present, proposing to design and implement a computational method to fully address

one of these challenges, in the context of comparing tens of thousands of binding sites, would constitute a very ambitious goal.

As is typically done when developing high-throughput computational methods, we introduce a number of simplifying assumptions so that our computational method to compare protein-ligand binding sites provides a reasonable result within an acceptable time frame. Ideally, protein binding sites would be modeled using quantum mechanics. However, at the present, quantum mechanical interactions are very computationally demanding and challenging to model. In the case of proteins, quantum mechanics are approximated using Newtonian mechanics with very small timesteps (these approximation methods are called molecular dynamics [18, 19, 56]). Molecular dynamics simulations are, at the present, computationally expensive and not feasible for high-throughput computational chemistry methods. To achieve sufficient throughput and sidestep addressing the challenging questions of molecular motions, binding site comparisons are performed with the proteins approximated as rigid objects The binding site atoms and features of a protein are modelled as a labeled point cloud with directions.

The presented method is a compromise over several competing goals. From the beginning, our main goal has been to push the envelope and find, in proteins unrelated by sequence or structure, sites that can bind similar molecules but could not be aligned/detected by existing tools. A major engineering goal was to have a method that could search one query site versus all the binding sites in the Protein Data Bank (PDB) [10, 11] within one day on one processor core. Using our implementation of the method, presented in this chapter, we provide some examples of significant hits that could not be found with other methods.

3.1 Methods

This section presents the details of the design and implementation of the binding site comparison method that is implemented in SimSite3D version 3.3. The site representation, the computing of site alignments between pairs of sites, and the scoring of site alignments are covered.

3.1.1 A Detailed Representation of Protein-Ligand Binding Sites

Definition. The **ligand binding volume** is that portion of the volume of a protein-ligand binding site that is not occupied by one or more protein atoms.

Definition. A **site map** is a specific class of chemically labeled point clouds with associated directions used to model binding sites in this chapter.

Definition. The **site map volume** is the portion of a ligand binding volume that is used to create an associated site map.

A site map captures the essential chemical and some shape features of a binding site, and is computed directly from the local geometry and chemistry of the binding site atoms. A site map represents the chemistry and shape of ligands that would make strong favorable interactions with the protein part of the binding site. A site map is a chemistry labeled set of points with associated direction, and the points lie in the ligand binding volume (a site map is derived from a SLIDE template [107]). This emphasis on abstract chemical points allows the comparison of binding sites to be independent of the explicit degree of similarity of the residues that comprise the binding sites. As an example, when comparing two site maps, if a hydrogen bond donor atom from the query protein is an amide nitrogen, its acceptor site map points may correspond to any acceptor site map points, in the dataset site map, from any hydrogen bond donor atom (not just an amide nitrogen). Since the site map model has relatively few points, the model allows for rapid alignment and comparisons of protein-ligand binding sites.

A site map can be automatically generated for a binding site given a user provided protein coordinates file and the location of a protein-ligand binding site. A binding site's location and volume are determined by the intersection of a user provided volume object and protein coordinates. Two easily supported volumes types are ligand based or spherical. If a ligand is given, one can compute the volume of the site using the axis-aligned bounding box with the smallest volume that contains the ligand and adds a buffer of 2.0 Å on each side of the box. If a sphere (point and radius) is provided, that sphere can be used as given (the user is expected to add a reasonable buffer). A given site volume focuses the search method to only consider those site map points that are inside the volume.

The placement and type of features in the site volume are based on biochemical observations and experience [50, 107]. When designing computational approaches to solve difficult problems, domain knowledge and understanding the questions posed are crucial to determine which types of features to measure and compare. A major challenge is to find a good balance between the details of the essential features and the computational cost to compare two objects. In the case of protein-ligand interactions, the weak atomic forces are known to be important determinants and the driving forces of protein small-molecule complementarity. These weak forces or interactions are some of the features modeled at varying levels of detail by small-molecule docking tools [39, 62, 90], molecular dynamics simulation packages, and small-molecule similarity tools [43, 72].

In this chapter, the protein-ligand interactions are categorized into several classes of interactions. These types of interactions are hydrogen bonds, the hydrophobic effect, and small-molecule metal interactions; and are now presented as parts of a labeled 3D point cloud with associated directions.

3.1.1.1 Hydrogen bonds

It has long been recognized that the formation of hydrogen bonds between a protein and ligand is one of the main specificity determinants for protein ligand binding and can be

used to model protein-ligand interactions [62]. The comparison of the hydrogen bonding capabilities of two binding sites can be done by assessing the degree of overlap between complementary hydrogen bonding volumes.

> **Definition.** The **hydrogen bonding volume** is the volume in a given binding site where a ligand atom can be placed and form a hydrogen bond with an atom in the protein.

The hydrogen bonding volume for protein atoms can be defined by the parameters used to recognize protein-ligand hydrogen bonds in protein-ligand docking tools (e.g. SLIDE [90]).

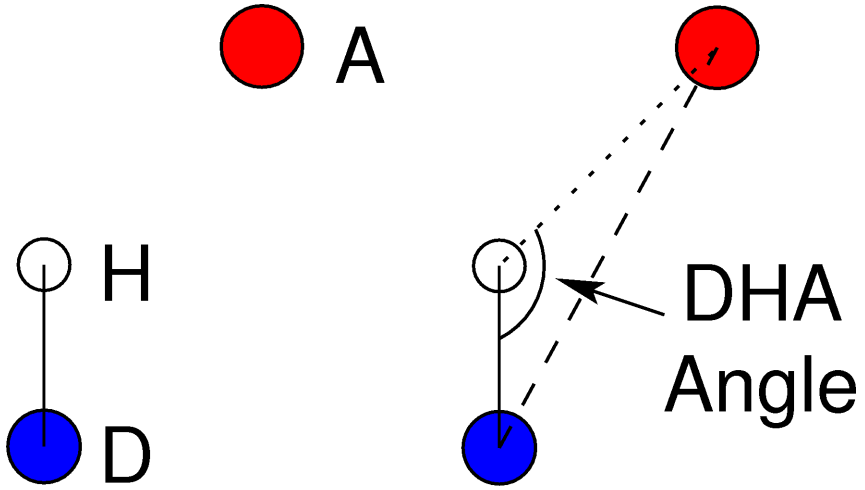


Figure 6: This is an illustration of a computational model of hydrogen bonds. On the left is a hydrogen bond donor atom D with a covalently bonded hydrogen atom H. The red ball is a hydrogen bond acceptor atom A. The dotted line is the distance between H and A; acceptable distances are in $[1.5, 2.5]$ Å . The dashed line is the distance between the acceptor A and donor D, and should have a length in $[2.5, 3.5]$ Å . The DHA angle must be in $[2\pi/3, \pi]$ radians.

In geometric terms, the **hydrogen bonding volume** is a truncated spherical cone C. C is defined as the subtraction of C_1 from C_0, where C_0 is the volume of a spherical cone given by the intersection of a 3.5 Å radius sphere with center at the center of the hydrogen bond donor atom and the apex of the cone at the center of the corresponding hydrogen atom (Figure 7). The cone's axis is placed where the angles for the hydrogen bond would be closest to the ideal values (Figure 6). C_1 is similar to C_0 in that it has the same axis

and apex, but the bounding sphere has a maximum radius of 2.5 Å . The volume of C can be approximated by a surface S_C that is in the middle of the volume with respect to the cone's apex and axis. The spherical cap S_C is defined by a sphere centered at the hydrogen donor atom's center, having a radius of 3.0 Å and keeping only the portion of the sphere that is inside C. The cap S_C can be approximated by a sparse sampling of points on the cap. One may start with the point lying on the axis of C and then add sparse sample points in regions of high probability of forming hydrogen bonds based on a survey of protein structures (i.e. experimental evidence) [50, 107]. Each sample point includes the chemistry of the ligand atom that could form a hydrogen bond with the protein at that point and the directionality of the hydrogen bond that is estimated by the normal of the surface at the sample point. To keep only those points that are relevant to the binding site and are not too close to protein atoms, points that fall outside of the site map volume or are within 2.5 Å of any protein heavy atom are discarded. In this manner, the volumetric representation is reduced to 0-5 sample points for each polar hydrogen and lone pair of electrons [107].

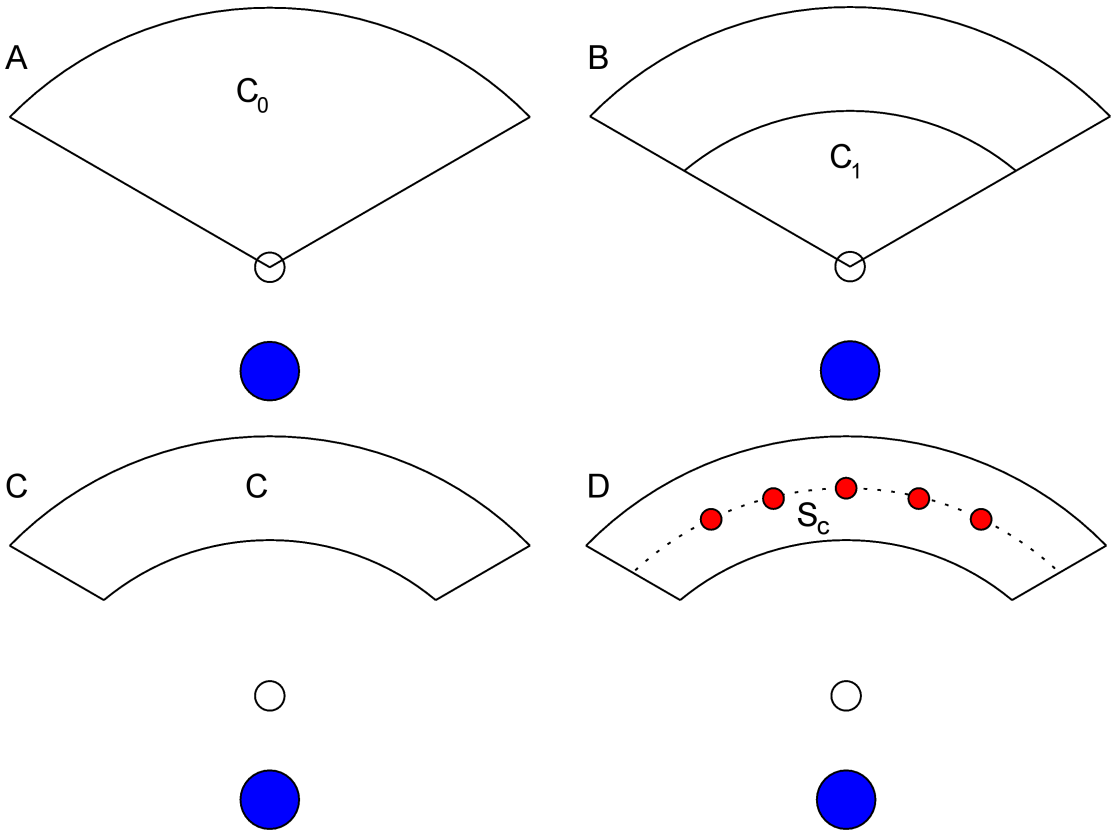


Figure 7: A two dimensional sketch of the three dimensional hydrogen bond model presented in this section. The center of the blue, white ball is the center of the hydrogen bond donor atom, hydrogen atom, respectively. A cross section of the spherical cone C_0 can be seen in panel A. Panel B shows a cross section of C_0 that overlaps with a cross section of C_1. Panel C is a cross section of the hydrogen bonding volume C. Panel D shows the center of a cross section of C and some hydrogen bond acceptor points that are 3.0 Å from the center of the hydrogen bond donor atom.

3.1.1.2 Hydrophobic interactions

The hydrophobic effect is an important component of protein-ligand binding. From a strictly geometric viewpoint, the main distinction between the matching of hydrophobic interactions and the matching of hydrogen bonds is that models of hydrophobic interactions, generally, do not have a preferred direction.

Definition. A **protein hydrophobic atom** is any protein carbon or sulfur atom that is not covalently bound to an oxygen or nitrogen atom.

The exposed hydrophobic portion of a binding site is represented by discretely sampled spheres of radius 2.5 Å centered at each hydrophobic atom. The poses of the spheres are computed with respect to the local coordinate system defined by two of the hydrophobic atom's neighbor atoms (for a given residue and atom name, the neighbors are fixed). Sample points closer than 2.5 Å to any protein atom or within 1.75 Å of a polar site point are removed. The remaining surface points represent the portion of the binding site where ligand hydrophobic atoms could be placed to make favorable hydrophobic interactions with the protein.

3.1.1.3 Metal-template points and metal interactions

Metal ions are found in about 30 percent of all protein structures and are an important (structural or catalytic) component of many ligand binding sites. Metal ions are typically positively charged. From a site map perspective, they are likely to interact with electron-rich, hydrogen-bond acceptor atoms in a ligand. Metal ions can be modelled as part of the protein surface by evenly distributing acceptor on a sphere centered at the center of the metal atom (the radius depends on the chemistry of the metal ion). Metal points that are within 2.5 Å of any protein heavy atom are removed. During alignment and scoring no distinction is made between acceptor points from hydrogen bond donors and acceptor points from metals.

3.1.2 Enumerating Candidate Alignments

At the present, there are no known 3D methods that can compare two arbitrary protein-ligand binding sites without first aligning the binding sites. Because there are no known features to compute a canonical orientation that is applicable for all binding sites, the alignments must be computed at match time. This absence of universal alignment fea-

tures makes it a challenge to determine which of a set of candidate alignments is the best alignment. Thus, a general practice is to compute a number of more probable alignments, and then, score those alignments with a suitable scoring function.

A straightforward method can be used to enumerate poses to bring one site into the reference frame of another site. This method is based on the fact that exactly three non-collinear points are necessary and sufficient to determine a unique pose in three dimensions. One could proceed by listing every possible pose by fitting all combinations of three points from one site and three points from a second site. However, many of the fits would have large residual errors, and can be eliminated by having a maximum threshold on the residuals for a fit. Another way to greatly reduce the number of candidate alignments between two 3D point clouds is to only match points with complementary labels. If we consider a site that has each third of its points colored with a distinct color, then based on the number of color bins alone, the number of possible alignments to consider is reduced by about 100 (10 color bins for each sites). Another heuristic is to only consider those combinations for which the edges between the three points meet some problem specific geometric criteria. In practice, such heuristics have been used to reduce the average number of matches when a polar query site had 30 points and a dataset site has 50 points to about 2000 poses. However, worst case performance occurs when all the points in both point clouds have the same label; the problem reduces to the unlabeled case where the query cloud has M points and the dataset cloud has N points giving $O(N^3M^3)$ candidate alignments (if we disregard geometric features).

In particular, heuristics are used to bound the distances between the three points and each point can have one of three labels. Each set of three points is considered as the vertices of a triangle. The considered features of a triangle are:

- The perimeter (sum of edge lengths)
- The longest edge length

- The shortest edge length

The bounds on the features are:

- Perimeter in $[9, 13]$ Å
- Longest edge length in $[3.5, 4.5]$ Å
- Shortest edge length in $[1.8, 3.5]$ Å

These bounds were chosen as compromise between the number of alignments to consider and the accuracy of the candidate alignments.

Our implementation uses a histogram with overlapping bins to group the query triangles by the colors of their vertices and triangle features. The histogram allows one to immediately disregard dataset triangles with incorrect color combinations or unlikely geometry, and to concentrate on the pairs of triangles that have a higher probability of a match (i.e. smaller residuals). If a bin exists for a dataset triangle, then, for each query triangle in the bin, determine which of the six permutations of corresponding points are valid with respect to point color.

Definition. The **distance matrix error** (DME) is the weighted root mean squared differences of lengths of the corresponding edges

Definition. Weighted least squares error is the weighted average of the Euclidean error between corresponding points [1].

For each valid permutation, compute the weighted Distance Metric Error (DME). If the best DME is within 0.3 Å use the corresponding permutation to assign the point correspondences used to compute the weighted least squares error fit. If the weighted least squares error fit is within 0.3 Å , keep the computed transformation (rotation and translation) as a candidate alignment.

Algorithm 2 An algorithm to populate a four dimensional histogram of all possible triangles for one point labeled point cloud. One level is all possible combinations of three vertex labels. The other three levels are one for each of the triangle features.

Require: A labeled point cloud with N points
  Initialize a 4D array for the bins B
  for all 3 point combinations of the N site points (triangles) **do**
    Form a Δ with the 3 points as its vertices
    Compute the lengths of the edges and the sum of the lengths (perimeter of Δ).
    Sort the point labels to get a unique key k based on the label of the points
    Place Δ in the bin for k, perimeter, longest side, and shortest side
    Place Δ in the immediate neighbors of the perimeter, longest side, and shortest side bins
  end for

Algorithm 3 An algorithm to enumerate all acceptable triangle matches between two labeled point clouds with directions.

Require: Query point cloud's 4D histogram (algo 2).
Require: Dataset's labeled point cloud with directions
Require: List L to store candidate alignments
  for all triangle a of the M dset site points **do**
    Compute label key k, longest side l, shortest side s, and perimeter p
    Get the bin for the current triangle's features $b := B[k][l][s][p]$
    if b is empty **then**
      continue
    end if
    for all query triangles t in bin b **do**
      enumerate valid permutations between a and t with respect to point labels
      for all valid permutation **do**
        Compute the weighted DME for this permutation
        if DME $\leq$ 0.3Å and DME is current best **then**
          save current permutation as best
        end if
      end for
      if a best permutation exists **then**
        Get the weighted least square error fit between the points (LSE)
        if LSE $\leq$ 0.3Å **then**
          append LSE transformation to L
        end if
      end if
    end for
  end for

3.1.3 Scoring and Ranking Alignments

In the previous section, we considered how to compute candidate alignments for a pair of binding sites. A alignment ranking method, typically called a scoring function, is needed to select the best candidate alignment between a pair of sites. Ideally, for a high-throughput method, the scoring function would be both computationally inexpensive and exhibit good ranking performance. Good ranking is needed since few, if any, users will want to consider more than one alignment per query, dataset pair in the results from a high-throughput object recognition method.

The ranking of binding site poses for sites with low sequence similarity is not necessarily straightforward. Predicting the ranking of small molecules versus a protein target by an estimate of the energetic favorableness of binding for each pair (binding affinity) [74] can be done by a scoring function that was trained to predict an experimentally observed measurement (e.g. binding affinity). However, the ranking of alignments between binding sites does not have a direct experimental counterpart. Because we don't have direct experimental data to design a scoring function, we must rely on heuristics based methods such as error of fit measurements. Commonly used error norms (ℓ_2, ℓ_1, ℓ_{inf}, etc.) in object recognition and protein structural alignment can be used to estimate the alignment accuracy. Although not knowing which error estimate best fits the binding site comparison problem may be an issue, a larger issue is that a comparison of the state-of-the-art scoring functions to predict binding affinity has shown that the current methods are not sufficient to correlate the predictions of protein-ligand binding affinity with the experimentally determined affinity [101]. Thus, it is naive to assume that a simple scoring of abstract features used to compare binding sites could correctly rank binding sites based on their affinity to a particular small molecule.

Although a scoring function designed to predict the similarity of two binding sites may not be able to accurately rank sites with respect to their binding affinity for a particular ligand , the scoring function should be able to give a good indication of how well two

sites are aligned. Determining which machine learning techniques is best suited to build a site similarity scoring is challenging. Based on numerous anecdotes and experience with the site similarity features, it is our experience that for two similar binding sites from distinct protein structures that the signal-to-noise ratio for even the best site alignment (with respect to site similarity features) is relatively low and the energy landscape is very noisy. This is due in part to the facts that the feature correspondences are short-ranged in nature, a relatively small number of distinct site similarity features are used, the relative placement error of the site points is large, and binding site features are relatively periodic (because binding sites are formed by amino acids). Given that more ”simple” techniques tend to be less affected by noise, and the fact that we would like to interpret the model used to make the predictions, linear regression was used to build the candidate scoring functions.

Given the relatively short range of the point correspondences, using linear regression to directly predict the error of alignment in the protein-ligand docking problem (i.e. docking RMSD) generally yields poor performance.

Definition. Binding site RMSD is the RMSD of a particular pose of binding site's points with respect to the reference pose of that binding site.

During analysis of protein-ligand docking scoring functions, Tonero noticed that plots of individual features versus alignment error (RMSD) showed a relationship similar to -1/RMSD [97]. Although a Gaussian function of RMSD appears to be a more accurate parametric form, in practice, linear regression functions to predict -1/RMSD exhibit similar performance and it seems to be easier for some to grasp a multiplicative inverse rather than a Gaussian function. The increase in alignment selection performance is due to the fact that linear regression relies on the assumption that a suitable parametric form is chosen for the predicted values, such that, the relationship between the independent variables (features) and the dependent variable is linear. For these reasons, the linear relationship between the aligned site features and alignment accuracy (as RMSD) is taken

to be -1/RMSD in this chapter.

Before building a scoring function using linear regression, one requires one or more site features that are viable for site similarity comparisons. The assumption is that if two similar sites are well aligned (i.e. close to the best alignment) that many of their similar site features should be brought into close proximity. Based on that assumption, a nearest neighbor method with a maximum distance of 1.5 Å is used to determine the best point correspondence for each point in the query site; the details may be found in Algorithm 4. The computed correspondences are ”one-sided” because of the partial matching nature of the problem, and the fact that the query site is the site for which we are seek the best partial match. The idea of computing and using ”one-sided” correspondences for the partial matching problem in object recognition has been formally presented and initially applied to face recognition by Bronstein and Bronstein [17].

The site alignment features are:

1. Closest polar sum: Sum of pairs of the closest polar points within 1.5 Å of each other for which the points in each pair are complementary. Each term in the sum is weighted by the dot product between the pair of vectors with a weight of zero if the dot product is less than zero.

2. Polar mismatch sum: Similar to the first sum, but this sum is a weighted count of the pairs of acceptor-donor mismatches.

3. Closest AA & DD sum: Similar to the first sum, but this sum does not include any doneptors [1] (either from the query or database site)

4. Closest doneptor sum: Similar to the first sum, but this sum includes only those terms where at least one of the points is a doneptor. Note: The first sum is equal to the sum of the third and fourth sums.

1 A point in the binding site where a hydrogen bond acceptor or donor could interact with the protein, is called a doneptor

5. Hydrophobic point count: Number of query hydrophobic points having the closest database point within 1.5 Å and being hydrophobic

6. Unsatisfied query polar count: Number of query polar points for which the closest point is within 1.5 Å and is hydrophobic.

3.1.3.1 Training data

As was mentioned in the background (Chapter 2), the machine learning approaches to building scoring functions to predict alignment quality require a set of training examples. The training data that we curated contains twelve distinct protein folds and their experimentally resolved structures. Each protein within a given fold is known to bind similar molecules (see Table 1). Each protein fold can be represented by one representative protein sequence and structure. To encourage diversity between folds, the datasets were constructed such that, the pairwise sequence identity of any two fold representatives is less than 25 percent and the class of small molecules bound by each fold has substantial differences. Two protein databases, DSSP [86] and FSSP [46], were used so that the sequences of the proteins within any given fold provide a reasonable coverage of the sequence identity space with respect to that fold's representative sequence. To that end, a histogram of the sequence space of each fold was used as a guide to partition the sequence space into bins with $[0, 25\%]$, $(25, 50\%]$, and $(50, 75\%]$ sequence similarity with respect to the fold representative. The goal was to have at least one example from each bin for each fold. As is frequently the case with actual data, a number of the 12 protein folds do not exhibit an adequate cover of the sequence space either due to the actual distribution of protein sequences in that fold or the sequence distribution of proteins with resolved structures. In such cases, the bin boundaries were relaxed with a goal of four structures per fold. The resulting training sets may be found in Table 1.

Algorithm 4 A way to estimate the features in common between two aligned site, and count the number of query polar points that do not have a correspondence.

Require: A dataset set, query site, and alignment between their labeled point clouds with directions

```
Initialize hbond_sum, doneptor_sum, AA_DD_sum, mismatched_hbond_sum, hphob_count, unsat_polar_count
for all X in query_site.hbond_pts do
  A := closest hbond_pt in dset_site;      d_A := dist(X.pos, A.pos)
  B := closest hphob_pt in dset_site;      d_B := dist(X.pos, B.pos)
  if d_A ≤ 1.5 and d_A ≤ d_B then
    dot_prod := A.dir ∘ X.dir
    if dot_prod > 0.0 then
      if A and B have compatible colors then
        hbond_sum += dot_prod
        if A or B is a Doneptor then
          doneptor_sum += dot_prod
        else
          AA_DD_sum += dot_prod
        end if
      else
        mismatched_hbond_sum += dot_prod
      end if
    end if
  else if d_B ≤ 1.5 then
    unsat_polar_count += 1
  end if
end for
for all X in query_site.hphob_pts do
  A := closest hbond_pt in dset_site;      d_A := dist(X.pos, A.pos)
  B := closest hphob_pt in dset_site;      d_B := dist(X.pos, B.pos)
  if d_B ≤ 1.5 and d_B ≤ d_A then
    hphob_cont += 1
  end if
end for
F := [ hbond_sum, doneptor_sum, etc. ]
return W^t F  # W is the weight vector determined by linear regression
```

Table 1: Twelve protein families used to train the SimSite3D alignment and site similarity scoring function. The protein structures in each family were aligned by Dali to the first member in their family. The Z-score is the Dali structural score for the alignment. The RMSD is the CA RMS between the pairs of aligned structures. Dali gives a measure of the sequence identity (%id) between the aligned proteins and the number of residues (nres) used in the alignments are provided to help gauge the significance of the sequence scores. The ligand column notes three character PDB code for the ligand bound in the binding site. Note that structures determined by NMR do not have resolution or R-factor values.

PDB	Ligand	Source	Res Å	R-factor	Protein	Z-score	RMSD	%id	nres
GTP-binding proteins; G(*) α subunits of transducins									
1got A	GDP	B. taurus & R. norvegicus	2.0	0.21	Chimera GT-α & GI-α1	100	0.0	100%	338
1tnd A	GSP	B. taurus	2.2	0.19	GT-α	43.4	1.3	87%	338
2bcj Q	GDP	M. musculus & R. norvegicus	3.1	0.24	Chimera GQ-α & GI-α1	36.4	1.8	52%	337
2ihb A	GDP	H. sapiens	2.7	0.21	GK-α	41	1.5	71%	337
DNA ligases; NAD+ dependent (adenylation domain)									
1ta8 A		E. faecalis v583	1.8	0.20	DNA ligase	100	0.0	100%	322
1b04 A		B. stearothermophilus	2.8	0.23	DNA ligase	42	1.4	59%	311
1zau A	AMP	M. tuberculosis	3.2	0.25	DNA ligase	29.2	2.9	40%	311
Aspartate transcarbamoylase catalytic subunits (atases)									
2air A	_CP & AL0	E. coli	2.0	0.24	Atase	100	0.0	100%	310
2be7 A		M. profunda	2.9	0.21	Atase	47.2	0.9	74%	307
1ml4 A	PAL	P. abyssi	1.8	0.18	Atase	39.6	2.4	52%	295

Table 1: (cont'd)

PDB	Ligand	Source	Res Å	R-factor	Protein	Z-score	RMSD	%id	nres
Carboxypeptidases and precursors (inactive carboxypeptidases)									
1dtd A		H. sapiens	1.7	0.19	A2	100	0.0	100%	303
1pca A		S. scrofa	2.0	0.20	A1	51.7	0.5	64%	301
1zli A		H. sapiens	2.1	0.16	B	50.8	0.8	47%	302
1obr A		T. vulgaris	2.3	0.15	T	41.9	1.4	33%	290
FKBP12s (3fap & 1c9h) and FKBP-like peptidyl-prolyl cis-trans (1fd9 & 1ix5) isomerases									
3fap A	ARD	H. sapiens	1.9	0.21	FKBP12	100	0	100%	107
1c9h A	RAP	H. sapiens	2.0	0.21	FKBP12.6 (lung)	22	0.7	83%	107
1fd9 A		L. pneumophila	2.4	0.23	MIP[a]	16.1	1.4	35%	104
1ix5 A		M. thermo-lithotrophicus			FKBP	10.7	1.7	31%	88
Ferredoxin-NADP(H) oxidoreductases (FPR)									
2bgi A	FAD	R. capsulatus	1.7	0.22	FPR	100	0.0	100%	272
1a8p A	FAD	A. vinelandii	2.0	0.21	FPR	38.3	1.2	53%	253
1fdr A	FAD	E. coli	1.7	0.18	FPR	31.2	1.8	33%	237
Peptidyl-prolyl cis-trans isomerases									
1pin A		H. Sapiens	1.4	0.22	pin1	100	0.0	100%	163
1j6y A		A. thaliana			pin1	11.7	3.5	49%	113
1jnt A		E. coli			parvulin	10.9	2.1	37%	75
1fjd A		H. Sapiens			parvulin-like	8.5	3.4	36%	111

a) MIP: macrophage infectivity potentiator

Table 1: (cont'd)

PDB	Ligand	Source	Res Å	R-factor	Protein	Z-score	RMSD	%id	nres
Racemases									
1jfl		P. horikoshii	1.9	0.19	aspartate racemase	100	0.0	100%	
1b74	DGN	A. pyrophilus	2.3	0.22	glutamate racemase	17.1	3.5	21%	
CMP * synthetases									
1eyr	CDP	N. meningitidis	2.2	0.19	CMP acylneuraminate synthetase	100	0.0	100%	
1qwj	NCC	M. musculus	2.8	0.24	CMP acetylneuraminic acid synthetase	26.6	2.7	25%	
Transcription regulatory proteins (receptor domains)									
1dbw	15P	R. meliloti	1.6	0.19	FIXJ receiver domain	100	0.0	100%	
1l5y		S. meliloti	2.1	0.18	DCTD receiver domain	17	2.3	37%	
3tmy		T. maritima	2.2	0.18	CHEY protein	16.7	2.0	30%	
1mvo		B. subtilis	1.6	0.19	PHOP receiver domain	16.8	2.2	23%	
Phosphatases									
1yn9	PO4	baculovirus[a]	1.5	0.17	RNA 5-phosphatase	100	0.0	100%	
1ohe	SEP	H. sapiens	2.2	0.21	cdc14b phosphatase	18.3	3.0	22%	
Structural genomics xray structures with unknown function									
1tuv A	VK3	E. coli	1.7	0.21	novel quinol monooxygenase	100	0.0	100%	
1x7v		P. aeruginosa	1.8	0.17	PA3566 protein	14.8	1.5	30%	
1y0h		M. tuberculosis	1.6	0.20	RV0793 protein	13.7	1.8	24%	

a) autographa californica nucleaopolyhedrovirus

3.1.3.2 Alignment sampling

To our knowledge, it is unknown if other research groups have used protein folds with a similar range of sequence and structural diversity to train their scoring functions to predict site alignment accuracy and binding site similarity. In fact, it is not clear how others have trained their scoring functions which makes it nearly impossible to reproduce their results without using their provided tools [53, 89, 92]. In our case, we used approximately 400 pairwise alignments between each pair of binding sites within each fold. Our working hypothesis is having a good coverage of the range of good to poor quality alignments from a set of binding sites of proteins that diverge in sequence space, helps to build a scoring function that can predict the quality of alignments of any two binding sites.

The error of a given alignment is approximated by the RMSD of the pose of the points in the query's site map with respect the reference pose for the query site. [2]. Because the proteins within a given fold share many structural features with the fold's representative, a structure based alignment tool, DALI [47], was used to align each protein structure to the representative structure. The DALI structural alignments are used as the reference (approximating zero error) alignments.

The training samples were computed as follows:

1. The alignments of the training samples were computed for each pair of binding sites within each training fold using the triangle matching method described in section 3.1.2 to list many candidate alignments that have at least three points with low error.

2. For each alignment, the six alignment features were computed as presented in Algorithm 4.

2 Note: A common practice in designing and comparing object recognition methods is to have a set of "gold standard" examples to benchmark new methods and compare the performance of competing methods. Unfortunately, there are no current tools or universally agreed upon standards to closely align binding sites (e.g. $\leq$0.1 Å RMSD). Thus, we prefer to call the pose that corresponds to an alignment with zero error as the reference pose rather than "gold standard"

3. The RMSD of each alignment was computed with respect to the reference pose (for the query site).

Thus, from a machine learning point of view, each training sample has six independent variables and one dependent variable and represents one alignment of one dataset site to one query site.

Using all of the computed alignments to train a scoring function is not feasible as the average number of alignments is 2000 per site pair. Another challenge is the pairs of more similar sites had many more candidate alignments than the pairs of less similar sites. For this reason, the training data was sampled using a stratified sampling method to get approximately the same number of good, fair, and poor alignments for each pair of binding sites within each fold. To help sample the data, the set of alignments for each pair of binding sites was partitioned into 11 bins in RMSD space $[0, \text{inf})$. The bin edges are $1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5$. The first bin is larger than the bins in the middle as it is difficult to get alignments with RMSD < 0.5Å for sites from distinct protein coordinates. The last bin is large since all alignments in that bin can be considered as equally poor with respect to the measured features and alignment error. The stratified sampling used was to randomly select (without replacement) 20 alignments from each of the first 10 bins and 200 alignments from the last bin. To alleviate the problem of pairs of sites with few good alignments and to balance the number of alignments in $[0.0, 3.0]$ with those in $[3.0, 5.5]$, the first five bins were sampled so that the cumulative total at each bin edge was as close as possible to the maximum allowed number of alignments at that bin edge (e.g. if only 15 alignments total were in bins 0 and 1, then if there are N alignments in bin 2, $\min(N, 3 * 20 - 15 = 45)$ were sampled from bin 2. Given such a set of samples, one can apply a variety of machine learning techniques to predict the error of alignment based on the six alignment features.

3.1.3.3 Scoring Function Forms

Given established machine learning techniques and the fact that there are six features per sample, which technique(s) to use can be considered as a personal preference. The reason is there is little previous knowledge about the data that can be used to prefer one prediction method over another. On the surface, the fact that we have thousands of samples and only six features implies that over-fitting is likely to be a small issue. However, the assumption that the samples are independent and identically distributed may not be reasonable for the site alignment features.

There are a number of considerations due to the nature of the problem. There is an average error of $\sim$0.2 Å in the relative positions of the site points because of the relative error in atomic positions (i.e. experimental/model error). The reference alignments of the sites have an average global reference error that is at least 0.5 Å RMSD due to the error in structural alignment methods and the relative location of the binding site with respect to the protein backbone. We seek a reliable ranking of those samples that have the sites well aligned (under 2.0 Å RMSD), but for the samples that correspond to poorly aligned sites we only seek to recognize that they are poorly aligned. Finally, given the exploratory nature of our work, it would be very beneficial to be able to interpret the scoring function's form and performance. Given these considerations, linear regression is a good first choice to predict alignment quality based on site features.

Linear regression was used to train 27 distinct scoring functions to predict alignment quality. The number of scoring functions is due to the facts that one of the terms is the sum of two others, and manually selecting biologically meaningful combinations of terms was preferred over statistical feature selection techniques. The independent features used in the scoring functions are listed in Table 2 where feature 0 is the constant term and the other features have been listed previously. The dependent variable or feature we seek to predict is the RMSD of alignment. Based on previous experience and the reasons given previously it is advantageous to transform the RMSD to -1/RMSD. The reason is

the relationship between the features and RMSD is better described by -1/RMSD than a straight line [97].

Table 2: Combinations of site similarity features for linear regression

SF #	terms	SF #	terms
1	0,1	15	0,3,5
2	0,2	16	0,2,3,5
3	0,3	17	0,2,3
4	0,4	18	0,1,5,6
5	0,5	19	0,1,2,5,6
6	0,6	20	0,1,2,6
7	0,1,2,3,4,5,6	21	0,3,4,6
8	0,1,5	22	0,3,4,5,6
9	0,1,2,5	23	0,2,3,4,5,6
10	0,1,2	24	0,2,3,4,6
11	0,3,4	25	0,3,5,6
12	0,3,4,5	26	0,2,3,5,6
13	0,2,3,4,5	27	0,2,3,6
14	0,2,3,4		

Given the resources required to build the training and testing datasets, a separate validation dataset was not constructed. Instead, dataset cross-validation was used to select the best performing scoring function. Specifically, 12 runs of training and validating the scoring functions were performed with a different training dataset reserved for validation each time. To keep the comparisons fair, the same stratified sampling was used for all 12 runs and all scoring functions. Matlab's implementation of LSQR (an iterative solver) was used to find a numerical solution to a weight vector that minimized the least squared error (i.e. determine weights that solved the linear regression problem).

In order to reduce the effects of sampling artifacts, the entire training and validation was performed for 10 stratified samples for a total of 120 sets of weights for each scoring function. To reduce the potential for variance, the final scoring functions are the result of stacking the 120 scoring functions by averaging the weights. The final scoring function with the best average performance was chosen as the scoring function of choice.

3.1.4 Scoring Function Training and Validation Results and Analysis

The process of determining which scoring function performs the best is not straightforward given the noise level of the data, the desire for a high quality predictions in $[0.0, 2.0]$ Å RMSD of alignment, and being less concerned about the actual fit for $(2.0, \text{inf})$ Å RMSD. The textbook method of picking the scoring function with the smallest error of fit [42] is not applicable because all of the fits are poor due to noisy data and the unknown parametric form of the data. Also, the smallest global error of fit does not necessarily correspond the smallest error of fit in the range of $[0.0, 2.0]$ RMSD. Since the goal is to have a scoring function that performs well at ranking, the RMSD of the best scoring alignment per pair of binding sites in each of the validation steps was saved. The performance of each scoring function was estimated by the average of the RMSD values of the best scoring alignments over 120 validation steps (see Table 3).

Table 3: Mean, median, and standard deviation of the sitemap RMSD of the best ranked alignment per pair of validation set binding sites across 120 runs. Computed using the "hold one dataset out" method and across ten stratified samplings.

SF #	mean	median	stdev	SF #	mean	median	stdev
12	2.98	1.81	2.94	7	3.72	1.94	4.27
8	3.01	1.83	2.80	23	3.72	1.94	4.27
15	3.08	1.66	3.11	10	3.84	2.43	3.62
18	3.13	1.81	3.17	19	3.86	2.03	4.24
22	3.22	1.66	3.42	14	3.87	2.04	4.14
1	3.25	1.88	3.24	17	3.88	1.92	4.23
11	3.25	1.84	3.44	27	3.91	1.92	4.33
25	3.28	1.66	3.49	24	4.00	2.04	4.40
3	3.48	1.76	3.82	20	4.12	2.34	4.22
13	3.51	1.92	3.52	5	4.12	3.33	3.44
16	3.54	1.92	3.58	6	4.65	3.40	4.32
21	3.58	1.84	4.04	2	5.41	4.58	4.11
26	3.69	1.92	4.29	4	6.78	6.79	3.98
9	3.69	2.20	3.33				

Looking at the scoring function validation data in Table 3 one can make several re-

marks about the alignments chosen by the scoring functions. First, no scoring function performed particularly well since the best average RMSD of alignment for the best scoring alignments is about 3.0 Å ; this means that for many of the pairs of sites, the best scoring alignment is one with a relatively high alignment error ($>$ 2.0Å RMSD). Second, the median RMSD for scoring function 12 is 1.81 Å which is about 1 Å RMSD less than the mean and indicates that the average is shifted higher by a number of outliers with high alignment error. Thirdly, the relatively large standard deviations also point to outliers with very large alignment errors because 0 Å RMSD is the minimum. Finally, scoring function 12 was chosen as the scoring function of choice because it has the best average RMSD and the second smallest standard deviation.

Table 4: The average and standard deviation of the weights for three of the scoring functions listed in Table 3. The sample size is 120 for each weight. The weight numbers correspond to the terms listed in the previous section

Term	SF # 12		SF # 8		SF # 1	
C	-0.0662	0.0149	-0.0524	0.0169	-0.0589	0.0155
1			-0.0189	0.0013	-0.0197	0.0013
3	-0.0208	0.0018				
4	-0.0088	0.0034				
5	-0.0023	0.0019	-0.0021	0.0019		

Looking at the standard deviation of the weights relative to the average weight we see several points of interest. The hydrogen bonding terms that include the acceptor-acceptor matches and donor-donor matches (terms 1 and 3) have a standard deviation that is about 10 percent of the average weight, and this indicates that the acceptor-acceptor and donor-donor point matches are consistently considered as being favorable. On the other hand, the standard deviation of the weight assigned to the hydrophobic term (term 5) is approximately of the same magnitude as the weight itself and indicates that in a number of training cases the hydrophobic weight was almost zero or even positive.

3.1.5 Score Normalization

One problem with global averaging schemes, such as linear regression, is the form of the scoring function is a constant weight times each term. When the features are computed as in Algorithm 4 and are not scaled, query objects with fewer high-value points have a smaller range of possible scores than query objects with more high-value points. In term of binding sites, those sites with fewer hydrogen bond site points will have, on average, a less favorable score than sites with more hydrogen bond site points. Such a "feature" makes it difficult to set one reliable threshold value for a score to be significant and to compare scores between different query objects with respect to the same dataset object.

To address this problem, each query site is compared to the same dataset of 140 binding sites from structurally diverse proteins (i.e. each protein is from a pairwise distinct fold). The score distribution of the best score per site pair for one query site can be roughly approximated by a Gaussian distribution. The mean and standard deviation of the Gaussian for a given query site is estimated by the mean and standard deviation of the sample population (140 scores). The raw scores for a query site are normalized by subtracting the query's mean score and then dividing by the standard deviation.

Definition. **Normalized score** is the number of standard deviations above or below the mean score.

The advantage of score normalization is a score significance threshold of 1.5 standard deviations better than the mean was found to strike a delicate balance between the number of false positives and the number of interesting true positive hits (for our implementation).

3.2 Results

One way to test the soundness of a scoring function is to apply it to several challenging test datasets. In this section, our alignment and scoring method is evaluated as it

is expected to be used in practice. The candidate alignments are found using the previously explained method, and then the alignments are ranked using scoring function 12 (see Tables 2, 3, 4) from the previous section. The alignment and scoring methods are implemented in version 3.3 of our software package SimSite3D.

3.2.1 Test Dataset

To test our method, we have constructed five unbiased test sets. These test sets are unbiased because they were constructed from classes of small-molecules and protein folds that are distinct from those of the 12 training datasets. A comparison study of SimSite3D and two competing methods is given for one of the test datasets. Because of the dataset sizes and the fact that users are expected to look only at the best scoring alignment per pair of sites, all analysis is with respect to the best scoring alignment per pair of sites.

3.2.1.1 Protein Kinases and other Proteins Binding Adenine

Kinases have been a frequent drug target, and are an important class of proteins in pharmaceuticals and understanding protein signaling and pathways. This dataset is particularly challenging as the protein kinases in the set diverge in structure and sequence, the non-kinase structures are structurally distinct from kinases, and crystallographic evidence for water mediated hydrogen bonds exists in most of the structures.

Table 5: Adenine binding proteins: two-thirds of the sites are from serine/threonine kinase, one is from a tyrosine kinase, and the remainder of the sites are from a diverse set of non-kinase proteins that bind adenine.

Abbrev.	PDB	Adenine ligand	Source	Res.	R factor	Protein
Hs CDK2	1b38	ATP	H. sapiens	2.0	0.18	Cyclin dependent kinase 2
Hs GSK3	1j1b	ANP	H. sapiens	1.8	0.22	Glycogen synthase kinase-3 β (gsk3 β or τ kinase)
Hs PIM-1	1yxt	ANP	H. sapiens	2.0	0.18	Pronto-oncogene kinase pim-1 (Unique: has Pro at 123)
Hs CDK7	1ua2	ATP	H. sapiens	3.0	0.22	Cyclin dependent kinase 7
Hs Aurora-A	1ol5	ADP	H. sapiens	2.5	0.19	ipl1-related kinase 1
Mm PKA	1u7e	ANP	M. musculus	2.0	0.17	cAMP dependent kinase (pka Cα)
Hs IRK	1ir3	ANP	H. sapiens	1.9	0.19	Insulin Receptor
Hs PDK1	1h1w	ATP	H. sapiens	2.0	0.20	3-Phosphoinositide dependent kinase-1
Hs ATK2	1o6l	ANP	H. sapiens	1.6	0.20	Protein kinase B
Hs CK2ii	1jwh	ANP	H. sapiens	3.1	0.27	Casein kinase II
Mm TRP	1iah	ADP	M. musculus	2.4	0.22	Transient receptor potential
Hs SRPK1	1wbp	ADP	H. sapiens	2.4	0.23	S/R rich protein specific kinase
Mm EphB2	1jpa	ANP	M. musculus	1.9	0.23	EPHB2 receptor tyrosine kinase
Hs MTAP	1cg6	MTA	H. sapiens	1.7	0.20	Methylthioadenosine phosphorylase
Hs HSP90	1byq	ADP	H. sapiens	1.5	0.19	Heat shock protein 90
Mc -MMC	1aha	ADE	M. charantia	2.2	0.18	Alpha-momorcharin
Hs HSP70	1s3x	ADP	H. sapiens	1.8	0.20	Heat shock protein 70
Ss F16P	1frp	AMP	S. scrofa	2.0	0.19	Fructose-1,6 bisphosphatase
Pf PHBH	2phh	ADP	P. fluorescens	2.7	0.17	P-hydroxybenzoate hydroxylase

3.2.1.2 Proteins that can bind Ligands Containing Pterin

The proteins in the folate biosynthesis pathway bind ligands that contain a fused two hexagonal ring system called pterin. Of these proteins, 6-hydroxymethy-7,8-dihydroxypterin pyrophosphokinase (HPPK) is of considerable interest to our lab as a potential drug target for Yersinas Pestis (the bacteria responsible for the plague). It would be helpful if we could characterize the pterin binding sites in other protein folds with respect to the pterin binding site in HPPK. This dataset has representatives from four distinct protein folds that each have a site that binds pterin.

Table 6: Pterin binding proteins: proteins that natively bind a ligand containing the pterin rings system. Four distinct protein families are represented DHFRs, HPPKs, aromatic amino acid hydroxylases, and DHPSs

Abbrev.	PDB	Pterin ligand	Other ligand	Source	Res.	R factor	Protein
Hs DHFR[a]	1u72	MTX	NDP	H. sapiens	1.9	0.16	DHFR
Pc DHFR	2fzh	DH1	NAP	P. carinii	2.1	0.25	DHFR
Ch DHFR	1qzf	FOL	CB3 UMP UDP	C. hominis	2.8	0.23	DHFR
Mt DHFR	1df7	MTX	NDP	M. tuberculosis	1.7	0.19	DHFR
Pf DHFR	1j3i	WRA	UMP NDP	P. falciparum	2.3	0.19	DHFR portion of DHFR-TS
Gg DHFR	1dr1	HBI	NAP	G. gallus	2.2	0.14	DHFR
Ca DHFR	1aoe	GW3	NDP	C. albicans	1.6	0.16	DHFR
Tm DHFR	1d1g	MTX	NDP	T. maritima	2.1	0.20	DHFR
Yp HPPK[b]	2qx0	PH2	APC	Y. pestis	1.8	0.23	HPPK
Sc HPPK	2bmb	PMM		S. cerevisiae	2.3	0.18	HPPK portion of HPPK-DHPS
Ec HPPK(t)	1q0n	PH2	APC	E. coli	1.3	0.12	HPPK (ternary complex)
Ec HPPK(b)	1rb0	HH2		E. coli	1.4	0.16	HPPK (binary complex)
Hi HPPK	1cbk	ROI		H. influenzae	2.0	0.16	HPPK
Hs PAH	1mmk	H4B	TIH	H. sapiens	2	0.20	Phe hydroxylase
Hs TPH	1mlw	HBI		H. sapiens	1.7	0.21	Trp hydroxylase
Rn TH	2toh	HBI		R. norvegicus	2.3	0.21	Tyr hydroxylase
Cv PAH	1ltz	HBI		C. violaceum	1.4	0.16	Phe hydroxylase
Sc DHPS[c]	2bmb	PMM		S. cerevisiae	2.3	0.18	DHPS portion of HPPK-DHPS

a) DHFR: dihydrofolate reductase
b) HPPK: 6-hydroxymethy-7,8-dihydroxypterin pyrophosphokinase
c) DHPS: dihydropteroate sythase

3.2.1.3 Glutathione-S transferases

The glutathione-S transferases were added as they are an important group of proteins and contain a polar binding site and a hydrophobic binding site. This dataset is used twice. Once for the Glu binding site of glutathione in the structural diverse portion of the dataset and once for the hydrophobic binding sites (Hsite) in all of the structures. The glutathione binding site is relatively conserved across the species and protein isoforms. The Hsite for the H. sapiens *pi*-class structures has local changes due to different ligands bound, and the Hsites for the diverse set are very different and in most cases bind very different classes of ligands. Thus the Hsite portion of the dataset can be used to illustrate the handling of local changes in the same binding site, and very large changes in the binding site between different species and protein isoforms [3].

3 Proteins within a species can differ somewhat in sequence and structure depending on the tissues or environment in which they are present

Table 7: A test dataset of glutathione-S transferases (GSTs). Both the hydrophobic sites (Hsites) and the Glu part of the glutathione sites are used in this dissertation. The H. sapiens π-class structures have a variety of inhibitors bound in the Hsite and can be used to gauge the sensitivity of SimSite3D to small changes in the binding site of the same protein. The structures from other species have a glutathione or an analog bound in the glutathione pocket.

Abbrev.	PDB	GSH ligand	Hsite ligand	Source	Res.	R factor	Protein
Hs π - SAS	13gs	GTT	SAS	H. sapiens	1.9	0.19	π class GST
Hs π -	10gs	Glu	PG9 BCS	H. sapiens	2.2	0.18	π class GST
Hs π - EAA	11gs	GTT	EAA	H. sapiens	2.3	0.21	π class GST
Hs π - BSP	19gs	GTT	BSP	H. sapiens	1.9	0.21	π class GST
Hs π -	1aqx	ILG TNB GLY	ILG TNB GLY	H. sapiens	2.0	0.20	π class GST
Hs π - CBD	20gs		CBD	H. sapiens	2.5	0.23	π class GST
Hs π - EAA	2gss		EAA	H. sapiens	1.9	0.21	π class GST
Hs π - GPR	2pgt	GPR	GPR	H. sapiens	1.9	0.18	π class GST
Hs π - CBL	3csj		CBL	H. sapiens	1.9	0.18	π class GST
Hs π - EAA	3gss	GTT	EAA	H. sapiens	1.9	0.21	π class GST
Hs π - GTX	4gss	GTX	GTX	H. sapiens	2.5	0.20	π class GST
Hs π - GTX	9gss	GTX	GTX	H. sapiens	2.0	0.19	π class GST
Mm π	1glp	GTS		M. musculus	1.9	0.17	π class GST
Hs	1xw5	GSH		H. sapiens	1.8	0.21	class GST
Sp β	1f2e	GTT		S. paucimobilis	2.3	0.20	β class GST
Hs ϵ	1pkw	GTT		H. sapiens	2.0	0.16	ϵ class GST
Hs PGDS	1iyi	GSH		H. sapiens	1.8	0.19	Prostaglandin D synthase
Ac θ	1jlv	GSH		A. cracens	1.8	0.22	ADGST1-3
Hs ω	1eem	GSH		H. sapiens	2.0	0.22	ω class GST
Mm α	1b48_A	HAG		M. musculus	2.6	0.24	MGSTA4-4
Rn M-κ	1r4w	GSH		R. norvegicus	2.5	0.20	Mitochondrial κ-class

3.2.1.4 Matrix Metalloproteinases

Given the prevalence of metal sites in proteins, we curated a dataset of proteins that use a metal ion to cleave other proteins. Somewhat to our surprise, it was observed that although the overall sequence and structure of the proteins diverges from that of collagenase, the peptide cleavage sites are structurally conserved and align very well using structure based tools (e.g. DALI and SSM).

Table 8: Peptide cleavage site of matrix metallo-proteinases (MMPs)

Abbrev.	PDB	Ligand	Source	Res.	R factor	Protein
Hs MMP1	1cgl	PHQ-ABU-Leu-Phe-EMR	H. sapiens	2.4	0.19	collagenase
Hs MMP8	1a85	HMI-DSG-DBP	H. sapiens	2.0	UNK	MMP-8
Hs MMP3	1b8y	IN7	H. sapiens	2.0	0.20	stromelysin 1 (MMP-3)
Hs MMP7	1mmr	SRS	H. sapiens	2.4	0.19	matrilysin (MMP-7)
Hs MMP10	1q3a	NGH	H. sapiens	2.1	0.28	stromelysin 2 (MMP-10)
Ss MMP1	1fbl	HTA	S. scrofa	2.5	0.22	collagenase (MMP-1)
Mm MMP11	1hv5	RXP	M. musculus	2.6	0.22	stromelysin 3 (MMP-11)
Ca HT-D	1atl	SLE-Tyr	C. atrox	1.8	0.16	atrolysin c form d
Sm SP	1af0	Leu-HMA	S. marcescens	1.8	0.18	serratia protease
Bt Thermo	1gxw	Val-Lys	B. thermo-proteolyticus	2.2	0.16	thermolysin

3.2.2 Test Dataset Results

We would like to have an estimate of the difficulty of aligning and assessing the similarity of the sites in the test datasets. To that end, Secondary Structure Matching (aka SSM or PDBeFold) [60] was used to compute the best pairwise alignment of the residues near the binding sites within each test dataset. Because SSM requires significant secondary structure features to align peptide fragments, a residue was considered near a binding site if any heavy atom in the residue is within 9.0 Å of any ligand heavy atom. Based on these residues, SSM provided a pairwise Q-score and sequence similarity score of the binding sites within each fold. The Q-score characterizes the structural similarity of the residues near the binding site; the sequence similarity characterizes the amount of sequence similarity near the binding sites. The SSM results are illustrated in Figure 8.

One can make several remarks about the binding site datasets based on the SSM results. There is little if any structural similarity between the DHFR, HPPK, and amino acid hydroxylase protein folds (Figure 8, B). The matrix metalloproteinases are relatively conserved except possibly for Bt Thermolysin (Figure 8, D). The kinases' and other proteins' adenine sites are in general less structurally conserved as can be seen by the more blue colors than for the other protein folds, and it is difficult or impossible for SSM to find structural alignments between the adenine sites in kinases and the other adenine binding proteins (Figure 8, A). The Glu pocket of the glutathione binding sites are structurally similar at about the same level as the adenine sites in the kinases except for the rat mitochondrial κ-class GST (Figure 8, C). The SSM results for the hydrophobic binding site of the GSTs are not presented as the atomic positions of α-carbons of the H sapiens *pi*-class structures are almost identical and the Hsites of the other isoforms are in general very distinct from each other. Given the SSM results, the advantage of a site alignment tool, such as SimSite3D, would be the ability to find significant hits in the regions where SSM was unable to find an alignment (besides providing a more detailed comparison of binding site features).

As is commonly reported with other binding site alignment tools [33, 89, 92], SimSite3D performs very well for the same binding site from proteins within moderately conserved protein folds. In addition, SimSite3D is able to find some closely aligned and significant scoring hits between HPPK and amino acid hydroxylase pterin binding sites (Figure 9, B). As mentioned previously, the peptide cleavage site for the MMPs is very highly conserved, and this is confirmed by the SimSite3D scores (Figure 9, C).

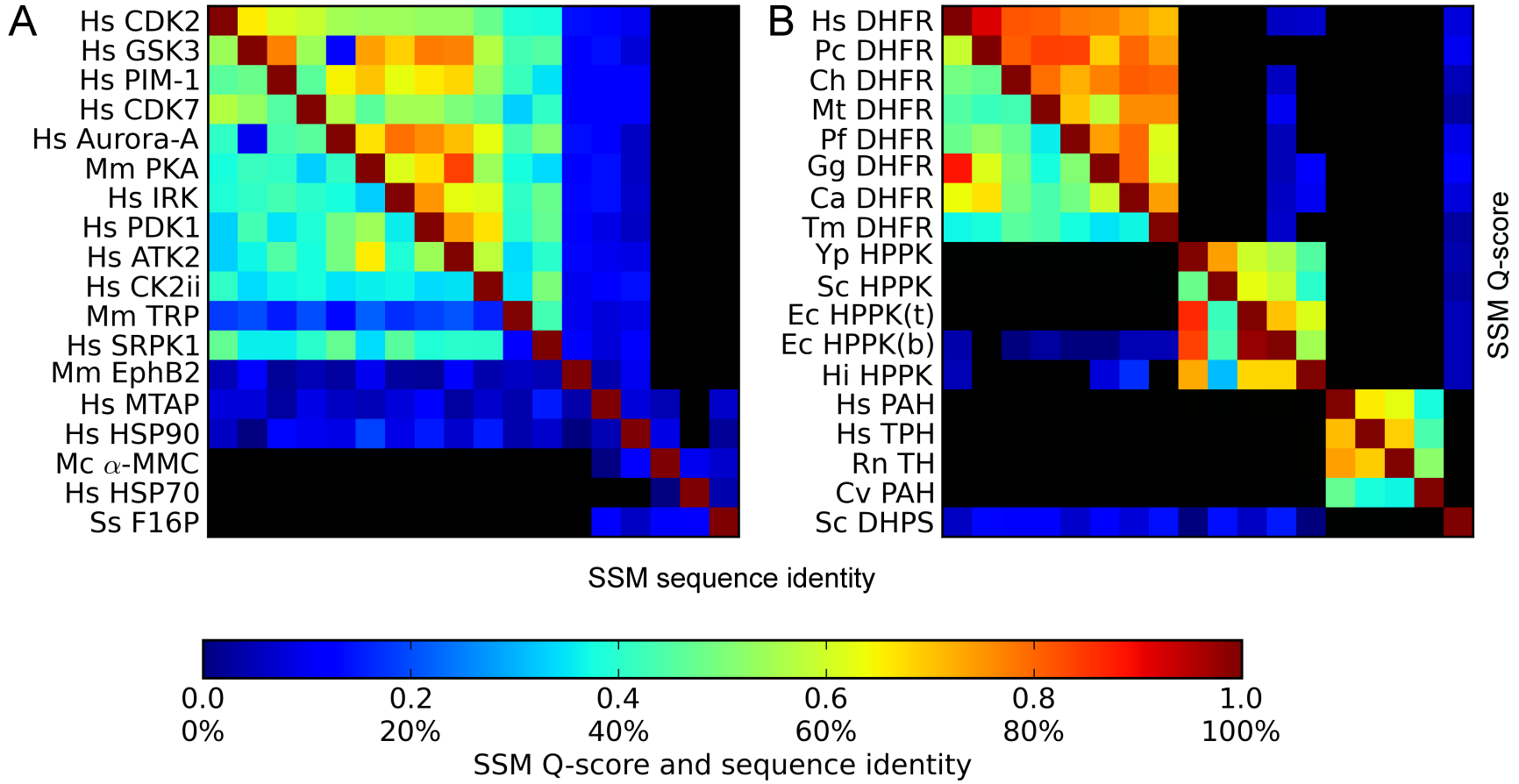


Figure 8: SSM score matrices of the best scoring pairwise alignment of the residues flanking the ligand binding sites (all residues within 9.0 of the ligand defining the binding site volume). Matrices A, B, C, D display the SSM results for the adenine (Table 5), pterin (Table 6), diverse GST (Table 7), MMP (Table 8) binding sites, respectively. The column labels are identical to the row labels. Within a matrix, a row corresponds to the results of one query site compared with all the sites in that dataset. Likewise, each column shows the similarity of one dataset site with respect to all the query sites (in that dataset). A black cell denotes that SSM was unable to find a structural alignment between the corresponding pair of sites.

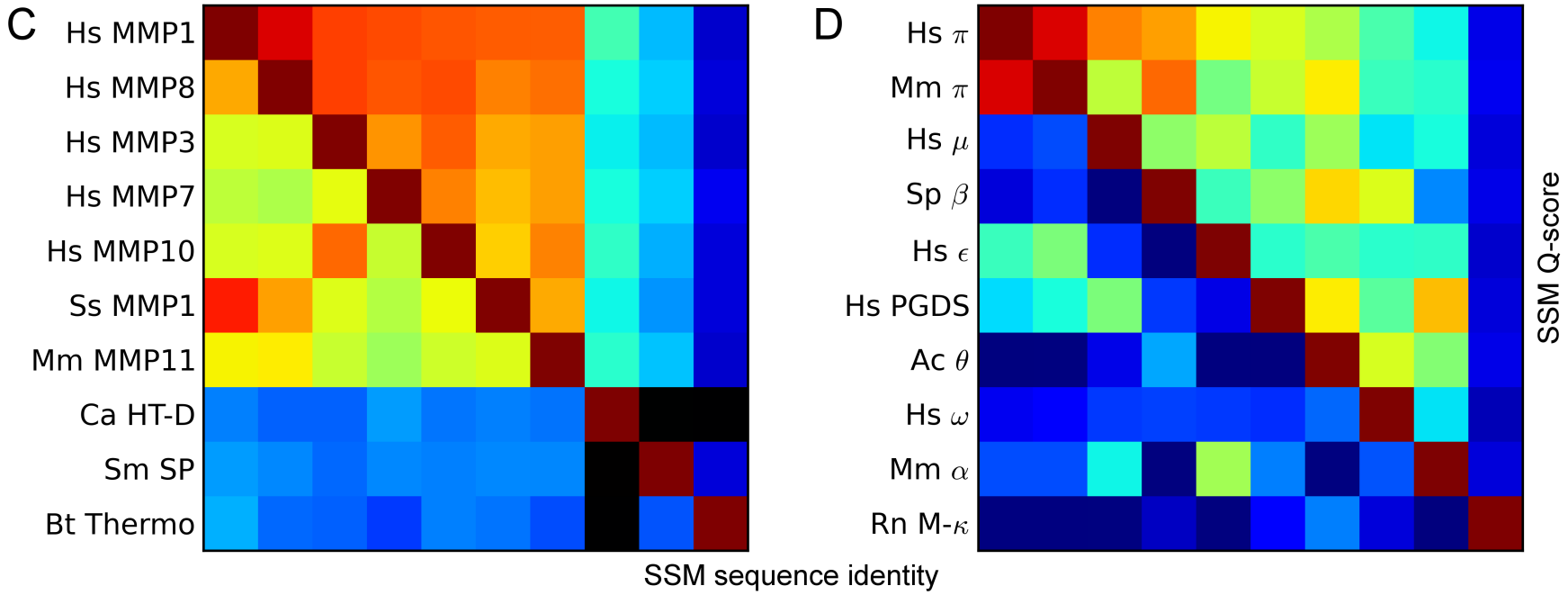


Figure 8: (cont'd) Since SSM scores are not necessarily symmetric, the values in each cell are the average of the corresponding SSM values when the pair switches which site is the query site. The lower triangles of the matrices show the SSM computed sequence identity near the ligand binding sites. The upper triangles show the SSM Q-scores for the secondary structure elements and the residues near the binding sites.

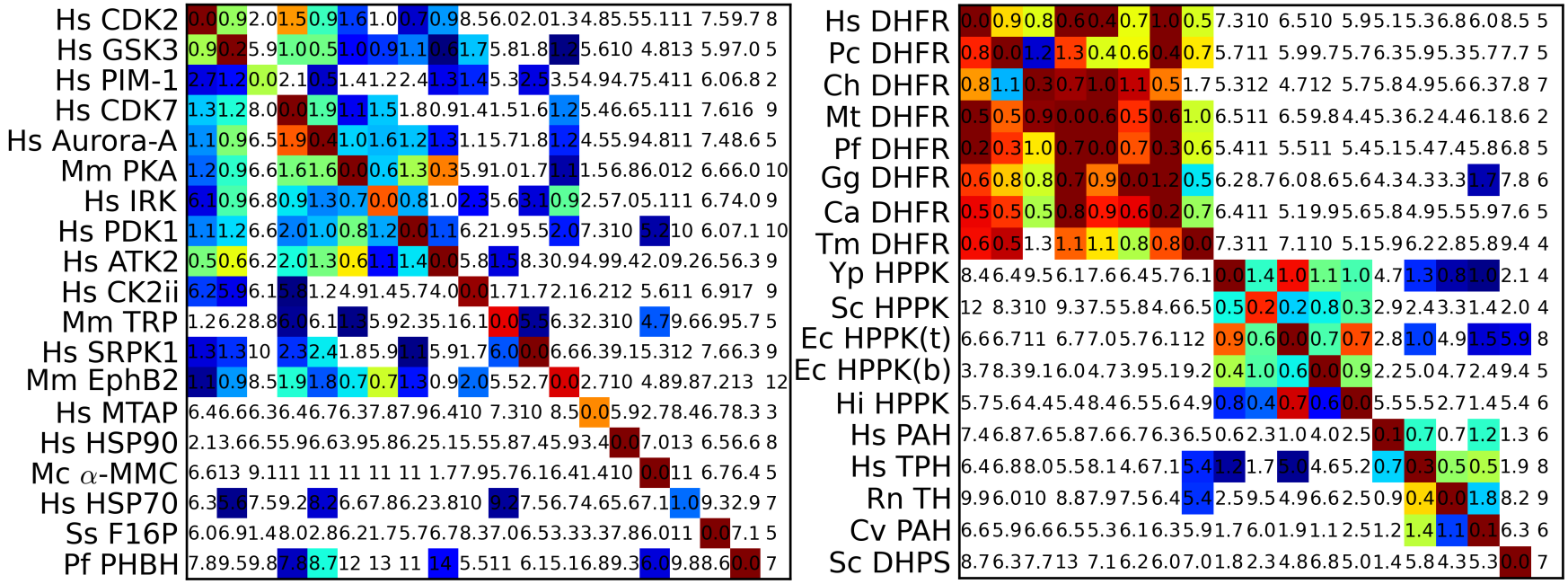


Figure 9: SimSite3D score matrices showing the score of the best scoring pairwise alignment of the query binding sites (site maps) to the dataset binding sites. The matrices are enumerated in the same manner as figure 8. The column labels are identical to the row labels except for the rightmost column. The rightmost column is the count of 140 diverse dataset sites that scored better than 1.5 standard deviations better than the average score. The scores of the hits for each row (1 query site) are scaled linearly to be in the range [self_score, -1.5] where self_score is the best possible score for the corresponding query pocket. The range [self_score, -1.5] is mapped linearly to the color bar. The color for a given score is found by computing the index for the score in the given color map. A black cell indicated that the best scoring alignment between the corresponding query pocket and dataset ligand site had a score worse than the threshold of -1.5. The number in a given cell is the RMSD of the best scoring alignment with respect to the reference alignment.

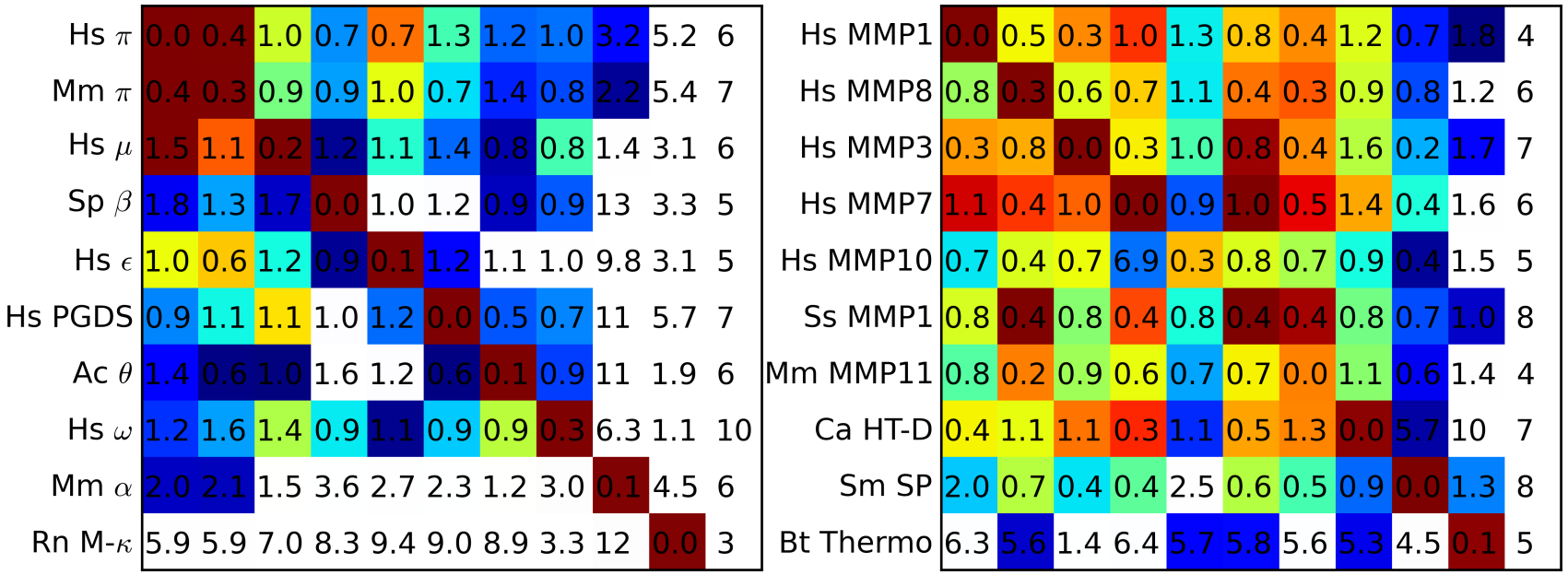


Figure 9: (cont'd)

3.2.3 Effects of Score Normalization

Score normalization has a significant impact on the performance of SimSite3D. As mentioned in the methods section, a score significance threshold of 1.5 standard deviations better than the mean was empirically determined to provide a good balance between finding interesting true positive hits and limiting the number of false positives.

Definition. A **true positive hit** is a valid match between a query site and dataset site that is correctly identified as a significant match by the selected scoring function.

Definition. A **false positive hit** is an invalid match between the query site and dataset site that is incorrectly identified as a significant match by the selected scoring function.

In addition, the normalized score performs much better (than the raw score) at predicting the error of site alignment. The advantage of using the normalized score to predict the error of site alignment can be visualized by ROC-like plots.

Here a brief definition of a ROC-like plot [4] is given; a more indepth introduction to ROC curves and analysis is given by Fawcett [32]. The goal is to show the interplay between the number of acceptable and poor site alignments as a function of site score. The data was compute as follows:

1. For each pair of query, dataset sites in the testing datasets, keep the best scoring alignment, its score, and RMSD.

2. Partition the set of alignments into two categories; acceptable and poor alignments based on a threshold of 2.0 Å RMSD [5].

4 The plots used are called ROC-like as the definition of ROC plots require percentages of the corresponding populations on both axes, and we prefer to see the number of samples on both axes. In addition, because of our emphasis on low error alignments, the area under the curve (AUC) is less relevant for our purposes.

5 An RMSD threshold of 2.0 Å to distinguish between acceptable and poor alignments is used since getting alignments under 1.0 Å RMSD is challenging, but for alignments over 2.0 Å the site feature will have incorrect correspondences and the computed score cannot be trusted.

3. Determine the score thresholds S, T at which no alignment met the score, all alignments met the score, respectively.

4. Partition the range $[S, T]$.

5. At each partition boundary compute the number of acceptable and poor alignments that meet the score threshold (that corresponds to the partition boundary), and plot the plot the number of good versus poor alignments.

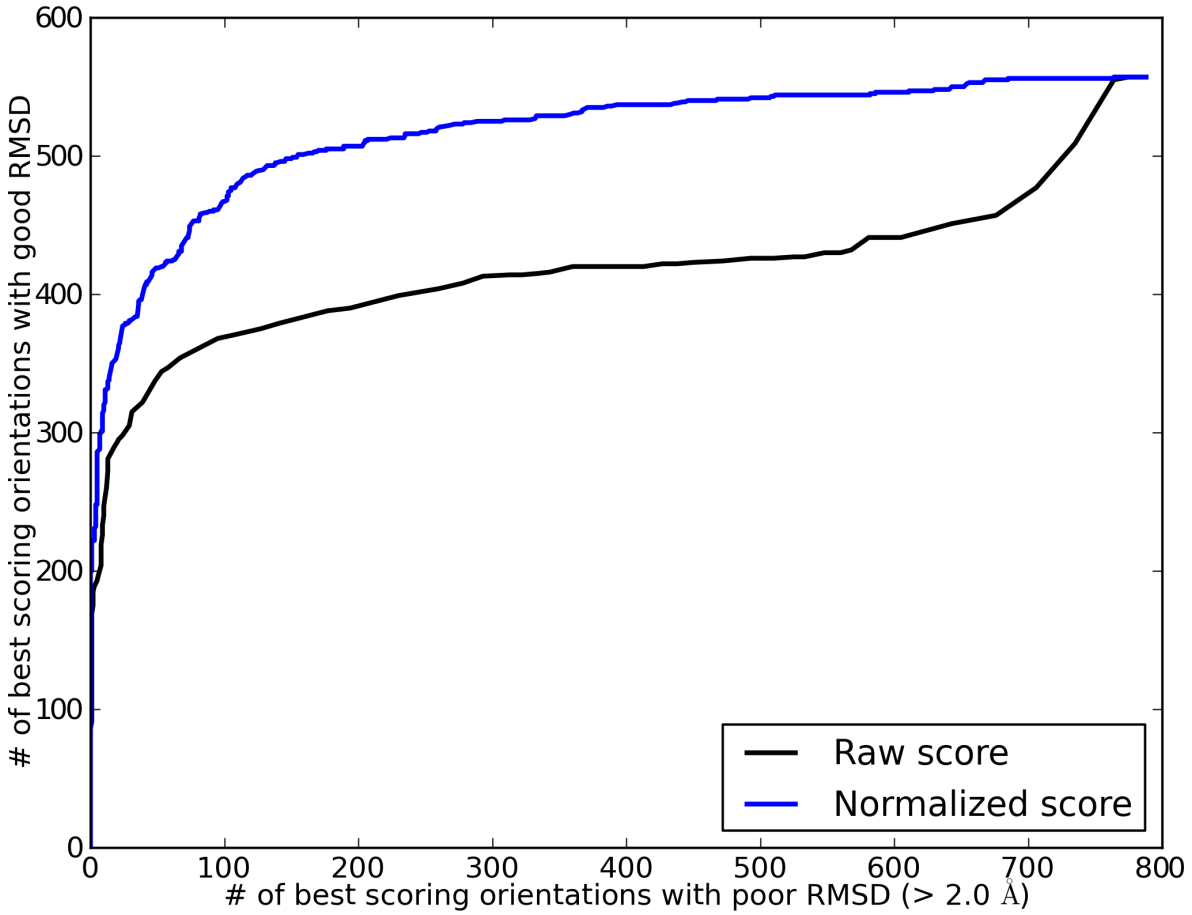


Figure 10: A ROC-like plot showing the advantage of using normalized site score thresholds rather than raw score thresholds for predicting the quality of site alignments. The plot data is the score and site RMSD corresponding to the best scoring alignment per pair of query, data sites in the testing datasets. The site score increases monotonically as one moves along a particular curve from the lower left corner to the upper right corner (a lesser score is more favorable). Thus, an ideal scoring method would exhibit a vertical line at 0 poor alignments and a horizontal line at the number of acceptable alignments.

One can use the ROC plot for alignment quality (Figure 10) to see that using the normalized score is beneficial. At a cost of 25, 50, 100 poor quality alignments, the normalized score gains approximately an additional 50, 75, 100 acceptable alignments (respectively) over the raw score.

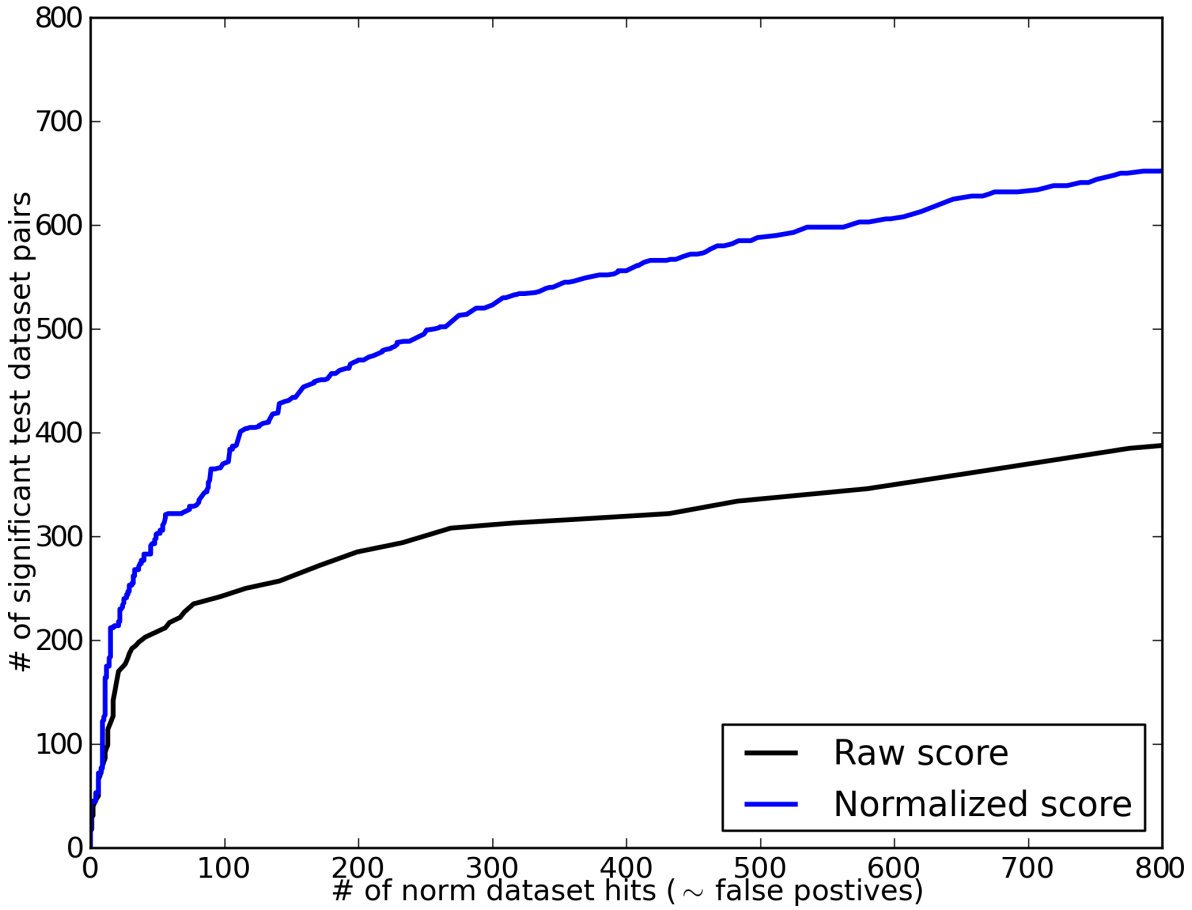


Figure 11: A ROC-like plot showing the ability of normalized site score to better distinguish between true positive and false positive hits. The plot data is the best score between each query site and the sites within the query's test set and between each query site and the sites in the normalization dataset. Here the normalization dataset is used as a proxy for the binding sites in the PDB. Therefore, an ideal scoring method would exhibit a vertical line at 0 norm dataset hits and a horizontal line at the number of test dataset hits.

Figure 11 shows that the normalized score gains approximately 150-200 true positives over the raw score between 100 and 300 normalization database hits. Note, as is commonly the case with current high-throughput protein computational chemistry tools, a high false positive rate is the price one must currently pay in order to find interesting

examples.

Figure 12 shows that normalizing the score has a significant impact on the overlap of the distributions of scores within the test folds and scores for test query sites versus the normalization database. An ideal method (function) would be one that could separate the two distributions. Although the overlap of the score distributions is still significant after normalization, the severity of the overlap is reduced.

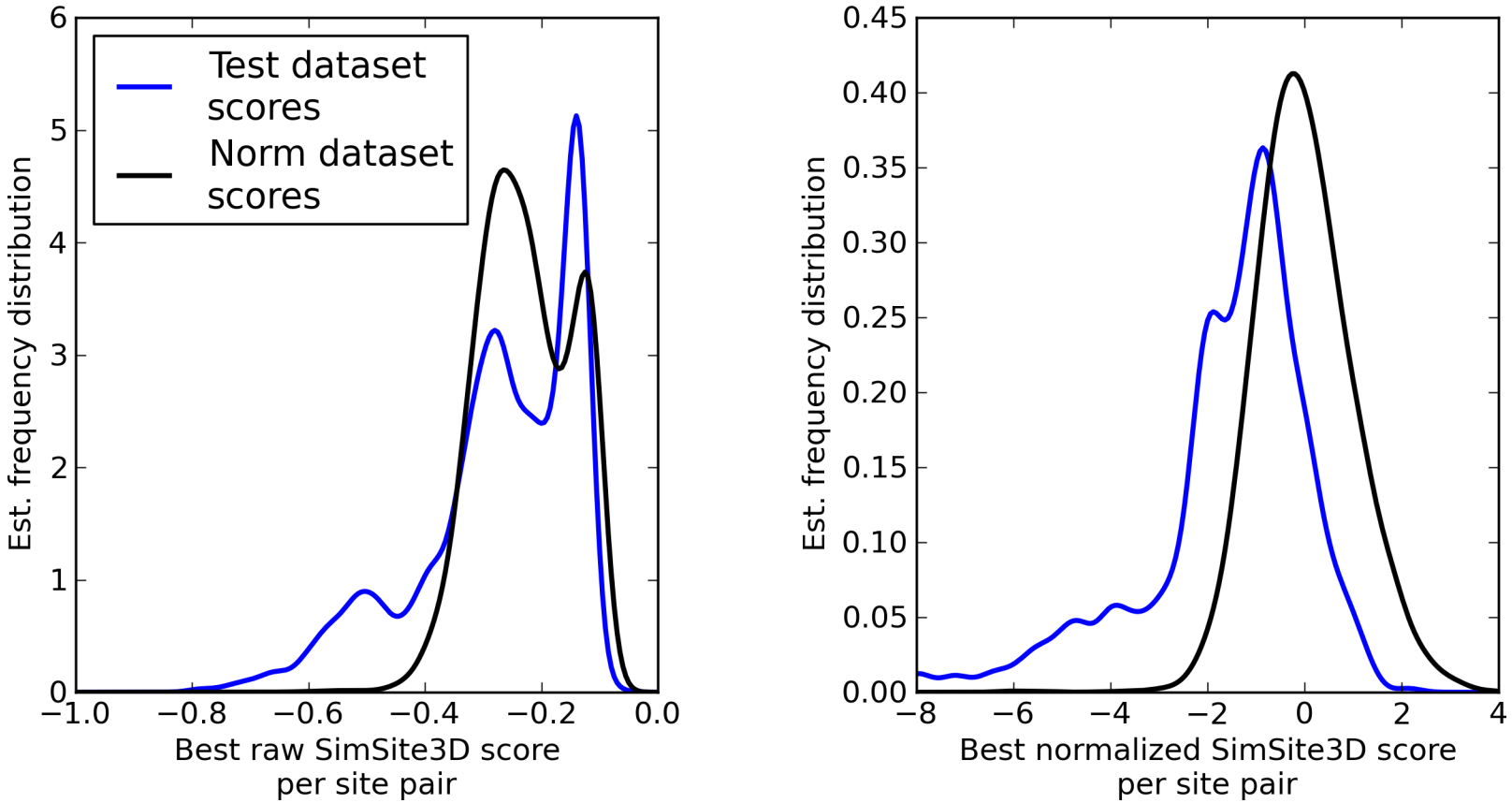


Figure 12: Class conditional density estimates showing the effects of score normalization on amount and shape of the overlap of the score distributions of the best scoring alignments per query, dataset site pair. The sets of scores are classified with respect to the test folds and normalization dataset. This plot highlights the level of difficulty of the problem and can be used to select a score threshold based on the percentage of true positives one wishes to recognize at the cost of a percentage of false positives. Given the samples used in the plot, an ideal scoring function would be one that minimized the amount of overlap between the test fold score distribution and the normalization dataset score distribution.

3.3 A Comparison of Existing Approaches to Aligning Binding Sites

To gauge the contributions of our methods, the performance of our implementation (SimSite3D 3.3) is compared to that of two other site comparison methods. MED-SuMo [53] was chosen because it is computationally efficient as it uses a relatively small number of points to represent a binding site. SiteEngine [91] was selected because the Principal Investigators are well respected, they rigorously evaluate their computational methods, and they have been addressing the binding site comparison method for many years. An additional deciding factor was the free availability of the two tools for academic laboratories. The pterin binding site dataset (Table 6) was used as the test dataset to compare the three methods as there are four distinct protein folds represented and three of the four folds have at least four distinct sequences.

The all-to-all comparisons between the pterin binding sites dataset for both MED-SuMo and SiteEngine were performed in approximately the same manner for both tools and similar to the method used for SimSite3D. In order to have the query sites of approximately the same size and location, the biopterin from 1DR1 was placed in the reference frame of each query protein structure using the reference ligand/structure based alignments. The MED-SuMo dataset binding sites were defined by the ligand bound in the pterin pocket of each crystal structure. Because SiteEngine searches the entire protein surface of each dataset protein, the dataset binding sites were not defined. As recommended by the tools' designers, the threshold for considering a chemical point as part of a binding site was at most 4.0 and 4.5 Å from any ligand heavy atom for SiteEngine and MED-SuMo, respectively.

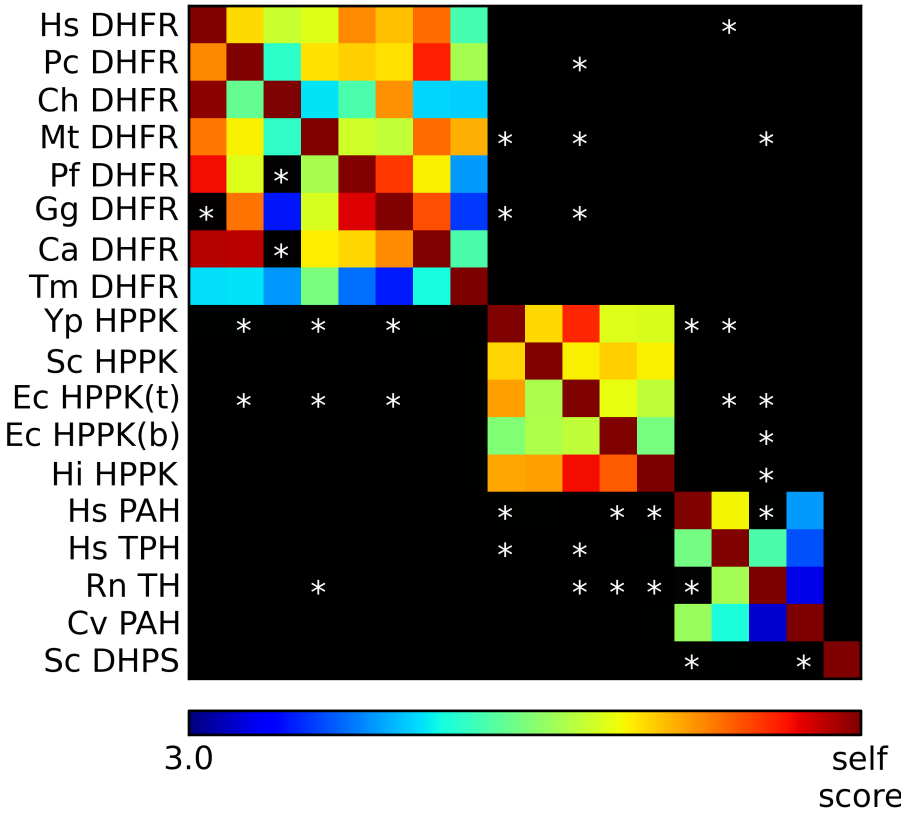


Figure 13: MED-SuMo score matrix for the pterin binding proteins dataset. The scores of the hits for each row (one query site) are scaled linearly to be in the range [3.0, self_score] where self_score is the best possible score for the corresponding query pocket. The range [3.0, self_score] is mapped linearly to the color bar. A black cell with an asterisk indicates that MED-SuMo was unable to find a significant alignment between the two corresponding sites (only 3 points matched). A completely black cell indicates that MED-SuMo did not find any matches between the two sites.

MED-SuMo performs well, but its scoring could be improved since it is basically a count of the number of points that were matched. If one ignores the recommended score threshold, MED-SuMo can hop between the pterin folds. However, one must remember that only 3 points matched for any of those ”hits”.

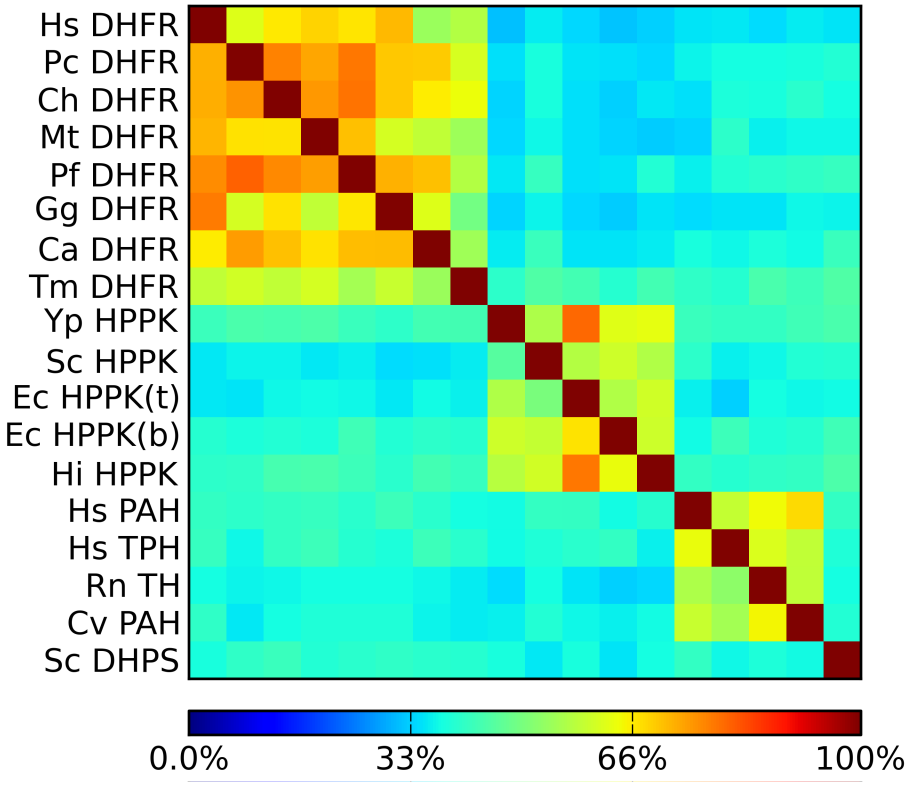


Figure 14: SiteEngine score matrix for the pterin binding proteins dataset. As recommended by the authors [91], the SiteEngine scores for each query were converted to a percentage of self-score by dividing each score by the query's self score. Notice that SiteEngine scores for pairs of sites within a protein fold are typically greater than 50 percent, and for those pairs outside of a fold the scores are about 33 percent.

Because of the more detailed nature of SiteEngine's site models, SiteEngine's scores show a range more like those of SimSite3D than MED-SuMo.

3.4 Discussion

Looking at the score matrices for the pterin binding proteins dataset, we see that SimSite3D, MED-SuMo, and SiteEngine all perform very well within each protein fold. Good performance within a given protein fold is expected because the binding sites will, in general, be formed by many of the same residues with similar relative poses. On the other hand, for protein within the same fold, tools such as DALI and SSM are generally

sufficient to correctly align the binding sites. Therefore, a useful binding site alignment tool must necessarily perform well for binding sites within the same protein fold, but that is not sufficient to motivate the use of binding site comparison tools as structure based methods can usually provide low error alignments. Of course, a primary advantage of binding site comparison tools is their emphasis on binding site features rather than more global structural features.

An advantage of SimSite3D is the score normalization is provided automatically, and we have provided a score threshold for a site alignment to be considered significant. A major issue with SiteEngine is one does not know which hits are significant and for sites outside the protein folds it does not seem like SiteEngine picks any ”winners” or ”losers”. In our view, MED-SuMo uses too few points to represent binding sites in order to use MED-SuMo to find similar pockets (i.e. binding sites of ligand fragments about the size of adenine). The score normalization and the spread of the scores of SimSite3D clearly designates some site alignments to be ”winners” and ”losers”.

Given the difficulty of the binding site alignment and comparison problem, our method and implementation has many areas that could be improved. Because of the heavy reliance on hydrogen bond points, hydrophobic sites are more challenging to align and have fewer high-value points to indicate the alignment is correct. Looking at high scoring alignments between some polar query sites and the normalization dataset, there are a number of cases where the polar points do match well, but the shape of the binding sites are very different. Unfortunately, the point clouds seem to not provide an adequate representation of the binding site shape in all cases. Therefore, it is likely that adding information about the complementarity of the shapes of aligned binding sites would help to better distinguish between true hits and false positives.

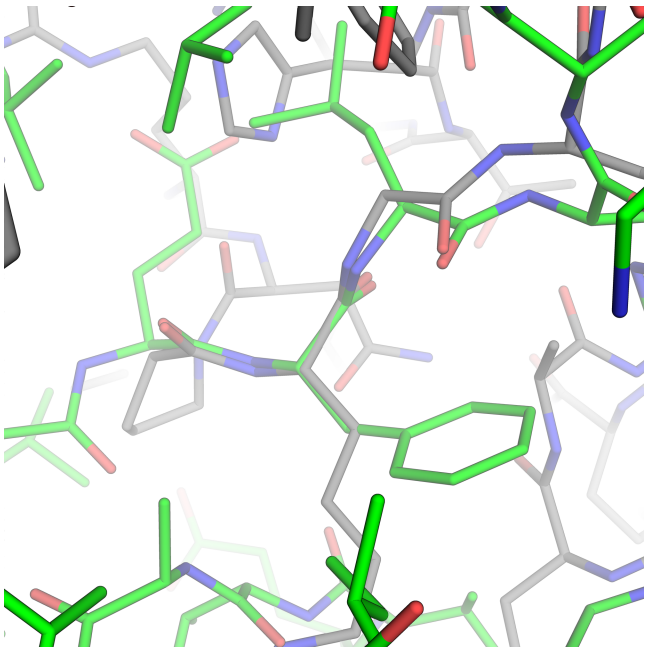


Figure 15: An example of a strong mainchain motif match, but poor binding site shape. The tubes with green carbon atoms are from a H. sapiens protein kinase CDK2 (PDB: 1B38). The tubes with gray carbon atoms are from a H. sapiens peptide binding protein (TRAF6). Notice the backbones (tubes) in the center of the figure match (typically called a similar protein backbone motif). The problem is the green set of matching tubes correspond to the canonical binding motif kinases use to recognize N1 and N6 of adenine, but the adenine binding site is too small for a peptide to bind.

Besides model and implementation details, there are several computational challenges that must be addressed before the accuracy of high-throughput computational chemistry tools can be increased with the goal of greatly reducing their number of false positive solutions. A major issue for both binding site comparison and protein-ligand docking tools is correct modeling of water mediated interactions. The modeling of water has too many details to present here, but the two extremes (including no water or all water molecules in the binding site) do not work well in practice. At the present, too many resources are required to specify which water molecules to include for each dataset site. Including all the water molecules that are near the binding site and are present in the crystal structure is likely to restrict the binding site to present a shape and chemical signature that can only be matched to a site with the same ligand or one of its analogues bound in a very similar conformation. The reason is: including all such water molecules in a GOLD redocking

study greatly increased the accuracy of the method and biased it to the crystallographic pose and conformation [41]. Since the inclusion of all water molecules seems to be about as ineffective as including no water molecules and such inclusion takes more computational resources, most (if not all) high-throughput methods ignore water molecules by default.

Chapter 4

Binding Site Surface Complementarity

Given the results in the previous chapter, our binding site comparison approach and implementation shows great potential for posing candidate ligands for proteins of unknown function and for pocket mining. However, as is commonly the case with high-throughput computational chemistry tools, the search results are plagued by a significant number of false positive hits. In particular, for any of the test site similarity searches, a number of the hits near the score threshold are from sites that have very different molecules bound than those the query protein is known to bind. Besides reducing the number of false positives, we seek to reduce the alignment error of the better alignments and reduce the number of poor alignments within the test folds. Our hypothesis is: if two binding sites have a similar shape, the preferential binding of ligands to one of the sites over the other will be based on chemical differences alone. In this chapter, we present the impact of including the molecular surfaces of the binding sites to represent their shapes.

In the previous chapter, the degree of similarity of the binding site shapes was not adequately addressed since the points in the chemistry labeled point clouds are sparse and unevenly distributed. Binding site shape is known to be important because for a protein and ligand to interact their surfaces should complement each other [28] in a manner somewhat akin to a soft lock and key rather than a mortise and tenon woodworking

joint [55]. An example of two aligned sites with a high degree of local chemical similarity but very low surface complementarity is the alignment between the adenine binding site in H. sapiens CDK2 (PDB: 1B38) and the antigen binding site in H. sapiens TRAF6 (PDB: 1LB6). Both proteins share a similarly exposed and oriented backbone segment (Panel A of Figure 16). Therefore, locally, one would expect the molecules that interact with the two proteins to place polar atoms in approximately the same relative position and orientation. However, the shape of the two binding sites is very different (Panel B of Figure 16). Given the very different pocket shapes, our best judgment is that the ligands bound by the two proteins will have very different shapes. Thus, in many instances, the chemistry labeled point cloud representation and partial matching of atomic positions is insufficient to characterize the degree of shape complementarity of two sites.

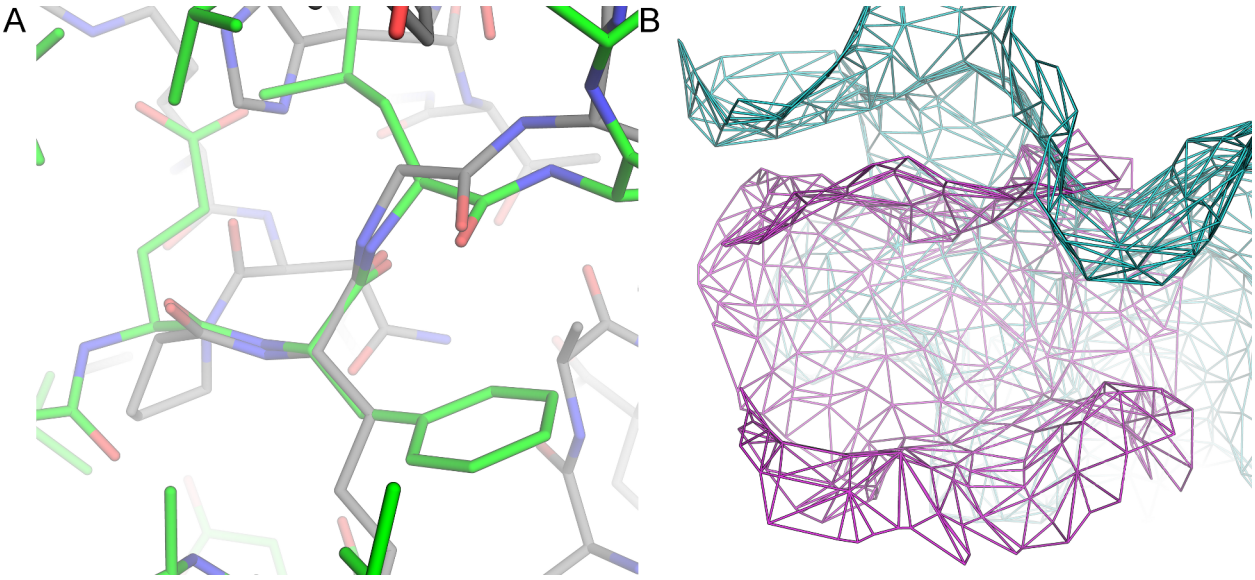


Figure 16: Example of a strong partial polar match between binding sites with very different shapes. Panel A illustrates the adenine binding site of H. sapiens CDK2 (green carbon tubes) as matched to the antigen binding site in H. sapiens TRAF6 (gray carbon tubes). Notice the very similar protein backbone pattern in the center of panel A. Panel B shows the molecular surface patches for the two binding sites from approximately the same viewpoint as panel A. In panel B the cyan surface is from TRAF6 and the magenta is from CDK2. The surfaces are quite distinct and only agree near the similar backbone pattern in the center of the adenine pocket.

Likewise, the fact that two binding sites have similar shapes, is not sufficient to fully

assess the similarity of two sites. The reason is there can be substantial chemical differences between the two binding sites. An example of similar site shape and different chemistry is a binding site shape alignment between the adenine binding pocket of a H. sapiens τ kinase I structure (PDB: 1J1B) and the indole binding pocket of a P. putida naphthaline 1,2-dioxygenase structure (PDB: 1O7N). In Figure 17, one can see that the polar site points have few correspondences (between red and pink and between blue and light blue), but the surfaces are quite similar over most of the two pockets. Therefore, in this chapter, we emphasize the use of both the chemistry labeled points and the site surface patches with the goal of increasing the number of true positives and reducing the number of blatant false positive site matches.

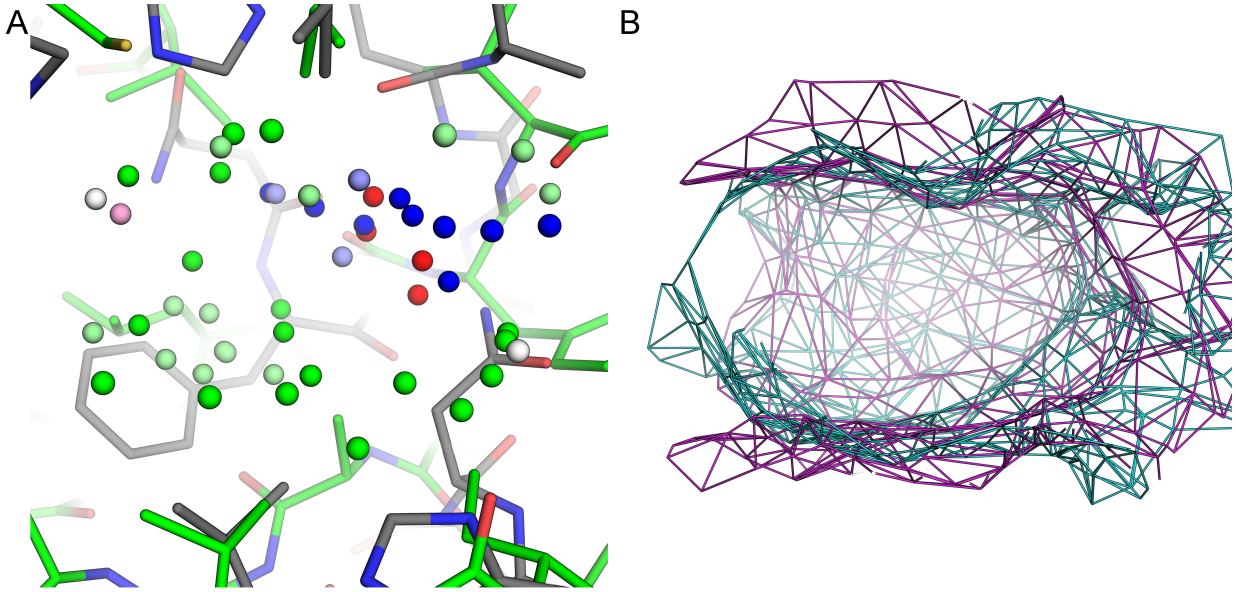


Figure 17: Example of a good partial surface match between binding sites with few polar points in common. Panel A shows the adenine site of a H. sapiens τ kinase I (green carbon tubes) as aligned to the indole site of a P. putida naphthaline 1,2-dioxygenase (gray carbon tubes). The site points are shown as spheres, with those from the naphthaline 1,2-dioxygenase in lighter shades than those from the kinase. In panel B one can see that the majority of the 2 mesh surfaces is complementary.

4.1 What is a binding site surface patch?

When analyzing how proteins interact with other molecules (e.g. proteins, water, and small molecules), one would like to characterize the boundary that separates the protein atoms from the atoms of other molecules. A common representation of biomolecules (including proteins) is modeling the atoms by a hard ball centered at each atom's center with each ball's radius specified that atom's chemical element. One example of a molecular surface is the van der Waals surface which is the set of the exposed surface points of all the balls.

For our purposes we list a few technical definitions from general topology that are reasonable, at least, for $\mathbf{R}^3$ [83].

- The **complement** of a set S contains all of the points that are *not* in S, and it is denoted as S^c. That is, $S^c = \{p | p \notin S\}$.
- A **ball** is another name for the volume of a sphere, and may be written as $b(c, r) = \{x | d(x, c) \leq r\}$.
- A **neighborhood** is another name for the interior of a sphere. A neighborhood as a set is $N_r(c) = \{x | d(x, c) < r\}$.
- An **interior point** of a set S is a point $p \in S$ that has a neighborhood $N_i(p)$ that is fully contained in the set of interest (i.e. $N_i \subset S$).
- A **limit point** of a set S is a point $p \in S$ such that for *each* neighborhood N_i of p, N_i contains a point $s_i \in S$ where $s_i \neq p$.
- Let L be the set of limit points and I be the set of interior points of a set S in $\mathbf{R}^3$. Then a **surface point** of S is an element of the set $L \cap I^c$.

Given these definitions, the van der Waals surface is the set of limit points that are not interior points of the union of balls that represent a molecule's atoms. Given two atoms a_i, a_j,

with centers c_i, c_j, their van der Waals radii r_i, r_j can be determined by and considered as the minimum of the distance between their centers when they are not participating in the same chemical bond (i.e. $r_i + r_j = min(\|c_i - c_j\|)$). Although the van der Waals surface of a protein is a reasonable approximation, it is defined as the intersection of spheres and has many sharp valleys which are not aesthetically appealing in molecular graphics. The valleys are not necessarily important shape features since atomic centers from other molecules can not be in the valleys as such molecules would then penetrate the protein.

The idea of generating a smoothed surface by rolling a probe sphere of constant radius over the van der Waals spheres was presented by Lee and Richards [64]. There are two general classes of smoothed molecular surfaces. The distinction is one surface is traced by the center of the probe and the other surface is defined by the extent of the molecule's van der Waals surface and the probe's surface.

> **Definition.** A **solvent accessible surface** (SAS) of a molecule M is the limit surface at which water molecule *centers* can be placed such that the water molecules do not penetrate M [64].
>
> **Definition.** A **solvent excluded surface** (SES) or smoothed van der Waals surface is the limit surface at which the *boundary* of a water molecule can be placed such that the water does not penetrate the protein [40].

Because of the offset with respect to the protein's volume, the SAS of a protein exhibits different local features than a smoothed van der Waals surface as it is approximately 1.4 Å farther out from the atomic centers. As an example, the grooves of $\sim$ 1.4 Å width in a van der Waals surface will be represented as creases in the corresponding SAS surface. At the present, many protein scientists prefer to consider smoothed van der Waals protein surfaces because they seem to be the more natural surface since they approximate the limit of proteins' volumes and shapes. Also, it has been argued that a smoothed van der Waals surface is more applicable to describing hydration effects [51, 98]. Finally, given two non-covalently bound molecules, if they are represented by their respective smoothed

van der Waals surfaces, the two surfaces will be complementary at the interface [28], but their respective Solvent Accessible Surfaces will have significant intersections and are not necessarily visually complementary.

Given the topological details of constructing molecular surfaces, we selected Michel Sanner's MSMS [87] to construct triangular meshes that represent the smoothed van der Waals molecular surfaces of proteins [1]. The main advantages of MSMS are its speed of surface construction, it computes a smoothed van der Waals surface, and the MSMS program is freely available for academic use. Our implementation is not restricted to surfaces generated by MSMS as the only requirement is that a site's surface files be in MSMS format.

> **Definition.** A **binding site surface patch** is constructed by pruning a given protein molecular surface mesh to keep only those faces near the site volume.

In our implementation, if a ligand was used to define the site volume, all faces which do not have at least one vertex within 4.0 Å of a heavy ligand atom are removed. If a sphere was used to define the site volume, all faces which do not have at least one vertex inside the sphere or within 1.0 Å of the sphere are removed. In this manner, only those molecular surface faces near the binding site are kept, and this set of faces is called a binding site surface patch.

4.1.1 Computing surface patch complementarity

How to practically compute the surface complementarity of two arbitrary 3D objects is both a research and an engineering problem. Two aligned surface patches may be compared as a set of corresponding points in a manner similar to the methods proposed by Besl and McKay [13]. In this manner, the first surface is represented by a set of sample points, that are given by the vertices of the surface's mesh. Since the two surface patches

1 From this point forward, when a molecular surface is referred to it is to be assumed that it is a triangular mesh representation of a smoothed van der Waals surface.

are assumed to be coarsely aligned, the point correspondences are determined by computing the closest point on the second surface for each sample point from the first surface. Because the analytical surface description of a molecular surface of a binding site is difficult to work with, the point correspondences are estimated by computing the closest points with respect to the second surface's triangle mesh. Such an estimation is reasonable since, in the limit, the sample points and mesh surfaces converge to the analytical surfaces However, the problem becomes computationally intractable as the number of points and faces approach infinity. Therefore, a balance is required between desired accuracy and computational efficiency.

Given a mesh surface, finding the closest point on the mesh with respect to a sample point can be a costly process. A naive method is, given a sample point, compute the corresponding closest point for each face in the mesh and keep the point with the minimum distance. A slightly better method is to have an upper bound at which we desire a point correspondence and to use an overlapping grid to partition the volume of space containing the mesh. In practice, our grid implementation assumes an upper bound of 1.5 Å for point correspondences, and produces exactly the same results as the naive method while checking about one percent of the total faces of an average dataset binding site.

The degree of surface complementarity of two surfaces is estimated by the RMSD between the query mesh vertices and their corresponding points on the dataset mesh. Because there is an upper bound on the distance for allowed point correspondences, the RMSD is perturbed by adding or removing points (i.e. having more point correspondences may increase the average point correspondence error, but could indicate a better partial match as there would be more points with correspondences within 1.5 Å). To address this discrepancy, each point without a correspondence is considered as having an error of 1.5 Å . The RMSD of the corresponding surface points is added as another term in the scoring function training process.

4.1.2 Updated Training/Validation Datasets

After gaining experience with comparing protein-ligand binding sites, it was noted that the initial training datasets suffered from a number of blatant flaws. Several of the datasets have only two proteins, the structures in the structural genomics structures dataset do not have similar binding sites, and the peptidyl-prolyl cis-trans isomerase dataset has three NMR structures. Having only two proteins is somewhat problematic since there are only four pairs of binding sites, and such datasets will be underrepresented in the training samples. Our method may be used to search using an NMR query structure or a dataset that contains NMR structures; however, protein structures determined by NMR typically suffer from higher relative atomic positioning errors than structures determined by xray crystallography. In general, our experience has been that the training datasets should be carefully prepared to reduce the probability of two binding sites being labeled as similar when they are in fact dissimilar with respect to the site representation.

To address these issues, several datasets were removed/added and the remaining training datasets were augmented to approximately double the number of binding sites used to training the scoring functions. If possible, the sites were aligned using both structure and ligand based alignments. For each family that could be aligned using both methods, the alignment method with the better average main chain RMSDs for the binding site residues was selected as the alignment choice for that dataset. The aligned structures were scrutinized by protein structure experts using molecular graphics and structural features to determine whether to partition the training datasets by protein families. Several of the training datasets had distinct protein folds for which the binding sites for the same ligand were so different that these datasets were split into subfamilies for the purpose of training the scoring functions.

To gauge the impact of improving the curation of the datasets and doubling the number of total structures in training datasets, one can compare the validation results for the SimSite3D site point score over both training and validation datasets.

Table 9: Mean, median, and standard deviation of the site map RMSD of the best scoring alignment per pair of validation set binding sites across 10 orientation samples. The ”old” values are those previously reported in Table 3; the ”new” values are the result of updating the training and validation datasets.

SF	mean	median	stdev	SF	mean	median	stdev
”old” site score	2.98	1.81	2.94	”new” site score	2.27	1.22	2.11

It is clear that the training/validation dataset enhancements are beneficial. The new scoring function (using the same terms as the ”old” scoring function) has a much better median RMSD with respect to the enhanced datasets (Table 9). Also, the average and standard deviation of the RMSD have dropped significantly.

4.1.3 Scoring Function Training and Validation

The scoring function training and validation was performed in a manner that is very similar to that of the previous chapter. The stratified method of sampling the orientation space is the same as in Chapter 3. For each sample population, each scoring function was trained ten times. With a distinct training dataset reserved for validation each of the ten times. Ten sample populations were used to help reduce the effects of sampling. The final scoring functions are the stacked scoring functions found by averaging the 100 values for each weight. The parametric form of the scoring functions with respect to RMSD of site alignment is again -1/RMSD [97]. The site point features (terms 1-5) are computed in the manner presented in Algorithm 4. The term numbers are the same as those presented in Chapter 3 with the exception of term 12, which is the RMSD of the corresponding surface points (i.e. average surface error – see Section 4.1.1). The combinations of terms in the scoring functions differ and can be found in Table 10. The solutions (weight vectors) of the linear regression problems were found using Python and NumPy to implement the standard QR-factorization method presented in [42].

Table 10: Combinations of terms (features) used in linear regression to construct linear scoring functions to predict site alignment quality and site similarity.

SF #	terms	SF #	terms	SF #	terms
0	1,5	1	4	2	2,3
4	12	7	1,2,3,5	8	2,3,4
9	1,2,3,4,5	13	2,3,12	14	1,2,3,5,12

In the previous chapter, we assumed that it was reasonable to use the mean, median, and standard deviation of the RMSDs of the best scoring alignments to evaluate the candidate scoring functions' performance. Rather than using such global parameters, ROC-like curves are used, in this section, as guides to choose the "best" scoring function. The main advantage of ROC-like curves is they show the interplay between true and false positives as the score threshold is varied from too strict (no binding sites pairs are similar) to too loose (all binding site pairs are similar). As in the previous chapter, the scoring function candidates are evaluated based on their performance on the validation datasets.

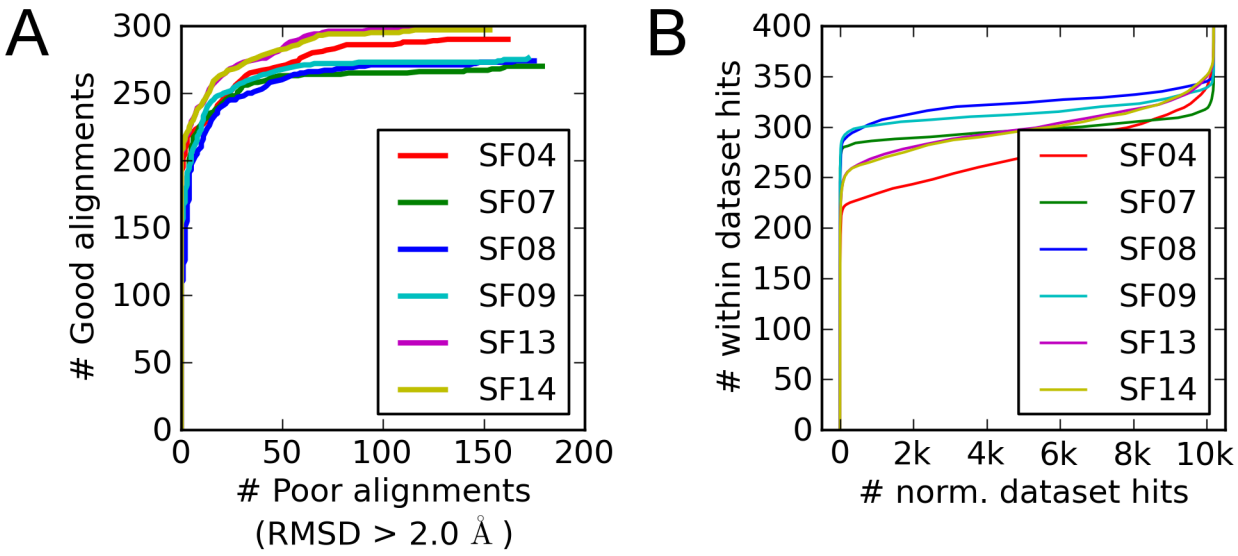


Figure 18: ROC-like curves comparing the performance of six of the scoring functions on the validation datasets. The plotted data is the score and RMSD of the best scoring orientation per query, dataset pair of binding sites averaged over the 10 stratified alignment samples. Graph A shows the alignment selection performance of the scoring functions. Graph B shows the ability of the scoring functions to discriminate between sites within the protein folds and those in the normalization database. As one moves along a curve from the bottom left corner to the upper right, the score threshold becomes more lenient.

Several observations can be made based on the validation results. First, the addition of the surface term gives a significant increase in the number of better quality alignments for the validation datasets. Second, the scoring functions with the surface term seem to at a disadvantage with respect to discriminating between within validation family alignments and high scoring normalization dataset hits. However, if the information from both plots is considered, one can note that about 260 of the SF8 within family hits are well aligned and about 300 within family hits score better than those from the normalization dataset. Therefore, one cannot definitively conclude that SF8 is better than, say SF13, at discriminating between true and false positives as about 40 of the higher scoring within family alignments (as scored by SF8) are based on pairs of sites with significant alignment error. Rather, SF13 might be preferred because after about 250 good alignments, it is difficult (based on score) to distinguish between good and poor alignments, based on their alignment error, and that is about the same number of alignments after which it is

difficult to discriminate between the within family hits and normalization dataset hits. Thus, SF13 is preferred over SF8 because the scoring is more consistent with the error of alignment.

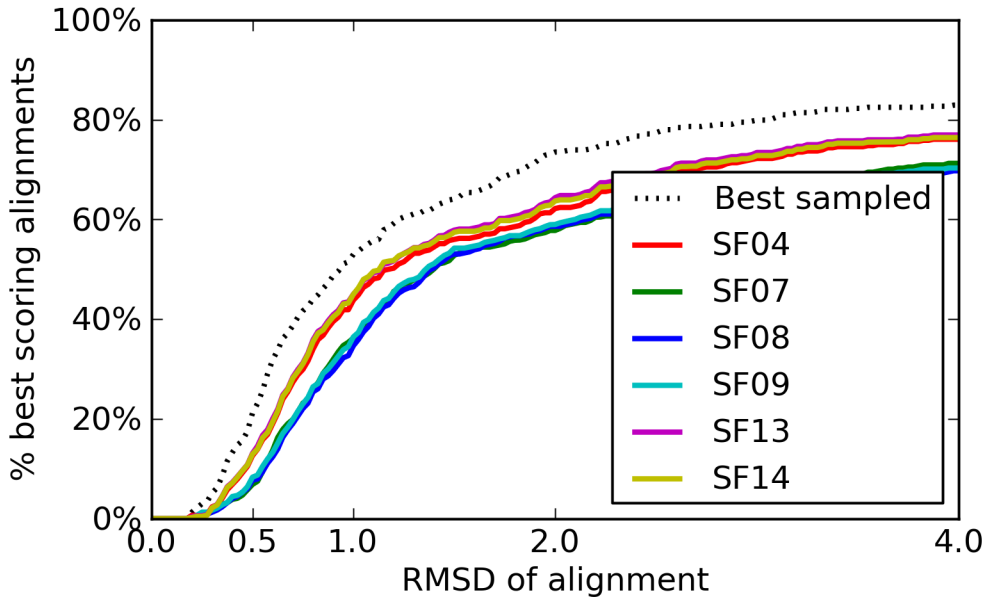


Figure 19: Cumulative distributions showing the percentage of best scoring alignments (one alignment per query, dataset pair of the validation binding sites) with error less than or equal to a given RMSD threshold. Each pair of binding sites is from one of the ten validation datasets. Notice that the scoring functions that use the surface information consistently "catch" more orientations at a any chosen RMSD threshold above 0.3 Å RMSD. The best sampled denotes the upper bound for any scoring function, since that is the upper bound on the alignment error present in the validation alignments.

When considering the percentage of alignments with less than or equal to a given error threshold (in RMSD), the scoring functions that use the surface error term show a strong gain in alignments in the range of $[0.3, 1.25]$ Å and the others in the range $[0.5, 1.5]$ Å RMSD. Although alignments are gained if an error of > 1.5 Å RMSD is allowed, the rate of increase is much lower than for thresholds below 1.5 Å RMSD. It is not unreasonable to expect that the RMSD of the best sampled orientation for each pair of validation sites be < 1.0 Å RMSD. However, several of the validation sets have more than one protein

folds with distinct modes of binding the same ligand and it can be difficult to consistently align and recognize low error alignments for such pairs of binding sites.

Because of the emphasis placed on surface complementarity, we will compare the performance of scoring functions that use the surface error term with those that do not. Scoring functions 8 and 13 were selected to be used in the scoring function testing step because they both perform well for their respective category and have fewer terms than scoring functions in the same categories that had similar performance (Occam's razor). Let us denote the scoring function 8 as SF8 and the scoring function 13 as SF13.

4.1.4 Scoring Function Unbiased Testing

Here the generalization ability is assessed of a scoring function that using the site point complementarity only (SF8) and of a scoring function that uses both the hydrogen bond component of the site point complementarity and the surface complementarity (SF13). In particular, we seek the effects of adding surface complementarity in the cases of otherwise unrelated proteins that bind the same small molecule. The results are presented for the three more challenging test datasets from the previous chapter: adenine binding proteins, pterin binding proteins, and GST hydrophobic sites.

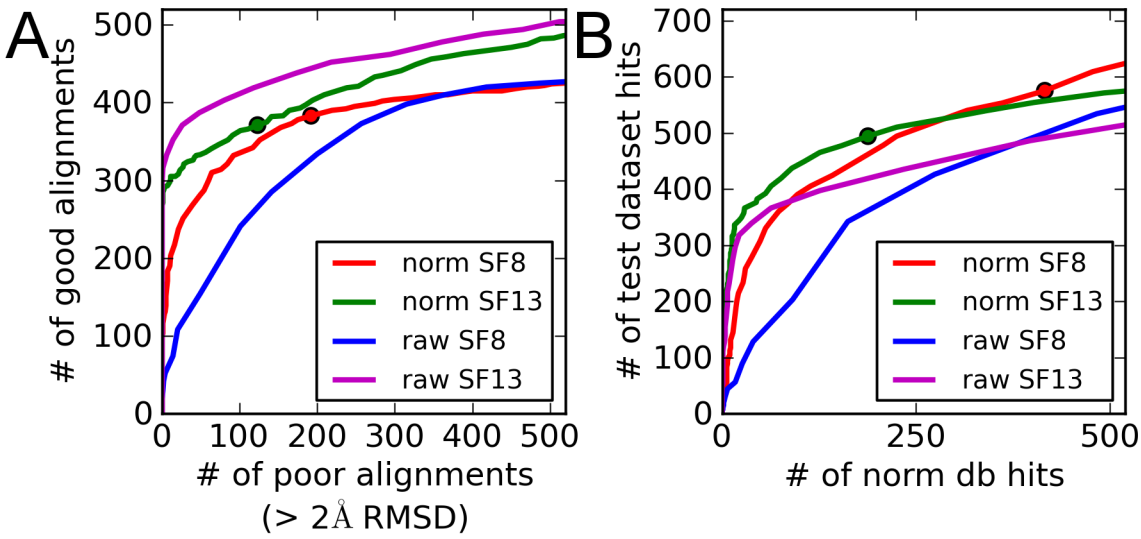


Figure 20: ROC-like curves showing SimSite3D performance when using site points (SF8) and site points and surface (SF13) to assess the similarity of aligned sites. Panel A shows the ability of the scoring functions to predict if the alignment error is significant for the best scoring alignments from the test datasets. Panel B shows the scoring functions' performance with respect to discriminating between the best test family alignments and the best alignments of query sites to those in the normalization dataset. The norm curves are the results when the raw scores are normalized using the mean and standard deviation of the query site's scores for the 140 sites in the normalization dataset. The dots on the norm curves denote the point where the score is 1.5 standard deviations better than the mean score with respect to the normalization dataset. Please note that the data in the two panels differs as is noted by the axes' labels.

The results of the two site scores illustrate that score normalization is not necessarily helpful (Panel A in Figure 20). Note that the alignments for any two pairs of sites are the same for both the raw and normalized scores from the same scoring function, but the relative ranking of hits between two or more query sites may change upon normalizing the scores. Score normalization provides a significant improvement when using SF8 because the number of points and point types varies between query sites. Score normalization is detrimental for the surface scores, when predicting alignment quality (Figure 20). Thus, score normalization is generally helpful if the scoring function terms are not scaled, but can add noise if the terms have the same possible range for all training and testing samples.

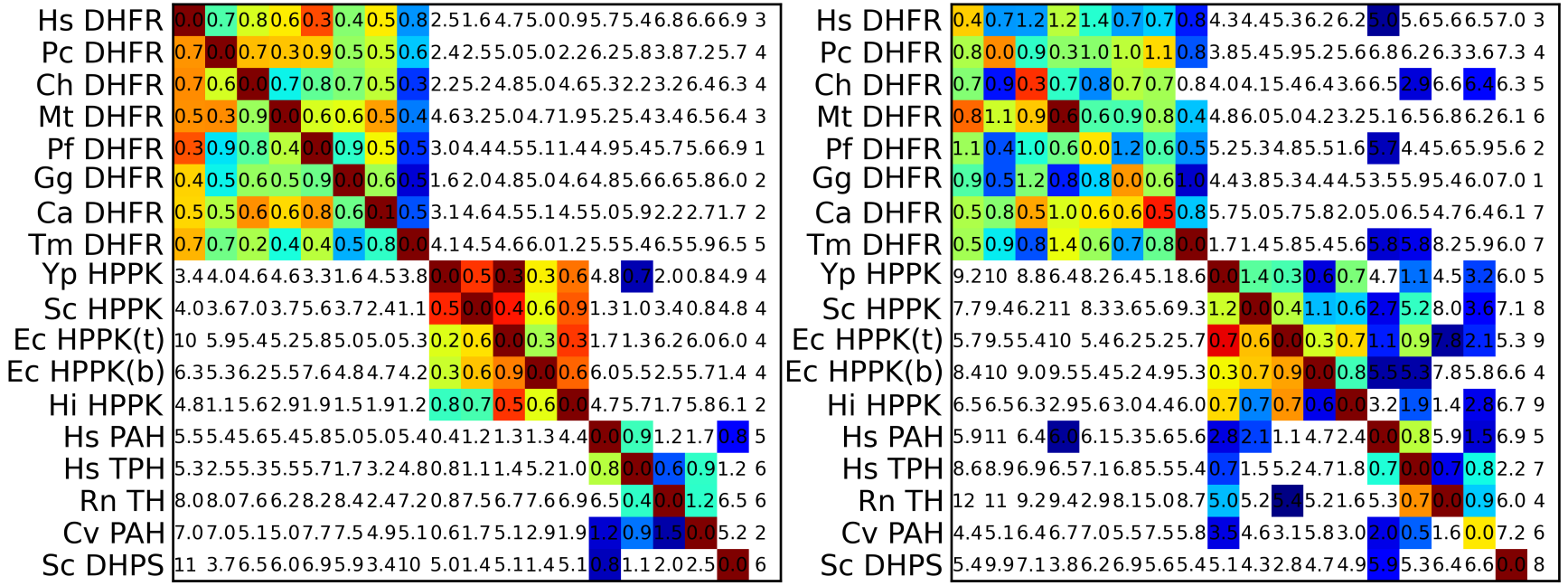


Figure 21: SimSite3D score matrices for the pterin binding protein families dataset. Each cell represents the best scoring alignment for that query, dataset pair of binding sites. The cells are colored with respect to the normalized score for that pair of sites. If a cell is white, the score is worse than 1.5 standard deviations better than the mean score for that query site with respect to the normalization dataset. If a cell is dark red, the score was at least 5.5 standard deviations better than the mean. The number in each cell (except for the last column) is the RMSD (error estimate) of the corresponding site alignment. The last column shows the number of normalization dataset hits (out of 140) which had a significant score. The left and right matrices show the best alignments with respect to SF8 and SF13 respectively.

We are interested in the effects of adding surface complementarity on the test datasets. One can see that the number of between family hits is much reduced when using SF13 versus SF8 on the pterin binding proteins (Table 21). However, when looking closely at the two matrices, a large number of the interfamily hits for SF8 (panel A of Table 21) have poor alignments (RMSD of alignment is much greater than 2.0 Å). The RMSD of alignment is consistently good for the hits recognized by SF13. Also, when looking at some of the alignments which SF13 did not recognize as significant hits, one can see that a large number of the sites are aligned within 2.0 Å RMSD (e.g. the Hi HPPK row and the cross family blocks between the aromatic amino acid hydroxylases and HPPK structures). These results indicate that for polar sites, SF13 out performs SF8 in choosing good quality alignments. However, given the current method to determining score significance, SF13 is unable, in most instances, to recognize when two similar sites from different folds are well aligned.

Because the GST hydrophobic sites have few polar points, SF8 has difficulty in predicting the quality of the sites and their degree of similarity. One can see that adding the surface complementarity to the site score (SF13) makes a clear distinction between the same binding site that has numerous inhibitors bound and the hydrophobic sites from other species and isoforms. The hydrophobic binding site of the mouse π-class GST is very similar to that of the human π-class GST, and this is clearly seen when using SF13 but not SF8 (Figure 22).

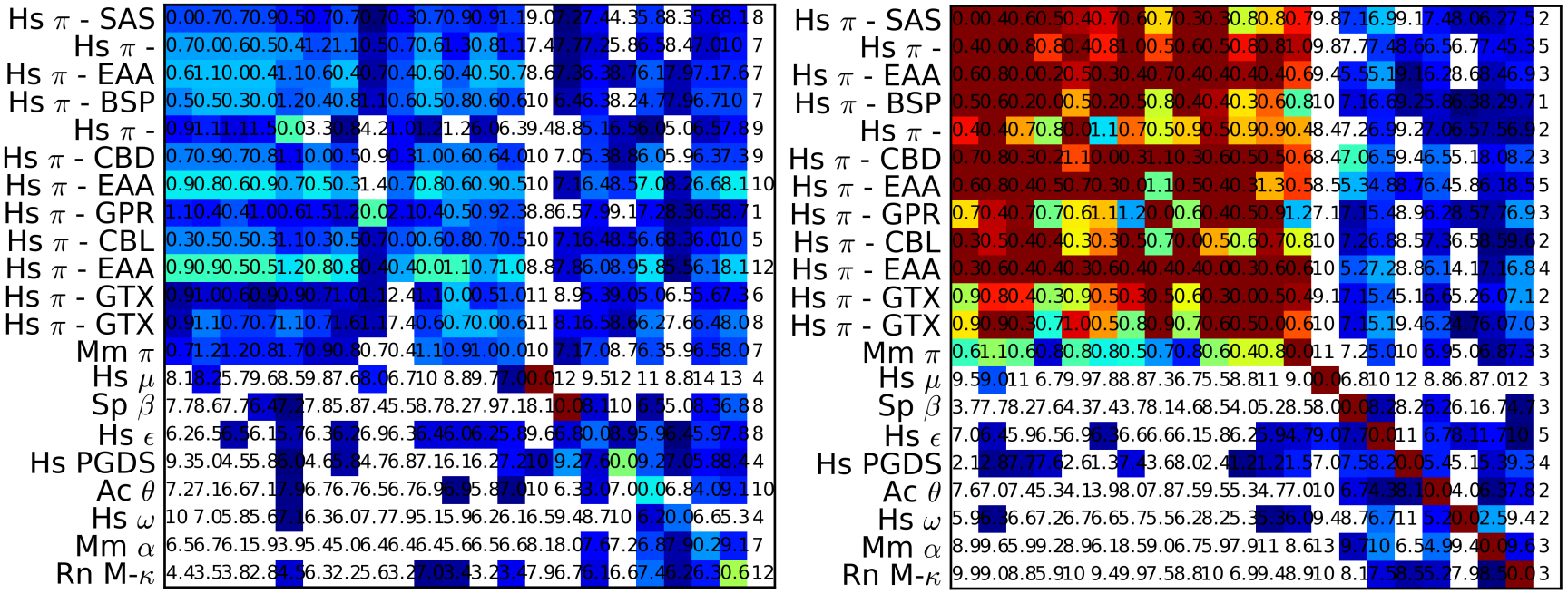


Figure 22: SimSite3D score matrices for the GST hydrophobic site dataset. Each cell represents the best scoring alignment for that query, dataset pair of binding sites. The cells are colored with respect to the normalized score for that pair of sites. If a cell is white, the score is worse than 1.5 standard deviations better than the mean score for that query site with respect to the normalization dataset. If a cell is dark red, the score was at least 5.5 standard deviations better than the mean. The number in each cell (except for the last column) is the RMSD (error estimate) of the corresponding site alignment. The last column shows the number of normalization dataset hits (out of 140) which had a significant score. The left and right matrices show the best alignments with respect to SF8 and SF13 respectively.

4.1.5 Discussion

Given the fact that SF13 tends to choose better site alignments than SF8, it would be advantageous to use SF13 to choose which orientations to consider. The main problem is SF13 is unable to recognize most well aligned interfamily hits as significantly similar, and SF8 considers a number of poorly aligned interfamily sites as similar. Close analysis of Panel B in Figure 20 shows that, at a score threshold of 1.5 standard deviations better than the mean, SF8 predicts about 80 more test dataset alignments as significant than does SF13. Unfortunately, looking at Panel A in Figure 20 one can clearly see that SF8 has about 75 more poor alignments than SF13 at the same score threshold. Given this dilemma, SF13 is taken as the better choice since it is more reliable at selecting lower error alignments for binding site pairs from distinct folds (for binding sites that are known to be similar).

4.2 Rigid Refinement of Aligned Binding Sites

Given the compromises in designing methods to search for candidate alignments, even the better candidate alignments for two 3D objects may have significant alignment errors. When such alignments are viewed in computer graphics, the human eye will easily detect the objects as being misaligned. A commonly applied method to refine global rigid alignments of 3D objects, in the context of partial matches, is iterative closest point (ICP) [13]. ICP is an iterative two-step optimization method that seeks to find the optimal rigid alignment and optimal point correspondences between two objects. ICP is typically implemented by keeping one object's pose fixed and adjusting the pose of the other site by looping over the following two steps:

1. The global orientation is held constant and is used to determine the best point correspondences.

2. The point correspondences are fixed and are used to update the global orientation.

Since the point correspondences and global orientation parameters can change after each iteration, the steps are repeated until one or more termination/convergence criteria are met. Although ICP need not converge to the global minimum, its relative simplicity and the fact that it works well in practice for coarse initial alignments has helped it to become widely used in object recognition applications such as refining the alignment of surfaces (e.g. matching range scans to CAD drawings).

An ICP method has been implemented in SimSite3D. Based on the features computed to score site alignment quality and site similarity, there are two sets of corresponding points: site map points and molecular surface points. Since the number of site point correspondences is small relative to the number of surface point correspondences, only the surface point correspondences are used to update the global alignment. The best rigid transformation is computed using the closed form method for unit quaternions as presented by BKP Horn [48]. The maximum number of iterations is set to a default of 100, and if after an iteration, the change in the RMSD of the corresponding points is greater than -1E-06 the method will terminate. Finally, because each iteration of ICP requires an update of the corresponding points, ICP is relatively computationally expensive and is only applied to the best scoring alignment for each pair of binding sites.

4.2.1 Results of Applying Iterative Closest Point

The most advantageous effect of applying ICP to the best scoring alignment per test dataset site pair is well illustrated by catchment curves (Figure 23). ICP improves the accuracy of most of the alignments (chosen by SF13) for which the initial RMSD of best scoring alignment is ≤ 1.25 Å . However, on average, ICP does not reduce the alignment error for those site pairs that have a larger initial alignment error.

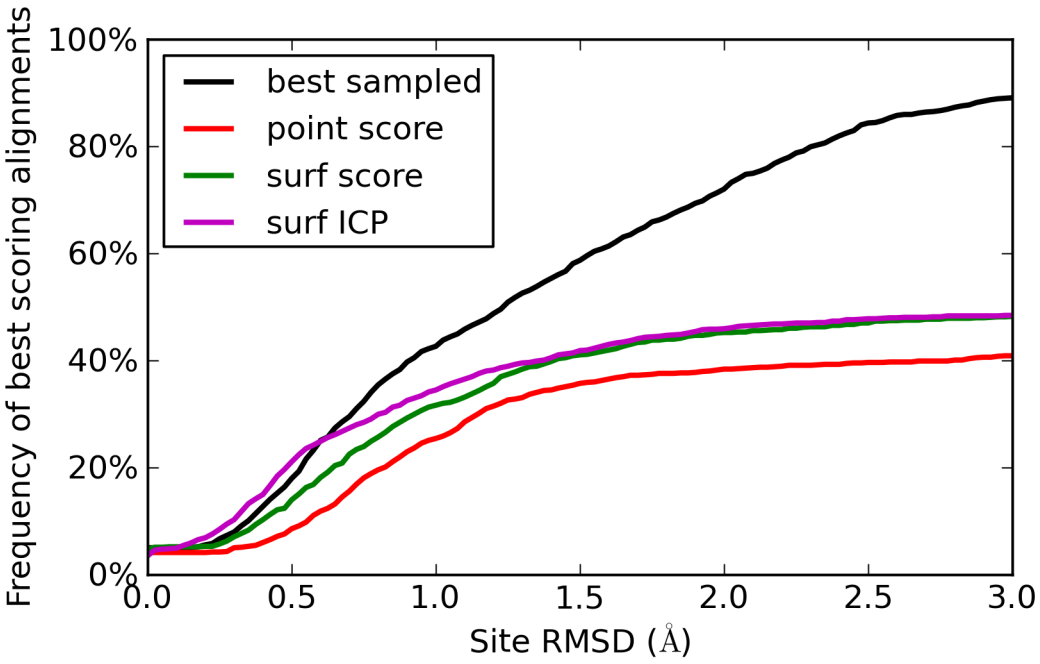


Figure 23: Catchment curves (cumulative distributions) showing the effect of ICP on the RMSD of the best scoring alignments (one alignment per query, dataset pair of test dataset binding sites) for the three test datasets. These curves show the percentage of best scoring alignments with error less than or equal to any given RMSD threshold in [0.0, 3.0] Å RMSD. The best sampled curve is the upper bound for any scoring function (before ICP), since it gives the percent of site pairs that have at least one candidate alignment with error less than or equal to a given RMSD threshold.

4.2.2 Comments

Based on the results for the three test datasets, ICP is seen as very useful in that it reduces the alignment error when the best scoring SF13 alignment is within 1.5 Å of the reference alignment. In particular, half of the alignments that had an error of 1.5 Å RMSD or less have their alignment error reduced to less than 0.5 Å RMSD after ICP (Figure 23).

Using SF13 to choose the best alignment per site pair and applying ICP to that alignment performs much better than SF8 at choosing alignments of good quality (Figure 23). ICP may be applied to the chemically labeled point clouds, but, on average, optimizing those correspondences did not improve site alignment or scoring. Given the improve-

ment in alignment quality when the starting alignment is close enough and that SF13 is preferred over SF8, the default mode of SimSite3D uses ICP to refine the best scoring alignment for each site pair.

Based on the results, the convergence funnel of ICP, within the SimSite3D search paradigm, is quite narrow with a "radius" of about 1.25 Å RMSD with respect to the dataset reference alignments. Given the coarse sampling of the surfaces, about one vertex per $Å^2$, and the local differences in pocket shapes and the global similarities of pockets of similar sizes, it appears that the energy landscape that is searched by the ICP implementation for two distinct sites is relatively noisy and has a number of local minima.

4.3 Two-tiered scoring

Computing the surface complementarity for a candidate alignment is relatively computationally expensive. For this reason, SF13 does not lend itself well as part of a high-throughput method on one processor core. As a heuristic, we assume that if two sites are sufficiently similar, ranking the candidate alignments of a pair of sites by their SF8 score will place at least one low error alignment within the top N alignments. The top N alignments for each site pair can then be scored with SF13 as SF13 is better than SF8 at predicting the quality of alignment. This scoring method is denoted as two-tiered scoring. It is our experience that using two-tiered scoring with $N = 10$ gives much better site alignments than using SF8 alone, and the computational cost is much less than determining the surface point correspondences for every candidate alignment.

4.3.1 Results

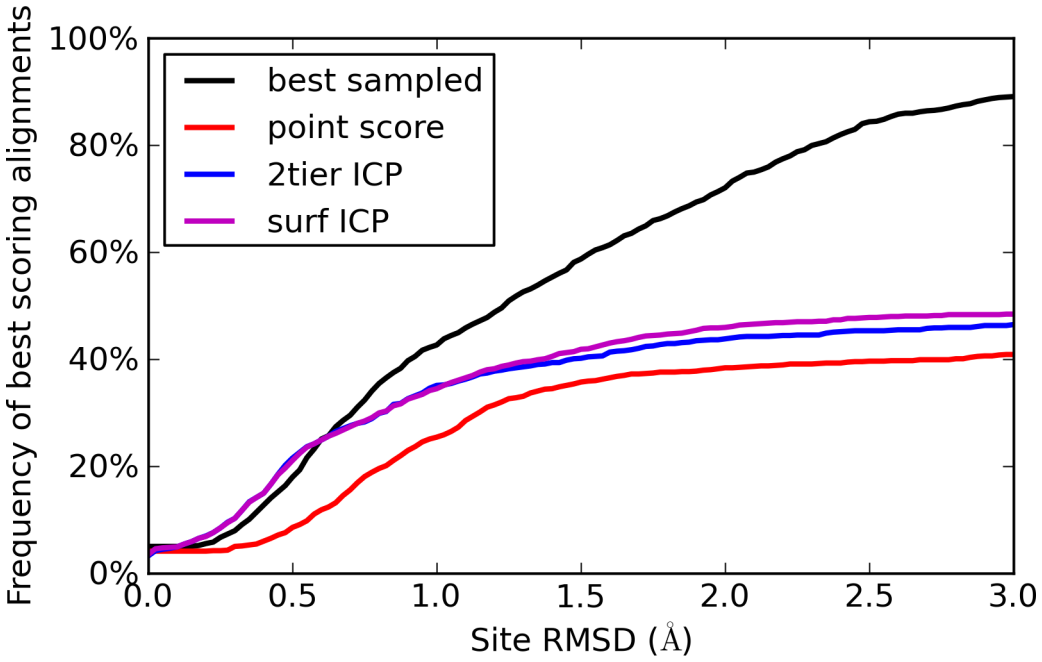


Figure 24: Catchment curves (cumulative distributions) showing the effect of two-tiered scoring and ICP on the RMSD of the best scoring alignments for the 3 test datasets. These curves show the percentage of best scoring alignments with error less than or equal to any given RMSD threshold in [0.0, 3.0] Å RMSD. The best sampled curve is the upper bound for any scoring function (before ICP), since it gives the percent of site pairs which have at least 1 candidate alignment less than or equal to a given RMSD threshold.

It is easy to see that on the three test datasets, two-tiered scoring & ICP on the best site alignment per site pair is virtually identical to SF13 & ICP for final best alignments within 1.25 Å RMSD. By 2.0 Å RMSD, SF13 does recognize the alignment of about five percent more pairs of sites to within 2.0 Å RMSD of the reference alignments than does two-tiered scoring. Also, it is clear that using two-tiered scoring & ICP provides a significant improvement over using SF8.

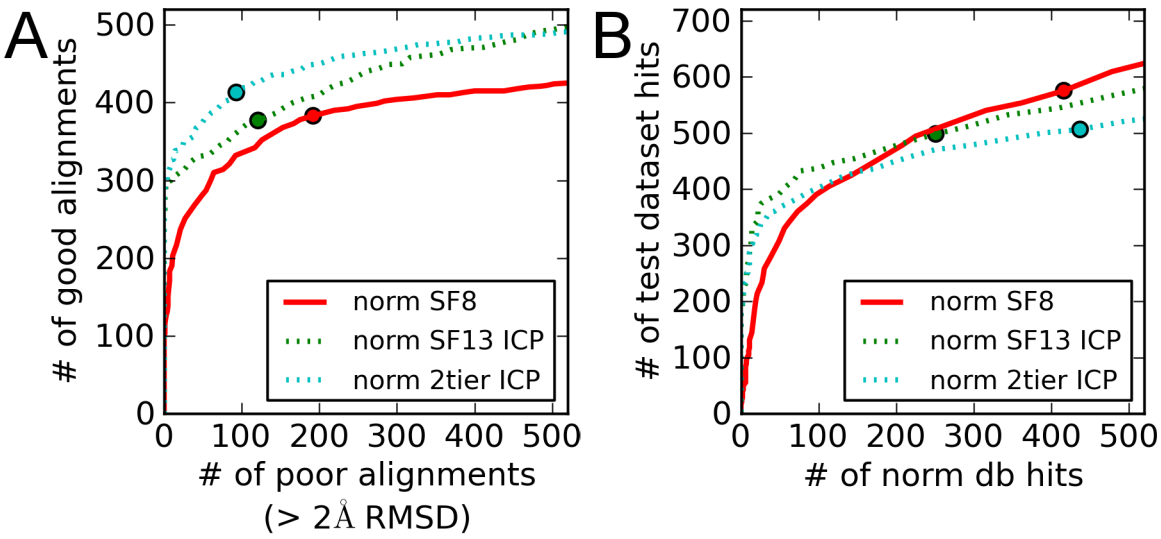


Figure 25: ROC-like curves showing SimSite3D performance when SF8, SF13 & ICP, and two-tiered scoring & ICP to select and refine the best scoring alignment for each pair of sites in the test datasets. Panel A shows the ability of the scoring functions to predict if the alignment error is significant for the best scoring alignments from the test datasets. Panel B shows the scoring functions' performance with respect to discriminating between the best test family alignments and the best alignments of query sites to those in the normalization dataset. The norm curves are the results when the raw scores are normalized using the mean and standard deviation of the query site's scores for the 140 sites in the normalization dataset. The dots on the norm curves denote the point where the score is 1.5 standard deviations better than the mean score with respect to the normalization dataset. Please note that the data in the two panels differs as is noted by the axes' labels.

The ROC-like curves comparing two-tiered scoring to SF8 and SF13 lend support to the idea that two-tiered scoring & ICP is a good compromise between using SF8 and using SF13 & ICP (Figure 25). In panel A, one can see that on the test datasets, two-tiered scoring & ICP does rank as significant more good quality alignments than does SF13 & ICP at the score threshold of 1.5 standard deviations better than the mean. In addition, two-tiered & ICP predicts fewer poor alignments as being significant when compared with SF13 & ICP; which is about half the number of poor alignments that were predicted to be significant by SF8. In terms of the ability to discriminate between test dataset hits and normalization dataset hits, the performance of two-tiered & ICP is better than that of SF8. The reason is the percentage of test dataset two-tiered & ICP hits with poor alignments is much lower than that of SF8 (Figure25).

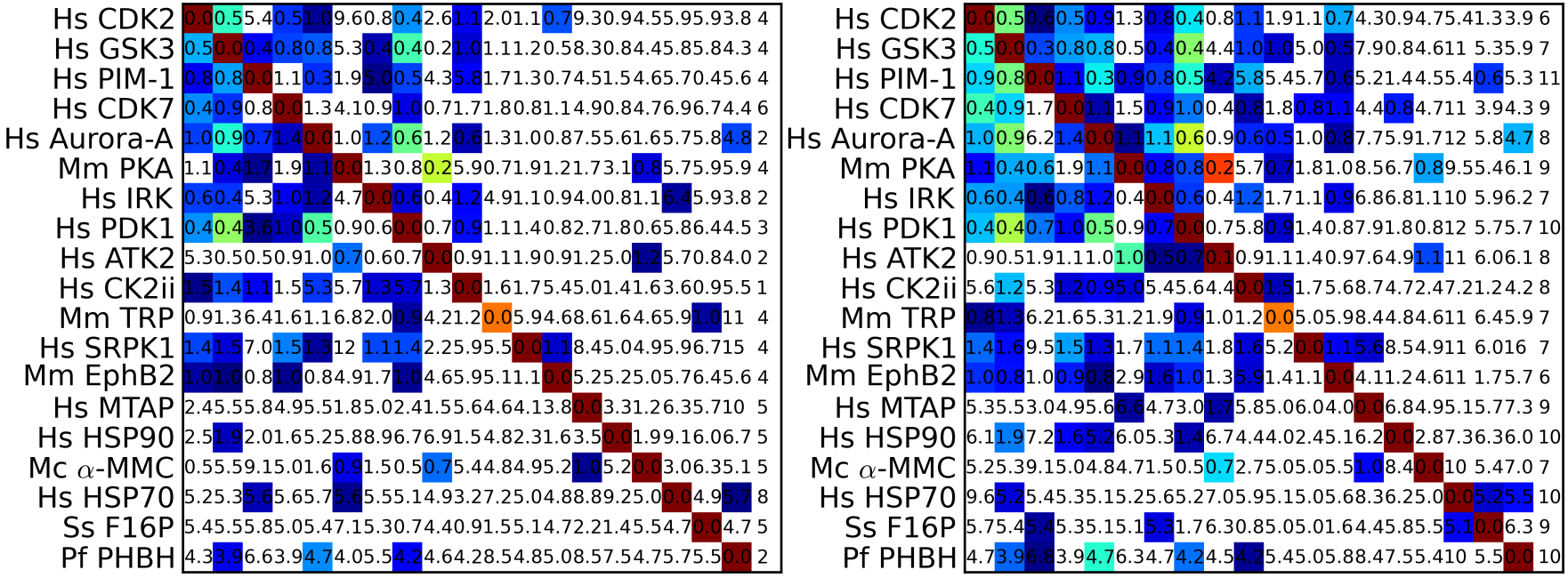


Figure 26: SimSite3D score matrices showing the difference in score significance and alignment quality between SF13 & ICP and two-tiered & ICP on the adenine test dataset. The left and right matrices are the score matrices for SF13 & ICP and two-tiered scoring & ICP, respectively.

Since two-tiered scoring & ICP selects the best alignment for each site pair using SF13 as the final sieve, direct comparisons can be made between SF13 and two-tiered scoring to help explain the presented results. In particular, the set of candidate alignments is the same for both scoring methods, but in two-tiered scoring, SF13 has at most ten alignments to rank. This means that before ICP, the alignment chosen by SF13 score applied to all alignments, will have a score better than or equal to the score of the alignment chosen by two-tiered score. Therefore, in the interest of a simple calculation, assume that, on average, when applied to the test datasets, the two methods will choose the same alignments or alignments with very similar raw scores. What we would like to address is how much of an effect does two-tiered scoring have on the mean and standard deviation of the scores with respect to the normalization dataset.

Upon viewing the ranges of the means and standard deviations for the 58 query sites (versus the normalization dataset) using the two scoring methods, it is clear that the mean scores are significantly better when using SF13 as opposed to two-tiered scoring (Figure 27). The impact of using SF13 over two-tiered scoring on the standard deviations is less clear. It is reasonable to argue that the reason SF13 does more poorly on the adenines dataset than two-tiered scoring is using SF13 clearly shifts the mean normalization dataset score to a better value, and as a result, the score threshold of -1.5 is more stringent. The fact that SF13 has about half of the normalization dataset hits at -1.5 than SF8 or two-tiered scoring indicates that there are fewer high scoring outliers from the normalization dataset using SF8 than SF13.

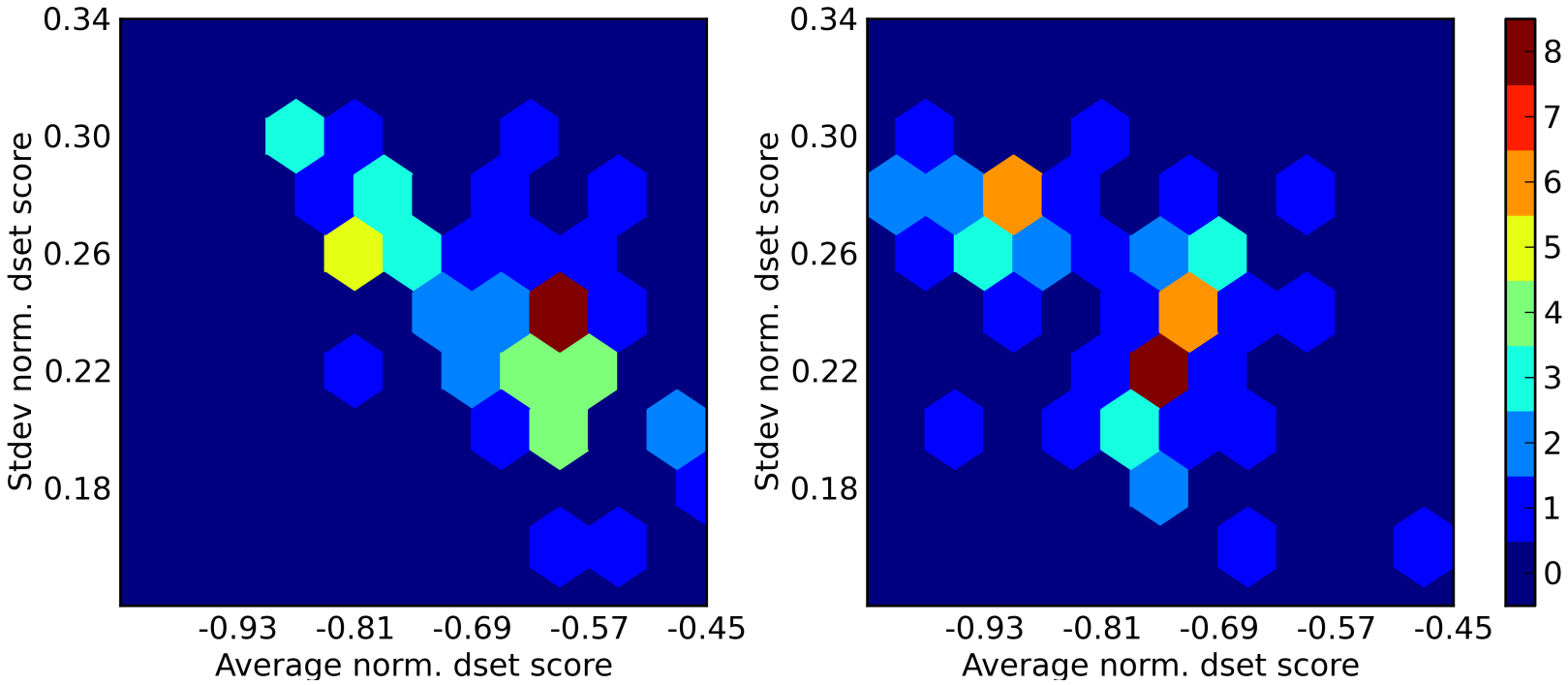


Figure 27: The three test datasets have a total of 58 query sites. For each query site, we compute the mean and standard deviation of the 140 scores versus the normalization dataset. A hexagonal grid is used to plot the frequency of the means and standard deviations. In short, the center of each hexagon is used as a grid point. The hexagons are colored by the number of query sites (samples) for which the center of the hex is the nearest grid point (i.e. nearest neighbor). The left and right plots show an estimate of the distribution of the means and standard deviations of the 58 query sites when scored with two-tiered scoring & ICP and SF13 & ICP, respectively. It is easy to see that the score averages are shifted higher when using SF13 alone.

4.3.2 Remarks

We have presented a two-tiered scoring method that captures some of the gains of including the surface complementarity to assess site alignment quality and site similarity. The reason that SF13 appears to outperform two-tiered scoring in terms of discriminating between test dataset hits and normalization dataset hits is twofold. First, two-tiered scoring typically chooses the same alignment as SF13 for within protein family hits. Second, on the normalization dataset, SF13, on average, chooses better scoring alignments than does two-tiered scoring, and this fact causes the normalization scores above the mean to be closer to the mean score (27).

Based on its results on the three test datasets and its greatly reduced computational demand relative to SF13, we recommend using the two-tiered scoring method for high-throughput screening.

Until this point, we have been assuming that hits from the normalization dataset are all false positives. However, that is not entirely true. In terms of the adenines, a number of proteins in the normalization dataset bind adenine, and the dataset contains a CDK2 structure with an inhibitor. Also, the benzimidazole inhibitor site of a poly ADP-ribose polymerase structure (PARP) (PDB: 1EFY) is very similar in shape to the adenine sites of the kinases and includes the same main chain motif kinases use to recognize adenine N1 and N6.

4.4 Search for More Optimal Surface Parameters

The results for the molecular surfaces have been presented for a specific set of molecular surface generation parameters. The probe radius used is 1.4 Å as it is close to the van der Waals radius for water (1.36 Å) as specified by Li and Nussinov [68]. The density of vertices used is the default MSMS value for proteins of 1 vertex per $Å^2$.

We are interested in the effects that modifying the parameters will have on surface

comparisons. An increase in the probe radius would omit water sites where the radius of the site is less than the probe radius. Similarly, a decrease in the probe radius is likely to result in a more nodular surface as smaller cavities than 1.4 Å radius will contribute to the shape of the surface. In other words, the fractal dimension of the molecular surfaces is expected to be inversely related to the probe radius. To test if a different value of the probe radius might yield better results, probe radii of 1.2, 1.4, and 1.6 Å are used.

In order to have an aesthetically pleasing molecular surface, many scientists choose to use a vertex density of 5 vertices per $Å^2$ of molecular surface. As the previous surface results were determined using a vertex density of 1 per $Å^2$, important surface features could be missing (due to the coarse sampling) from the binding site surfaces. Therefore, the molecular surface vertex density is sampled at the rates of 1, 3, and 5 vertices per $Å^2$. Finally, nine molecular surfaces were generated for each site in the three test datasets and in the normalization dataset (one surface for each parameter combination).

4.4.1 Results

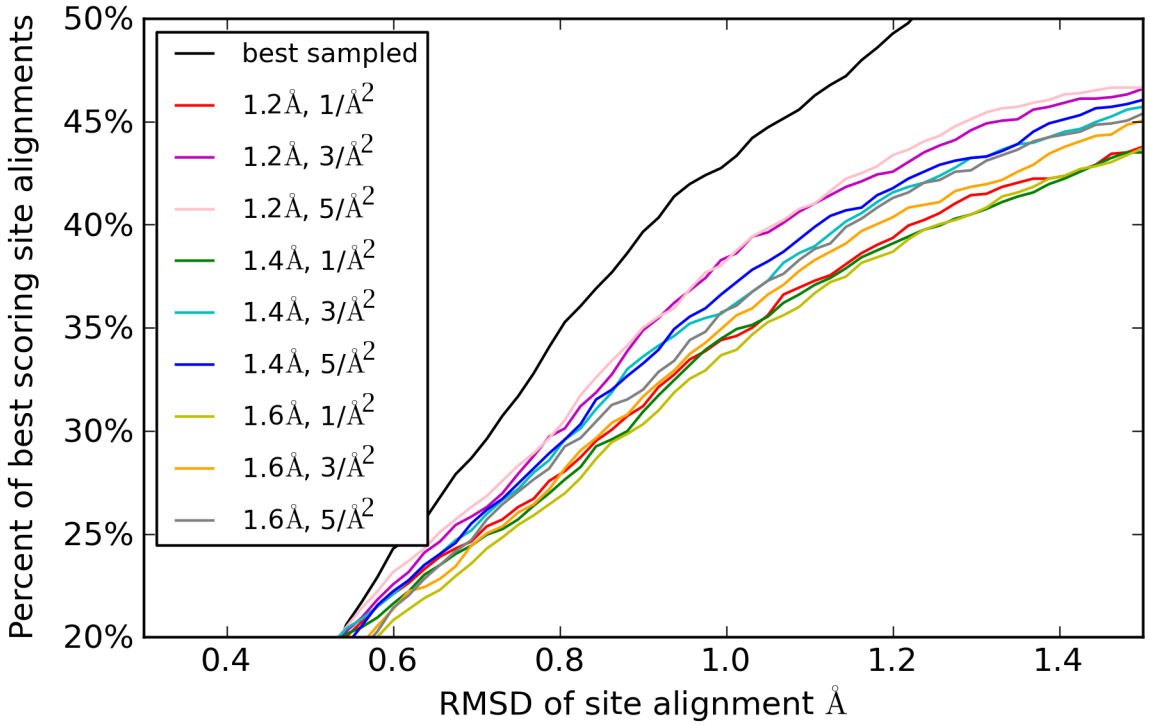


Figure 28: Catchment (cumulative distribution) curves for SimSite3D using two-tiered scoring & ICP with nine distinct pairs of molecular surface parameters on the three test datasets. In this plot, the range is focused on the region where the differences are most apparent. A given point on a curve represents the percent of test sites for which the best scoring alignment has an RMSD of alignment less than or equal to the corresponding value on the horizontal axis. The numbers in the legend indicate the probe radius in Å and the number of vertices per Å^2 of surface area.

It is easy to see that using a probe radius of 1.2 Å and at least three vertices per Å^2 of molecular surface area performs significantly better than searches with other surface parameters over the range of $[0.75, 1.5]$ Å RMSD of site alignment. In particular, the default values of 1.4 Å probe radius and 1 vertex per Å^2 (the red curve) catches a much lower percentage of alignments at any RMSD value in the range $[0.75, 1.5]$ Å than using a probe radius of 1.2 Å and three vertices per Å^2 (the purple curve).

Sampling three vertices per Å^2 does incur a significant computational cost. On average, the total computational time to compare two sites is about one second per pair of test

dataset binding sites with a vertex density of one per $Å^2$. When a vertex density of three per $Å^2$ is used, the average computational time increases to about three seconds per pair of binding sites in the test datasets.

For the most part, it is difficult to choose one of the nine sets of surface parameters in terms of site scoring performance. Increasing the number of points sampled (left plot in Figure 29) does help improve the quality of some alignments, but only at less stringent score tolerances. Using a larger probe radius and a coarse sampling appears to be beneficial in distinguishing between test dataset hits and normalization dataset hits (right plot in Figure 29). The initial best scoring alignment for each site pairs tends to depend on the surface parameters (i.e. the differences in the data are not due to ICP alone). When considering the three plots, it is not immediately clear which of the nine parameter sets is the best choice.

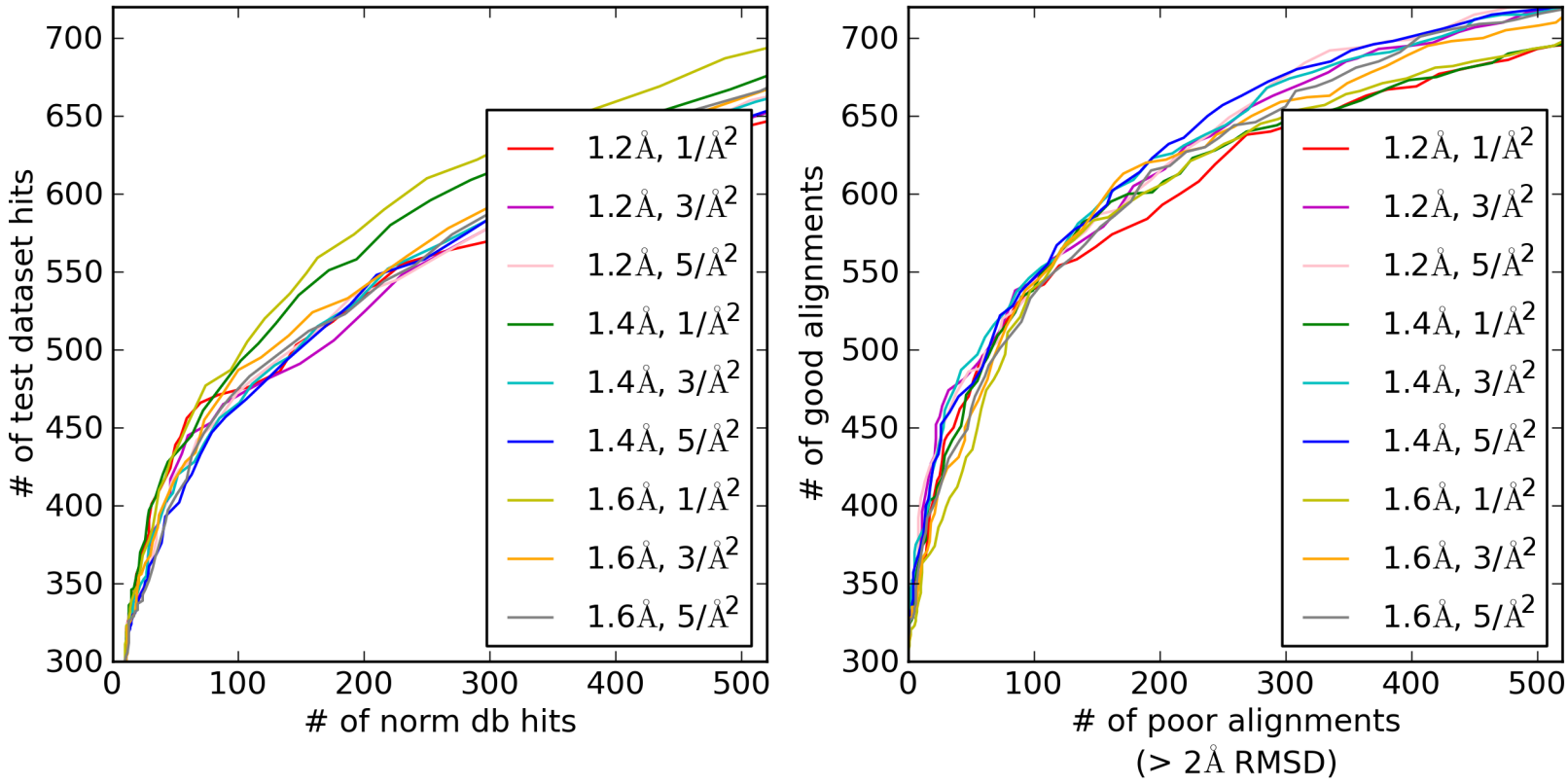


Figure 29: ROC-like curves showing the performance of SimSite3D, using two-tiered scoring & ICP, on the three test datasets for 9 pairs of molecular surface parameters. (SimSite3D was run 9 times. The best scoring alignments, in many cases, did differ between the runs.) In the legends, the first parameter is the probe radius in Angstroms, and the second parameter is the average number of surface vertices per $Å^2$ of surface area. The left plot shows the ability of SimSite3D to discriminate between good and poor alignments in the test datasets (vertical and horizontal axis, respectively). The right plot shows the ability of SimSite3D to discriminate between pairs of aligned test dataset sites (vertical axis) and pairs of test dataset query sites and normalization dataset sites (horizontal axis). Note: the data plotted differs in the two plots as is noted by the axes' labels.

4.4.2 Discussion

It is clear from Figure 28 that using a probe radius of 1.2 Å and at least three vertices per $Å^2$ results in better overall alignment accuracy than the other seven sets of surface parameters. However, the ROC-like curves (Figure 29) seem to indicate that using a small probe radius (1.2 Å) or finer sampling is counter-productive since using such parameters causes one to miss a number of test dataset hits over the range where all nine parameter sets have approximately the same ability to choose low error alignments.

Which parameter set to use depends on one's goal and the resources at hand. Given that a probe radius of 1.4 Å and a surface density of one vertex per $Å^2$ is one of the better performing parameter sets for site pairs with scores better than 1.5 standard deviations above the mean and using a finer sampling of the surface requires more computational resources, it is recommended that the original surface parameters be used in SimSite3D. On the hand, if the three test datasets are sufficiently general, it is likely that given a better scoring function, the use of a probe radius of 1.2 Å and a surface density of about 3 vertices per $Å^2$ would be beneficial since the alignment accuracy is substantially better than using larger probe radii or a coarser mesh.

4.5 Improving Alignment Sampling

One potential way to improve the performance of an object recognition method is to increase the sampling accuracy, such that, the error of the candidate alignment with the smallest alignment error is reduced. To illustrate this, suppose there exists an oracle [99] that provides a yes/no answer as to whether two objects, when aligned as given, are similar. Then, any object recognition method trained using such an oracle would still fail for those similar objects that were not reasonably well aligned.

4.5.1 Relaxed Triangle Geometric Constraints

The method to generate candidate alignments was fixed early on in the design process (Section 3.1.2). It is possible that the additional data and experience gained afterwards can be used to provide the scoring functions with more alignments with low registration error. The alignment method is based on three pairs of corresponding points chemical points. Three points from a site can be considered as the vertices of a triangle. In Section 3.1.2, we noted that the bounds on the triangle features are: perimeter in $[9, 13]$ Å , longest edge length in $[3.5, 4.5]$ Å , and shortest edge length in $[1.8, 3.5]$ Å .

The number and quality of binding site datasets has increased since those bounds were determined. One might inquire if loosening the bounds on the triangles would result in a more accurate method at the cost of considering more alignments. In order to have some data to guide the loosening of the bounds, we considered all of the possible three point correspondences for the adenine test dataset (i.e. no bounds on the triangle sizes, but still required corresponding points to chemically complementary). For each possible set of correspondences the three geometrical features and the RMSD of alignment was recorded. For each query site, the triangle features and RMSD of the ten alignments with the least alignment error were saved.

Box plots were used to view the range of the triangle features for each query site. Based on the box plots of saved features for the adenines dataset, the bounds on triangle sizes were loosened to have the perimeter in $[9, 16]$ Å , the longest edge length in $[4, 7]$ Å , and the shortest edge length in $[1.8, 4]$ Å . To test the impact of additional candidate alignments, SimSite3D was run with the alignments based on the loosened triangle bounds and the alignments were scored using two-tiered scoring & ICP.

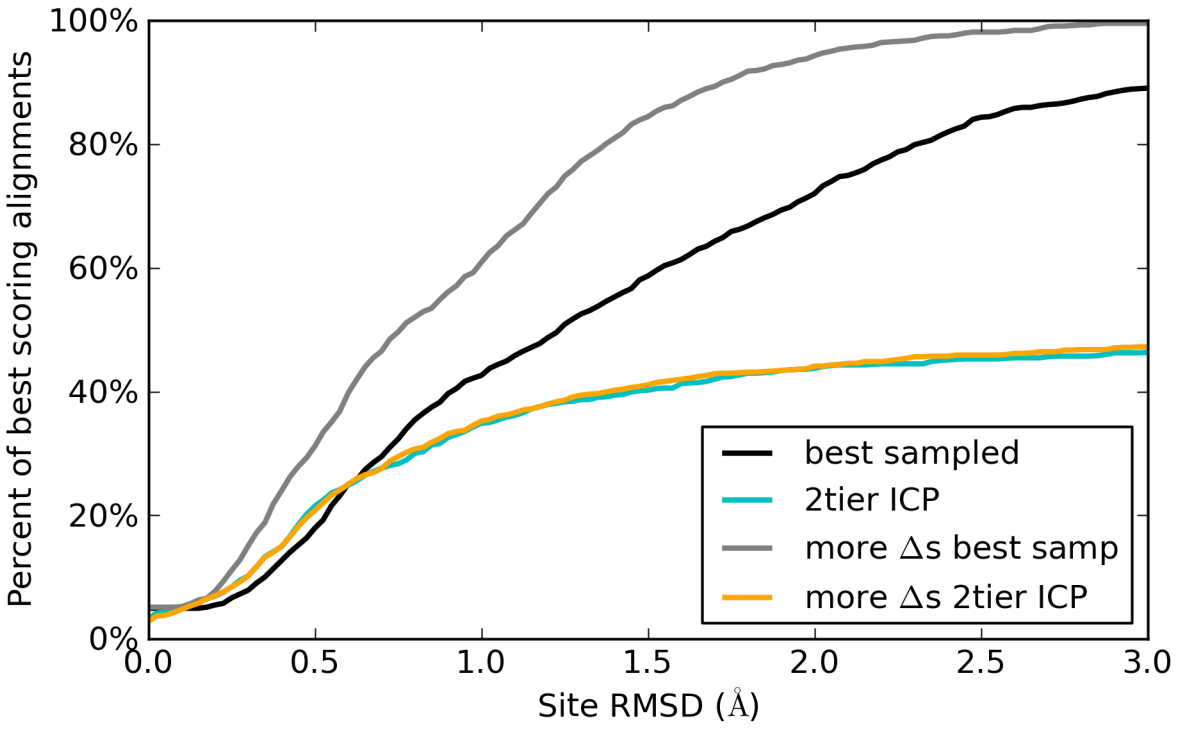


Figure 30: Catchment curves showing the effects of increasing the allowed triangle sizes for three point correspondences. Notice the improvement in best sampled alignment, but no significant improvement in best scoring alignment.

Notice that the error of the best sampled alignment per pair of sites does show a dramatic reduction (Figure 30) as more than 80 percent of the pairs of sites have candidate alignments with error less than 1.5 Å RMSD as opposed to 60 percent when the original triangle feature ranges are used. However, there was no appreciable change in the percentage of best scoring alignments at any significant RMSD value (< 3.0Å). Since the number of alignments using the relaxed bounds on triangle sizes is about ten times that of the original bounds, it is recommended that the bounds be kept at their original values until a scoring function is found that can take advantage of the additional low error alignments.

4.5.2 Grid Sampling of Pose Space

Over the course of the project it was observed that many of the candidate alignments found using the triangle matching method (Section 3.1.2) gave a very large number of dreadful alignments that had only three pairs of point correspondences. In order to test if a different sampling method might increase the performance of SimSite3D, a grid based sampling method was used to almost uniformly sample the pose space of the binding sites for translation values near the centroids of the binding sites.

One must be careful to sample the space of rotations correctly. The reason is that the space of rigid rotations is not a Euclidean space, but is the special orthogonal group of 3×3 matrices, SO(3). Therefore, although the space of rotations can be parameterized by 3×3 rotation matrices, quaternions, three Euler angles, an arbitrary axis of rotation and an angle, etc., it is a challenge to deterministically sample SO(3) in a uniform manner. The reason is SO(3) is similar to the 4 dimensional unit sphere (S^3) since the space of unit quaternions is exactly S^3, and the unit quaternions (and, of course S^3) provide a double covering of SO(3).

Therefore, the problem reduces to finding a deterministic method to uniformly sampling the sphere S^3. Although such a method has been sought for more than 60 years [35, 70, 85], at the present, there is no known method that provides a truly uniform and deterministic sampling of the spheres S^n for $n > 1$. However, there is a recent method (ISOI) [105], based on the Haar measure [73], that is shown to outperform all previous methods in producing a deterministic, almost uniform sampling of SO(n) [105].

The grid based method has been implemented as follows. The centroid of the query site is placed at the center of each heavy atom in the dataset ligand. The ISOI compute program was used to generate the level 2 grid for SO(3) which has ~ 4500 grid points. The grid based method was tested for one query site because of the large number of generated candidate alignments and the goal was to illustrate whether a more uniform sampling of alignments might help with the assessment of site similarity.

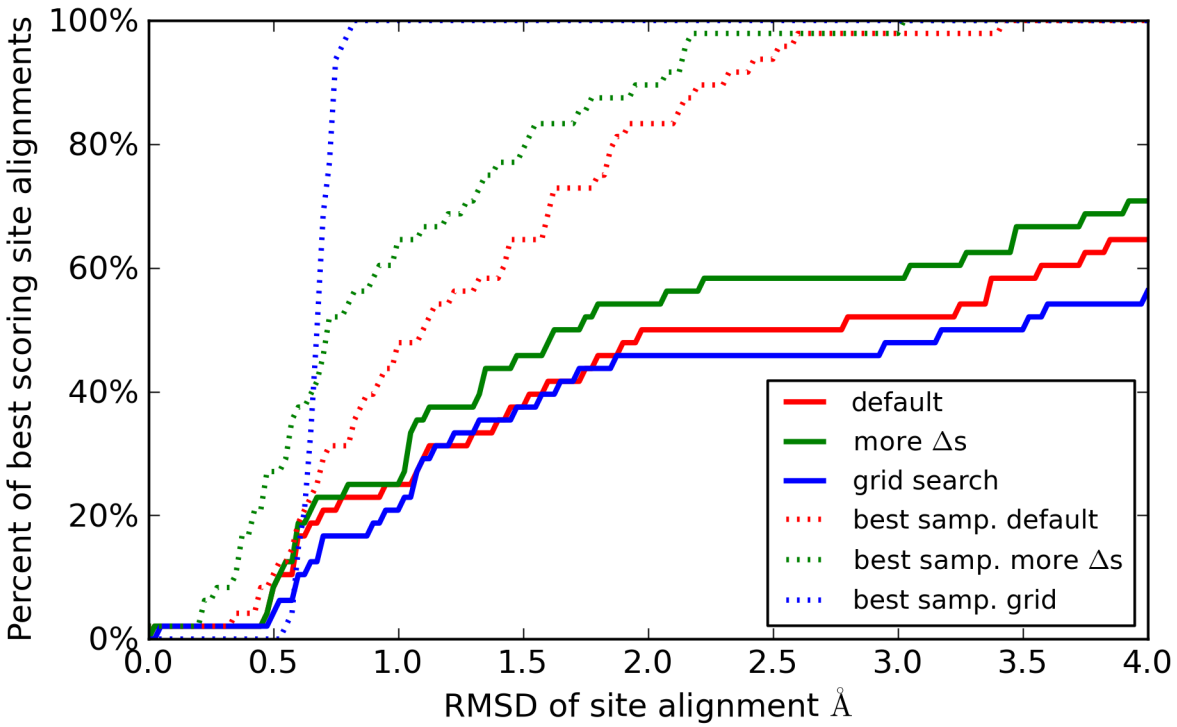


Figure 31: Scoring catchment plots showing the impact of generating candidate alignments on a grid and increasing the triangle bounds used in the triangle matching method. The best scoring alignments were chosen by the two tiered scoring method and were refined using ICP (the best sampled alignments were not refined). The data is the alignment error of the best scoring (or sampled) alignment for the H. sapiens CDK2 adenine binding site (PDB: 1B38) versus the dataset sites in the adenine dataset and the adenine sites in the normalization dataset (48 total sites that each contain an adenine site). A particular point on a curve gives the percent of site pairs for which the best scoring alignment had an error less than or equal to the RMSD value (horizontal axis).

It is easy to see that loosening the range of values allowed for features of the correspondence triangles or using a grid based alignment method results in having candidate alignments with less error than using the original triangle matching method in almost all cases. Notice that, as expected, the accuracy of the best sampled alignments of the grid-based sampling method is essentially independent of the site features. Again, as one might expect from the previous section, an increase in sampling accuracy did not result in the scoring function recognizing additional sites as similar.

4.5.3 Comments

The implemented grid based sampling is unbiased, and the error of the best sampled alignment is approximately the same for all 48 site pairs. On average, the loosening of the range of triangle features increased the number of candidate alignments by at least 10-fold, and the grid based sampling increases the number of candidate alignments by at least another factor of 10. Given the increase in computational cost and no appreciable improvement in the final results for the three test datasets, we do not recommend changing the sampling method until the site representation and/or scoring methods are improved.

There are two possible explanations why increasing the number of candidate alignments does not provide a decrease, on average, of the alignment error of the best scoring alignments. A key part of protein-ligand interactions has not been considered in our work. Correctly modeling the interplay between water molecules and protein-ligand complexes is required to accurately model and explain protein-ligand binding affinity [29]. Water molecules were not included in the binding site comparisons in this dissertation as the determination of which water molecules are important for binding is still an active area of research, existing methods are computationally expensive, and the false postive/negative rates are significant. A major challenge is that some water molecules can be displaced upon ligand binding, and the displacement depends upon which ligand binds. Additionally, some water molecules can be absolutely critical for ligand recognition while others are relatively negligible. Because of these considerations, the inclusion of water molecules in the protein-ligand binding site comparison problem is expected to add an additional layer of relatively high noise. For these reasons, we chose to focus on how well binding sites can be compared without considering water to have a method to compare and contrast to when future methods are built.

A second explanation is the features (scoring function terms) used to assess site similarity are akin to global averages over all the relative distances between corresponding

points of the same type (i.e. surface vertices, chemistry points, etc). Ideally, there would be a good metric to measure the similarity of two objects based on the relative position of feature points without resorting to sums such as RMSD or kernels. A possibility is to compare two sites versus the query by considering the overlap between the two sets of query points matched. However, computational comparisons of these sets of points has proved to be unsatisfactory because, at the present, we do not know the relative importance of interactions between the protein and ligand chemical groups and computational predictions of relative importance are at best expensive and an area of active research. Finally, if all of the dataset binding sites have ligands bound, explicitly considering the dataset protein-ligand interactions and whether the query protein can adopt such conformation might yield better performance than ignoring the ligand information (as is done in this dissertation).

4.6 Polar Atom Caps

As noted previously, part of the point clouds in a site map is used to represent the positions and types of atoms that would make hydrogen bonds with the protein. These points are a very sparse sampling of the SLIDE volume for allowed ligand hydrogen bond geometry with respect to the protein structure. In this section, we use spherical caps to represent the SLIDE volumes. Similar to computing the complementarity of the molecular surfaces, we can use polar caps from one site and a set of sample points on the caps from the second site to estimate the hydrogen bond similarities of the sites. Since the points in the point clouds are sparse, it is likely that determining the corresponding points using the caps will result in less correspondence error for the hydrogen bond points.

There are several advantages to the spherical cap representation:

- The method to find the closest point on a spherical cap can be defined and computed analytically and is relatively efficient to compute.

- The representation is analytical and does not depend on the parameter values, the parameters could be adjusted at match time.
- If desired, distinct parameters could be easily specified for each distinct protein atom type (e.g. His ND1's values may differ from those of His NE2 and Arg NZ).
- If the sites in a screening dataset are represented using the analytical representation, the sampling density for the caps in the query site may be changed at match time without the need to recompute the representations of the dataset sites.

This spherical cap representation and closest point method has been implemented in SimSite3D.

4.6.1 An Analytical Representation of a Cap

The analytical modeling of a polar spherical cap is as follows. Given a protein polar atom A, the position x of the atom's lone pair of electrons or hydrogen atom can be computed by rules similar to those used to compute the central point of a polar point group in the point cloud representation. Let $\overrightarrow{N}$ be a normal vector in the $A \rightarrow x$ direction. Let S be a sphere centered at the center of A and having a radius of 3.0 Å . Let P be the plane defined by the normal $\overrightarrow{N}$ and a point p_n that lines on the ray starting at the center of A and is parallel to $\overrightarrow{N}$ (p_n is a dependent parameter that depends on the maximum allowed angle α between $\overrightarrow{N}$ and a ray from the center of A; e.g. the minimum donor-hydrogen-acceptor angle for a hydrogen bond). Then, the spherical cap S_C is that portion of S that is above the plane P.

As with the point cloud representation, some regions of the cap S_C may be invalid as placing a polar atom in such regions would lead to large overlaps between the placed atom and one or more atoms in protein. For this reason, the volume of a ball with radius 2.5 Å and centered at each nearby atom's center is subtracted from the spherical cap (Fig-

ure 32). The remaining portion(s) of the spherical cap, if any, are taken to be the polar representation for that lone pair of electrons or polar hydrogen atom.

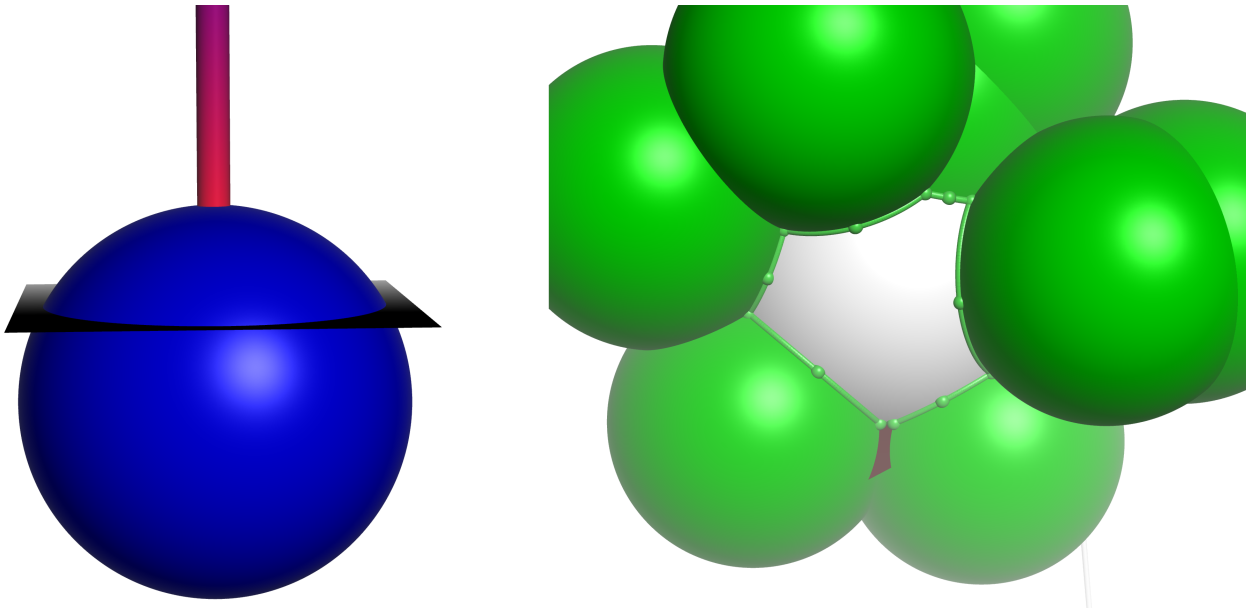


Figure 32: An example of a spherical cap representation of an allowed hydrogen bonding volume. On the left we see a sphere cut by a plane with the tube representing the normal to the plane. This normal would be parallel to the line segment between two atoms participating in a "linear" hydrogen bond. On the right is a cap that is partially occluded by spheres of influence of neighboring atoms. Each green sphere represents the volume in which one cannot place the center of an atom, from another molecule, as it would severely overlap with the corresponding protein atom. The small red shape is part of the plane defining the cap that is not occluded by a neighboring atom. The visible portion of the cap represents the surface where ligand atoms could sit and form a hydrogen bond with the corresponding protein atom.

To compute the closest point on a cap to a given sample point, suppose that we are given a sample point p and an cap S_C which is part of the circle S with center A and radius r.

1. Compute the unit direction $\overrightarrow{Ap} = (p - A)/\|p - A\|$.

2. The closest point p^* on the cap may be computed by projecting the point onto the sphere by $p^* = r\overrightarrow{Ap} + A$.

3. Check if the projected point p^* is above or below the plane P by computing the signed distance d' from p^* to the plane P.

4. If p^* is below the plane, it can be projected to the closest point on the cap by first projecting p^* to the closest point on the plane $p' = d'N + p^*$.

5. Project p' to the closest point p'' on the circle in the plane (defined by the intersection of the plane and the sphere S). This projection is computed by projecting p' to the closest point p'' on S (note that p'' is restricted to the plane unless $p' = A$; the reason is that the only point at which the sphere s', centered at $p' \neq A$, touching S at p'', and contained in S, will come in contact with S is at p'').

There is a maximum correspondence distance, and if at any step, the closest point distance is greater than the maximum allowed, the sample point p is denoted as not having a correspondence on the cap S_C.

Now that we know how to compute the closest point p'' on the cap with respect to a given sample point p, p'' must be moved if it is inside one or more of the neighboring balls. These moves require some reasoning about circles and spheres in three dimensions. It is well known, that there are three types of intersection between two spheres: no intersection, a point, or a circle [88].

Definition. A **circle of intersection** or **iCircle** is that circle representing the intersection between two spheres.

In our case, we only consider those neighbors of the spherical cap S_C for which the intersection of the surface of the ball (i.e. sphere) and the sphere S is a circle and at least two points on the circle are on the spherical cap S_C. If part of the circle of intersection is not on S_C, we can use the circle of intersection and the plane defining the spherical cap to define the arc of intersection between S_C and the neighbor. Finally, we check each arc of intersection to remove the portions of the arc which are inside the ball of any of the neighboring atoms.

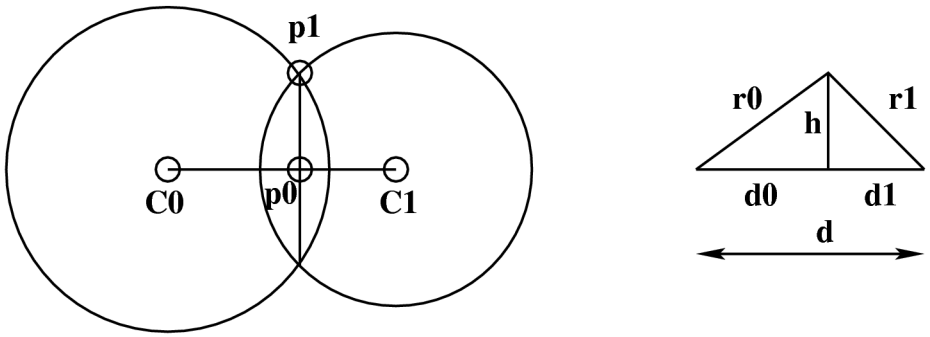


Figure 33: 2D figure to illustrate computing the iCircle parameters. Suppose we want the circle of intersection between spheres $S_0 = S_0(C0, r0)$ and $S_1 = S_1(C1, r1)$ that we know intersect, and they do not contain each others centers. Let $d_0 = \|C_0 - p_0\|$, $d_1 = \|C_1 - p_0\|$, $h = \|p_0 - p_1\|$, and $d = d0 + d1$. We are looking for p_0 and h.

Suppose we want the circle of intersection $I_{0,1}$ between spheres $S_0 = S_0(C_0, r_0)$ and $S_1 = S_1(C_1, r_1)$ that we know intersect, and they do not contain each others centers. We know the centers and radii of the spheres and the distance between the two centers, but we seek the radius h of the intersection circle and its center p_0. We can solve for $d_0 = \|C_0 - p_0\|$ and $h = \|p_0 - p_1\|$ by using the illustrations in Figure 33 and trigonometric rules. Using the law of cosines, substitutions, and algebraic operations we can write $d_0 = \frac{d^2 + r_0^2 - r_1^2}{2d}$ The center of the circle of intersection is given by moving from C_0 to C_1 by a distance of d_0; that is $C_0 + \frac{d_0(C_1 - C_0)}{\|C_1 - C_0\|}$. The radius is found using Pythagoras' theorem; $h = \sqrt{r_0^2 - d_0^2}$.

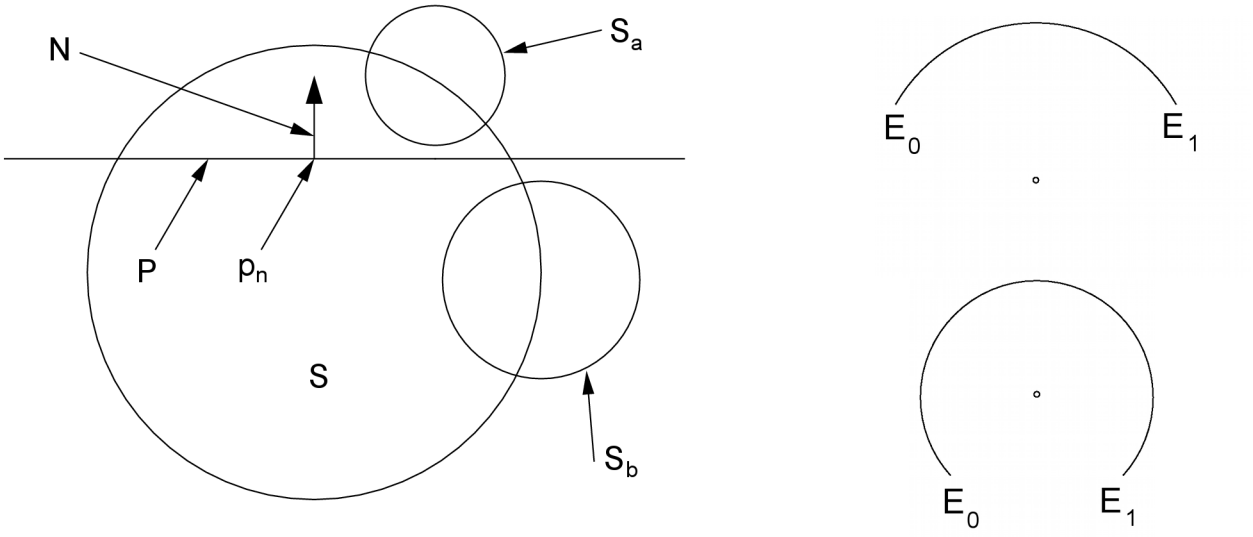


Figure 34: 2D figures to illustrate iCircle case and arc cases. On the left is a sphere S with 2 intersection spheres S_a and S_b which are the spheres on which two iCircles (I_a and I_b, respectively) lay. The line P is the plane used to define the cap, and P is itself specified by the normal N and the point p_n. Clearly, if the center of a sphere is above the line by at least the radius of the sphere, or is below the line by at least the radius of the sphere, it is impossible for the spheres to intersect the plane. On the right is an example showing the 2 cases for arcs; those with arc length less than πr radians (top arc), and those with arc length greater than or equal to πr radians (bottom arc). Here r is the radius of the corresponding circle and the points represent the center of the corresponding circles. It is easy to see that, if the arc is closed by the drawing the chord between an arc's end points E_0 and E_1, when the arc length is less than πr radians the closed curve will not contain the center of the corresponding circle.

We now need to check where the intersection circle $I = I(p_0, h)$ lies with respect to the plane P (with equation $\overrightarrow{N}X + p_n = 0$) and sphere S used to define the spherical cap S_c. First, we must determine if none of, part of, or all of I lies on the spherical cap S_c. To do this, find the signed distance from the center p_0 of the intersection circle to the plane P. If the signed distance is $\leq -h$, then the intersection circle I cannot intersect with the cap S_c and the intersection can be safely ignored. If the signed distance is $\geq h$, then the intersection circle I is fully contained in the cap S_c (i.e. does not intersect with the plane P). In the case where the signed distance is in the range $(-h, h)$, we handle the intersection by keeping only the arc A_I of the intersection circle I that is above the plane

P.

To compute the initial arc A_I, we first check to ensure that the intersection circle I does indeed intersect nontrivially with the plane P [2]. We do this by checking if the line of intersection L_I between the plane P_I that contains the intersection circle the plane P used to define the spherical cap, passes through the sphere $S_I = S_I(p_0, h)$. If L_I does indeed pass through the sphere S_I, the two points of intersection are the end points (E_0, E_1) of the initial arc A_I. To find the midpoint of the arc, define the unit vector N_{A_I} in the direction of $(E_0 + E_1)/2 - p_0$. If the arc A_I's angle is greater than π radians, the dot product between N_{A_I} and the normal to the cap plane N is negative, and N_{A_I} must be multiplied by -1. The midpoint of the arc is found by projecting the center p_0 of the intersection circle I to the circle I in the N_{A_I} direction.

Finally, we must check each arc and remove those portions of the arc that fall inside any of the neighboring balls. This is implemented by sequentially checking all of the intersection spheres. For a given intersection sphere S_i and arc A_i, we must check if it intersects the intersection sphere S_j and arc A_j for all $j \neq i$. If S_i and S_j do not intersect, that pair does not need to be considered. Otherwise, remove all arcs from S_i that are fully contained in S_j. If S_i is entirely contained in S_j, then S_i and all of the associated constructs are removed from the cap representation. Compute the line of intersection $L_{i,j}$ between the corresponding planes of the two intersection circles. If $L_{i,j}$ does not intersect with S_i, this pair does not need to be considered further (as the intersection circles do not intersect).

2 The signed distance based heuristic fails for intersection circles that almost intersect the cap's plane

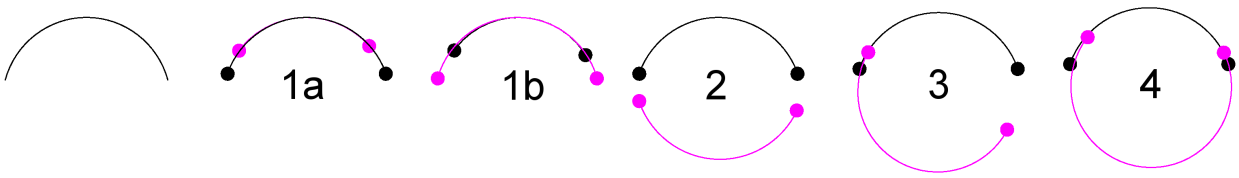


Figure 35: Four cases for the intersection of two arcs from the same circle. On the left is an example arc. The numbered arcs show the 4 cases. Case 1 is recognized by exactly one of the arcs containing both end points of the other arc; in 1a the black arc contains the magenta arc, and in 1b the magenta arc contains the black arc. For case 1 the intersection of the arcs is the arc with the shorter arc length. Case 2 is no intersection and it is easy to see the neither arc contains an end point from the other arc. Case 3 is partial overlap between the two arcs where both the magenta and black arcs contain exactly one end point of the other arc; the intersection of the arcs is the arc shared between the two endpoints which lie on both arcs. Case 4 occurs when both arcs contain both end points of the other arc; The intersection is two arcs, and they are the shared arcs between the end points from the two different arcs.

If the intersection circles do indeed intersect, the intersections need to be addressed. Suppose that $L_{i,j}$ does intersect S_i at two distinct points labeled E_0 and E_1. Then, the intersection circle I_i is partitioned into two arcs by $L_{i,j}$. These arcs have as their end points E_0 and E_1 and differ in that they have opposing mid points. The arc whose mid point is inside the sphere S_j is the arc that is removed by S_j, and is called the "rm" arc. The other arc is termed the "keep" arc. Finally, for each arc remaining for the current intersection circle I_i, keep only the portion(s) of the arc that intersects with the "keep" arc (see Figure 35). Continue processing for all $i \neq j$, and at the end the remaining arcs, on the spherical cap S_c, are those that are not inside any of the neighbors' volumes.

4.6.2 Determining the Closest Point on a Cap

Given the machinery from the previous section, it is relatively straightforward to compute the closest point on a cap for a particular sample point.

1. Project the point to the closest point on the cap.

2. Check all intersection circles to determine if the projected point is inside the circle.

3. If the projected point is not inside an intersection circle, that point is taken as the closest point for the sample point.

4. Otherwise, for each intersection circle that contains the projected point, project the point to each arc in the circle.

5. Take the projected point that is closest to the sample point as the closest point.

4.6.3 Training a Scoring Function

Here the complementarity of two site's sets of hydrogen bond caps and molecular surfaces are used to estimate the quality of their alignment and their similarities. Given a dataset site with hydrogen bond caps described by the analytical representation and a query site with the caps sampled at quasi-regular intervals, approximate the best correspondence for each query point as the closest point in the dataset caps with complementary chemistry and a distance of less than or equal to 1.5 Å . For each pair of corresponding points, consider its contribution as 1.5 minus the distance between them, and multiply that difference by the dot product of their corresponding directions. As when computing the complementarity of two hydrogen bond points clouds, form two sums: one when both points are either acceptors or donors and another sum for the cases where at least one point can be both a donor or acceptor.

These two sums and the surface complementarity (surface point RMSD) can be considered as three terms in a linear scoring function used to predict -1 over binding site RMSD [97]. The training and validation steps are the same as those presented in the previous chapter with the exception that each feature was scaled to $[0.0, 1.0]$ where 0.0 is no value and 1.0 is 100 percent of the query site's maximum value for that feature. The weights determined for the terms may be found in Table 11.

Table 11: The weights determined for a linear scoring function to predict -1/(site RMSD) from the 2 hydrogen bond cap terms and the surface complementarity term. Here "Constant" is the constant term (intercept), AA & DD sum is the cap sum for pairs of acceptor and donor points, N* sum is the cap sum for pairs of corresponding polar points where at least 1 of the points is a doneptor point, and Surf. RMSD is the RMSD of the corresponding molecular surface points. Here we see that when the terms are constrained to be in the range $[0.0, 1.0]$ then the polar term and surface term have approximately the same weight.

Constant	AA & DD sum	N* sum	Surf. RMSD
-1.57	-1.92	-0.00300	1.93

4.6.4 Results

The scoring function from the previous section (Table 11) was used to select the best alignment for each pair of binding sites in the three test datasets. The scores were normalized as previously using the scores of the query sites versus the 140 diverse structures. The best scoring alignment per pair of binding sites is refined using ICP on the site surfaces and site hydrogen bond caps. Based on data that is not presented here, it was determined that each hydrogen bond point correspondence should count as four site molecular surface patch correspondences for the purposes of ICP.

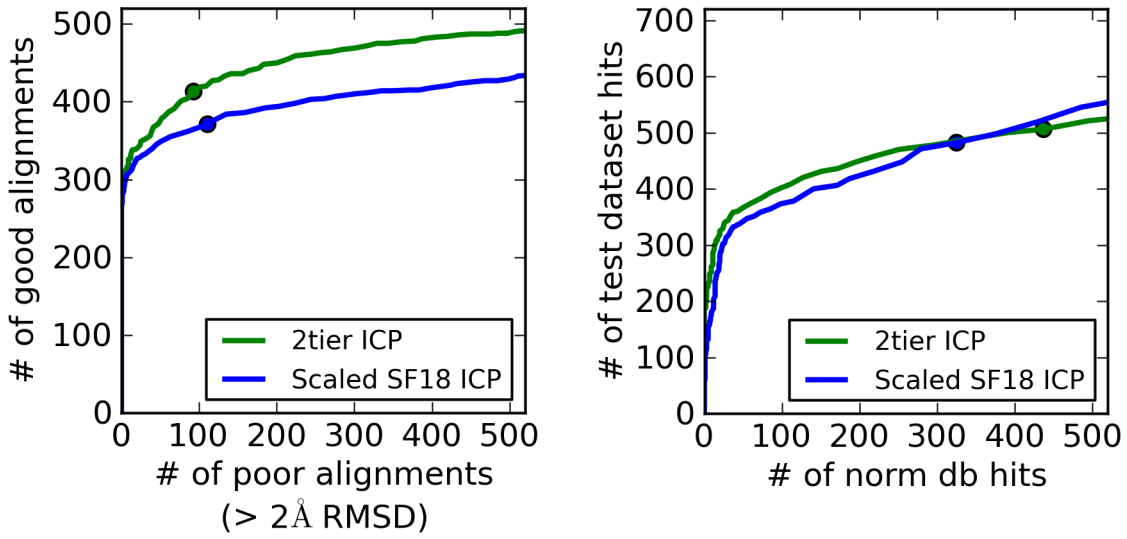


Figure 36: ROC-like curves comparing the scoring function performance of the two-tiered scoring and scaled terms for hydrogen bond caps and surface complementarity. On the left, the plotted data is the normalized score and site RMSD of alignment for the best scoring alignment for each pair of binding sites in the three test datasets. On the right, the data is the scores of the best scoring alignments for within test dataset pairs of sites and for test query sites versus the 140 diverse structures. If one considers the performance of the scoring function using the hydrogen bond caps with that of SF13 (Figure 25), the performance is very similar.

4.6.5 Discussion

Overall, the addition of hydrogen bond caps and using the surface complementarity of the binding sites did not significantly alter the results when compared with using hydrogen bond points and surface complementarity. One remark is that maximal overlap of complementary hydrogen bond caps need not be required for proteins from different families to bind the same ligand. In addition, the presented results for hydrogen bond caps does not address the issue of modeling waters in the binding sites as water molecules were ignored. Thus, an elegant model that seems to be more representative of binding site features need not work better in practice than more simple models if the more elegant model does not more accurately model the underlying mechanism.

Although the results on the test datasets do not show a great improvement (between

SF13 and scaled terms including hydrogen bond caps and surface complementarity), refinement of alignments using caps and surface does increase both terms. On the other hand, ICP on site map points alone rarely improves alignments with respect to score or RMSD of site alignment, and in many instances makes the alignments worse (with respect to site score and RMSD of alignment). ICP on two-tiered scoring (and SF13) improves surface complementarity but generally reduces the hydrogen bond point (and caps) complementarity (Section 4.2). Therefore, at the present, a primary advantage of using hydrogen bond caps and surface complementarity is that optimizing both sets of correspondences using ICP usually increases both the surface and chemical complementarity for those pairs of sites in the test datasets that have candidate alignments with relatively low alignment error.

4.7 Remarks

Clearly, the inclusion of binding site surface complementarity is beneficial as it helps to distinguish between binding sites with similar chemical complementarity based on their shape similarity. It is also rather obvious that both the hydrogen bond caps and increased alignment sampling did not yield substantial gains in the recognition of binding sites in the test datasets. Neither did that functionality improve the discrimination between hits from test datasets and hits from a diverse set of proteins. Thus, it is likely that binding site comparisons requires a paradigm shift and/or the inclusion of more accurate or descriptive features.

It is our opinion that one must be mindful of the magnitude of the errors present in crystal structures. Ideally computational methods would be somewhat stable with respect to perturbations of the same magnitude as the measurement and model errors. Therefore, it is unlikely that very detailed models (e.g. detailed force-field models) will substantially enhance methods to compare binding sites as the crystallographic uncer-

tainty should be considered as a lower bound on the sensitivity of the models.

Chapter 5

ArtSurf: Flexible Refinement of Aligned Binding Sites

A common issue for object recognition methods is that rigid body alignments are generally insufficient to recognize flexible objects. As an example, the limbs of the human body can move large distances relative to the scale of the body. A specific example is that many of the point correspondences found by rigid matching will be incorrect when comparing a person touching his toes to a person with her arms raised over her head. If the human body is modelled as a shell (surface) over a stick figure, the joints and connectivity of the human body can be exploited as part of the matching algorithm.

One algorithm that uses known joint parameters for human joints is articulated ICP. By using articulated ICP, the shells for the limbs can be aligned subject to joint constraints [81]. The general idea of articulated ICP is:

1. Segment the objects into rigid sections
2. Find the best alignment, via ICP, for one of the rigid sections
3. Loop by selecting one joint from one of the already aligned regions and use ICP to optimize both the joint parameters (of the selected joint) and the best surface

correspondences for the surface patch that depends on that joint and is not already aligned.

In this manner, the rigid sections are aligned iteratively, but the types of joints must be known or estimated [81]. Articulated ICP might be useful in binding site comparison cases where the binding site surface can be decomposed into a relatively small number of distinctive surface patches such as those exhibited by exposed side chains.

Current advanced object recognition methods are generally problem specific since descriptive features usually depend on the posed question. In addition, to reduce the time needed to recognize an object, many problem specific assumptions and heuristics are used, and the methods are tuned to address specific questions. As an example, suppose an articulated ICP method was tuned to perform well for pose prediction or tracking of limbs. Then, directly applying such an articulated ICP method to flexible, human face recognition is likely to perform poorly since facial expressions are more nuanced than limb motions, and facial points tend to have less relative displacement than human hands or feet. However, such phenomena do not preclude applying the general framework of articulated ICP to non-rigid face recognition, as certain regions of facial skin can and do move together. Therefore, the articulated framework presented in this chapter is likely to apply to other applications, but it is tuned for the comparison of protein binding sites.

In this chapter, the goal is: "Given aligned binding sites A and B, can A undergo directed shape changes and relative positioning and orientation of chemical hot spots, subject to protein constraints, to increase the chemical and surface complementarity between site A and B?" A specific problem case is: given three aligned binding sites A, B, and C, of which A and B bind the same ligand but C cannot, after the directed local changes, is it clear that A & B become more similar but A & C and B & C do not?" The problem statement is worded carefully because:

- If two binding sites are not well aligned, the method will not perform well.

- The method in this chapter fixes (freezes) residues outside of the binding site in their crystallographically determined relative positions.
- The protein side chains in the binding site are moved in a directed manner that is not necessarily the path actually taken by the side chains in solution.
- Flexible comparison of binding sites is a new area of research and ought to be addressed with methods that are not overly complex as to not obscure general observations.

Thus, our hypothesis is: "Optimizing binding site side chain positions and orientations of site A by maximizing the local shape and chemical complementarity between sites A and B will allow for a more accurate determination of whether site A can bind the ligand bound in site B".

The questions posed for flexible protein surfaces have details that differ from human face recognition or human pose recognition. As presented in Chapter 4, the surface of a protein is represented by an envelope surrounding solvent-exposed amino acids. However, unlike the limbs used in the articulated ICP example [81], many of the amino acids in a binding site are only partially exposed. As a result, it would be very challenging or impossible to accurately determine the underlying joints (atom centers) and links (covalent bonds) based solely on the surface patch of a binding site. In addition, our goal is to match sites that can bind similar small molecules, from otherwise unrelated proteins. As a result, the atom centers and covalent bonds from pairs of aligned binding sites rarely have direct correspondences which makes flexible binding site refinement a more difficult problem than that addressed by articulated ICP [81]. Finally, when comparing binding sites, the goal is to determine whether sites may present similar shape and chemistry, but not necessarily to place atom centers and covalent bonds in a similar configuration (due to the differences of amino acids among sites).

5.1 Problem Statement for Flexible Binding Site Comparisons

At present, the problem of addressing flexibility when comparing binding site surfaces has not been presented or published by any other research group. In fact, the problem of modeling flexibility to determine correspondences between binding sites is an untouched problem of great importance. The problem of the placement and orientation of amino acid side chains has been studied extensively in homology modeling and protein-ligand docking [6, 106], but is for one protein structure or binding site. Some flexibility modeling has been done for protein-ligand interfaces, but in general the majority of protein side chains are kept rigid to reduce the total number of degrees of freedom (so that the methods do not suffer from combinatorial explosion). The methods published that address protein flexibility in protein-ligand docking [2, 25, 39, 54, 90] tend to allow some flexible side chains, but the flexibility is driven by accommodating ligand binding rather than optimizing binding site shape and chemical complementarity between two binding sites. Most of the docking tools with flexible binding sites use discrete samplings of dihedral angles (called rotamer libraries [1]) and an optimization method such as integer linear programming [6], branch and bound, or mean field optimization (self-consistent field theory) [58, 90] to choose the dihedral angles to use in the interface. However, studies have shown that side-chain orientations in binding sites often adopt non-rotameric states to accommodate ligands [6, 45, 76, 106] Thus, modeling flexibility in proteins is not new, but has been tackled in a limited way and has not been addressed for protein-ligand binding site comparisons.

The general framework used to address the flexibility of binding sites in computa-

1 Rotamers is the name given to the preferred values for dihedral angles of protein side chains. Typically, for each bond that can rotate, there are two or three peaks in the angular distribution. The rotamers usually are the mean/median of the region of the distribution near each peak and may include the values +/- one standard deviation from the mean/median values

tional methods is now presented. The central idea is somewhat similar to that of "The Directed Tweak Technique" [49]. However, in this chapter, the idea is to maximize the surface and chemical complementarity of protein side chains instead of the overlap of small molecules, and more applicable mathematical techniques are used. The problem is separated into two components: optimizing the complementarity of the two sites, and modelling realistic protein motions. The methods used to determine the surface and chemical point correspondences are those used for site alignment and similarity scoring (Chapters 3, 4). The corresponding surface and chemical points are attracted to each other, are allowed to move to optimize the correspondences, and are subject to the underlying protein constraints. As proteins are comprised of one or more chains of amino acids, the major degrees of protein freedom are the dihedral angles of the single bonds. Thus, proteins can be modelled as articulated objects with atom centers considered as joints and covalent bonds as links/limbs.

A number of simplifying assumptions are used and include:

- The atomic positions in one protein are held fixed while the atoms in the binding site of the other protein may move relative to each other based on the attraction of corresponding points.
- Protein atomic centers can be considered as joint centers with the bond coordination angles held fixed (the angles between two bonds that share an atom are held constant)
- Protein covalent bonds can be modeled as arms/links
- Covalent bond lengths are held constant (no bond extensions/contractions)
- Only the dihedral angle many change at each joint (that corresponds to a single bond rotation)
- All main chain angles are held constant

Although the general method presented in this chapter can accommodate all the degrees of freedom that are held fixed, the constraints were chosen so that the prototype method was of reasonable scope and addressed the main protein degrees of freedom in binding sites.

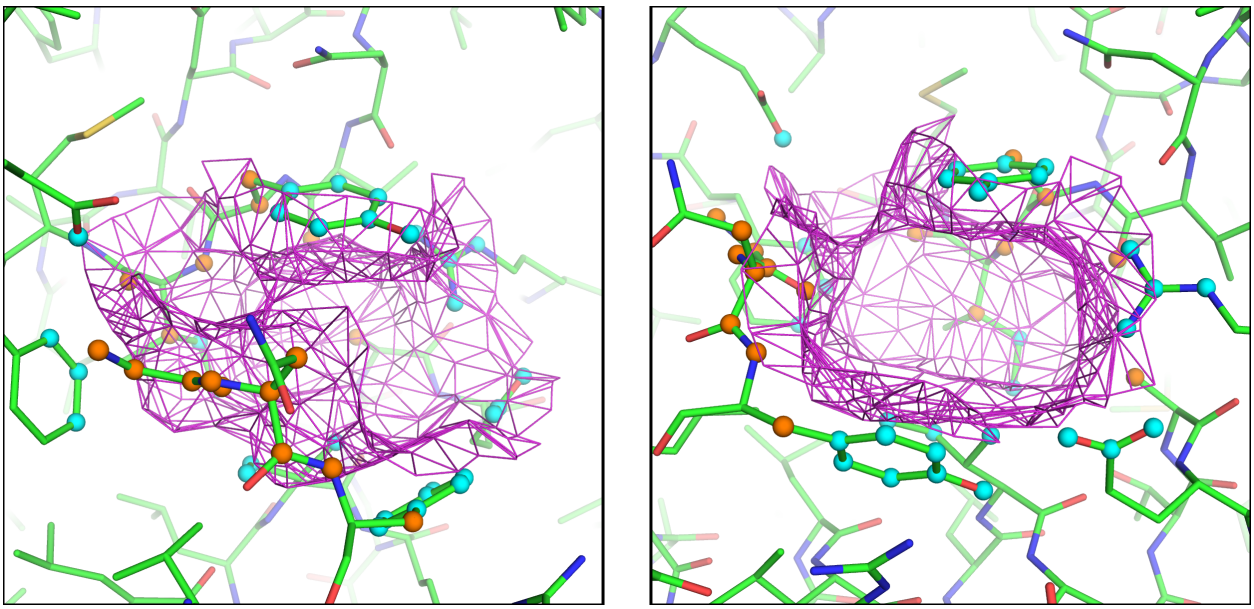


Figure 37: Molecular surface and atoms of the adenine binding site in the α-momorcharin structure (PDB: 1AHA). The magenta lines represent the edges of the molecular surface. The tubes are the bonds between protein atoms. The spheres denote those protein atoms that form the surface of the adenine binding site. The orange spheres are atoms that are held fixed relative to each other, and the cyan spheres denote the atoms which may move relative to their neighbors (subject to the presented constraints– including no bond stretching, etc.). The fixed atoms are invariant with respect to side chain rotations. If one assumes the vertical axis lies on the page, from bottom to top, the view on the right the result of rotating the protein about 90 degrees about the vertical axis from the view on the left. The key point here is some of the side chains form a large portion of the pocket's surface, while others contribute a relatively smaller amount.

5.2 Inverse Kinematics

The modeling approach that seeks to adjust the joints of an articulated object so that an end effector can reach objective points is called inverse kinematics (IK).

Definition. An **end effector** is a point of an articulated object that is to be moved to a goal (e.g. hand, foot, etc.).

Inverse kinematics is a well studied problem with applications in areas such as robotics and character animation. In IK settings, the modeled degrees of freedom are joints at positions in space (i.e. points), the objective points are called goals, and the points on the model to move to the goals are called end-effectors. Prior to this dissertation, the IK problem has had some applications in protein science, most notably, the protein loop closure problem [57].

Solutions to the inverse kinematics (IK) problem may be better understood by first considering the forward kinematics problem.

Definition. The **forwards kinematics** problem is given a particular set of joint angles for an articulated arm, determine the position of the end effector.

It is relatively easy to see that the forward kinematics problem can be solved by applying the corresponding coordinate transformation, at each joint, starting at the base (root) of the arm. However, the IK problem is: given a desired position of an end effector, what, if any, are the joint angles to reach that position? Conceptually, one could start by placing the end effector at the desired position and perform the inverse of the transformation used in the forward kinematics method, but the joint angles are unknown. Thus, one must solve for a set of joint angles, but this is a nonlinear optimization problem.

One method to solve for (estimate) the joint angles in the IK problem is using a first-order numerical optimization method [8, 102]. The key idea is to use linear approximations to the forward kinematics problem, since it is easy to compute, and invert the computation. Suppose that the rotational degrees of freedom of the joints are given by the vector $\mathbf{q} = (q_0, q_1, \cdots, q_m)$ and the position of the end effector by the vector $\mathbf{x} = (x_0, x_1, x_2)$. Then, the forward kinematics problem is: given a change in joint angles $\mathbf{q}_0 + \Delta\mathbf{q}$, what is the change in position of the end effectors $\mathbf{x}_0 + \Delta\mathbf{x}$? As stated previously, we can solve this problem by applying m coordinate transforms. These transforms can be represented as a function $f(\mathbf{q}_0 + \Delta\mathbf{q}) = \mathbf{x}_0\Delta\mathbf{x}$.

However, our goal is to find the joint angles to move protein atoms or robot arms to the desired location. Thus, we know where we want to move the end effector ($\mathbf{x}_0 + \Delta\mathbf{x}$), but do not know the change in joint angles $\mathbf{q}_0 + \Delta\mathbf{q}$ that will result in such a move. In general, the problem of finding an inverse to the forward kinematics problem can be over or under-determined (depending on the system of equations). Now, assume that one can determine the inverse of $f()$ ($f^{-1}()$). By applying $f^{-1}()$ to both sides of the forward kinematics equation, we get $\mathbf{q}_0 + \Delta\mathbf{q} = f^{-1}(\mathbf{x}_0 + \Delta\mathbf{x})$. As mentioned earlier, $f^{-1}()$ is nonlinear and difficult to compute. A commonly used numerical technique is to compute a linear approximation of $f^{-1}()$ which is basically a first order multidimensional Taylor series. The idea is to use the Jacobian $\mathrm{J} = [\frac{\partial x_i}{\partial q_j}]$ to form a linear approximation [2], $\mathrm{J}\Delta\mathbf{q} \approx \Delta\mathbf{x}$, to the forward kinematics equation ($f(\mathbf{q}) = \mathbf{x}$) at $\Delta\mathbf{x} = (0, 0, 0)$. Then, given small $\Delta\mathbf{x}$, the linear approximation will agree well with the true value [3]. By using an iterative process, one can keep the error of the linear approximations small enough, but still approach the desired solution. An iterative method is generally repeated until it has converged ($\Delta\mathbf{x}$ is minimized), or a maximum number of iterations has been reached.

2 In mathematical terms, J gives the instantaneous rate of change (i.e. partial derivative) of each end effector with respect to each joint angle

3 Of course, this statement relies on a well-behaved objective function

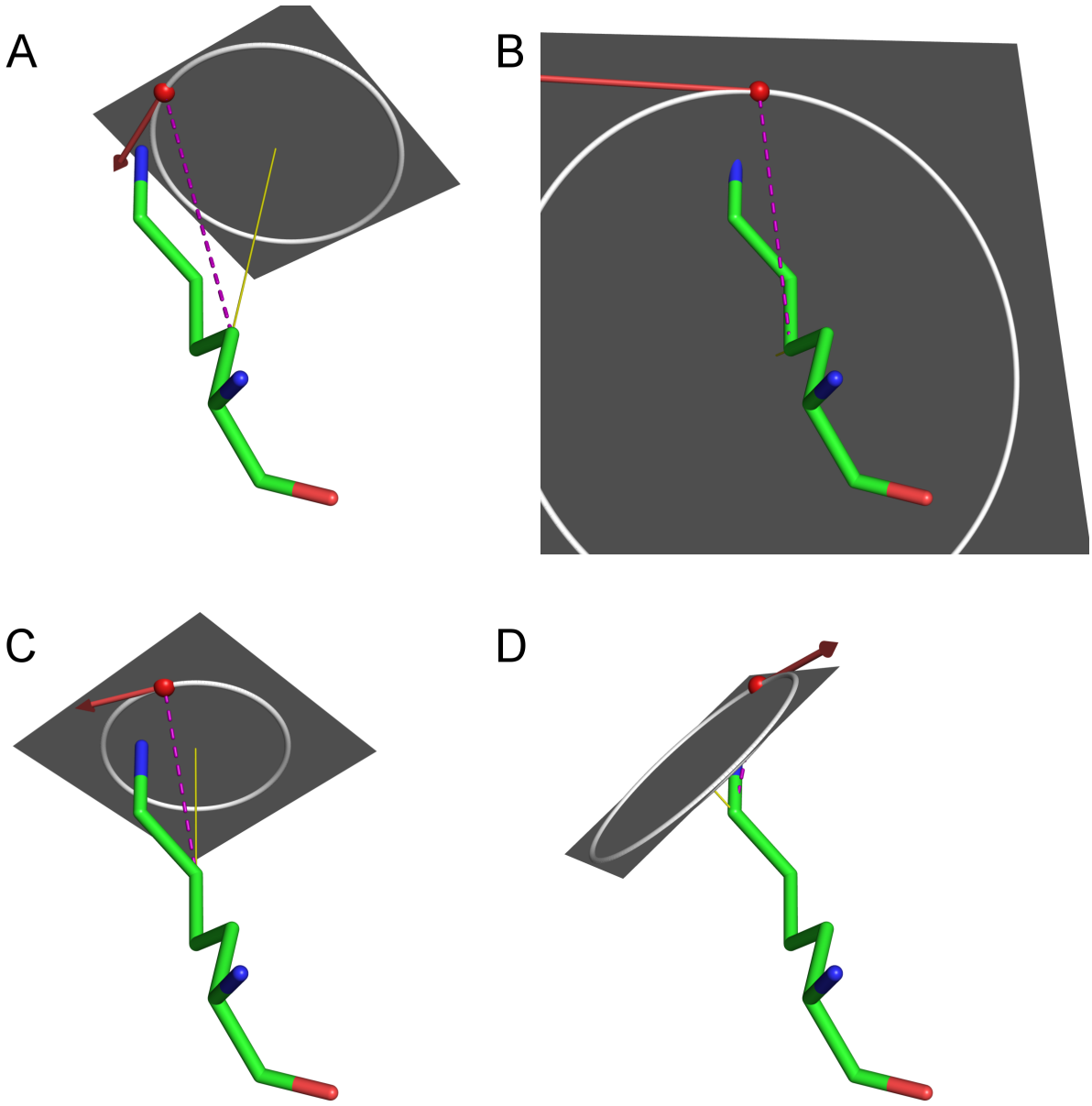


Figure 38: Example of effects of dihedral rotations on one chemical point. The tubes represent the bonds of a lysine amino acid side chain. The red point is a hydrogen bond acceptor point that corresponds to the terminal nitrogen atom. The yellow lines are axes of rotation. The white circles are the valid positions of the point with respect to rotation about the corresponding axis (with the other axes held fixed). The magenta dashed line sweeps out a nape of a truncated cone about the axis of rotation. Each red vector lies in the plane of its corresponding circle and is tangent to that circle at the red point. Panels A,B,C, and D represent a linear approximation to the rotation of the red point about the CA-CB, CB-CG, CG-CD, and CD-CE bonds, respectively (i.e. each vector is a graphical representation of the three corresponding values in the Jacobian J).

Because we know $\Delta\mathbf{x}$ and seek $\Delta\mathbf{q}$, we multiply both sides by J^{-1} to get $\Delta\mathbf{q} = \mathrm{J}^{-1}\Delta\mathbf{x}$. However, in most cases the Jacobian is not a square matrix and J^{-1} does not exist. The solution is to use a pseudo inverse [82] of the Jacobian, denoted as $\mathrm{J}^{\dagger}$, and the equation becomes $\Delta\mathbf{q} = \mathrm{J}^{\dagger}\Delta\mathbf{x}$. Since this is a linear approximation to a nonlinear equation, the system can only be adjusted by a small step in $\mathbf{q}$ towards the end point ($\mathbf{x}$) so that the value of $\Delta\mathbf{x}$ is sufficiently accurate. The Jacobian must then be computed for the new positions and joint angles and the system moved another small step towards the end point. Solving the IK problem using the pseudo inverse of the Jacobian provides a sound mathematical basis for the problem, and it allows for IK solvers to be improved by applying methods from numerical analysis to enhance the convergence rate and place reasonable bounds on the size of the changes in joint angles [8].

Solving the inverse kinematics problem using a linear solver is straightforward to implement, is conceptually clear, has strong mathematical foundations, and it is the method of choice based on experience with implementations [8, 20]. Use of the inverse Jacobian allows for larger time steps, tends to have more natural motions (all joints can move a small amount each iteration rather than adjusting one joint each iteration in which case a few joints may undergo large changes in angles while the other stay relatively constant), and suffers from fewer numerical problems [8, 20].

5.3 Optimization

Even the most straightforward IK problem requires one to know which end effectors to move and to which goals. One solution is to have a user select the end effectors and goals. However, for large scale processes, user interaction is not feasible. Another solution is to adjust the joint angles by optimizing the matching of joint-dependent features between the two objects. The problem is then formulated as an optimization problem that takes the form of a objective function and one or more constraints. The objective function is

generally problem and feature dependent and, in the context of IK, depends on the joint angles. Constraints may be added for preferred distributions of joint angles, feature correspondences, and etc. Such constraints may be incorporated into the objective function, and our implementation uses this approach. The gradient of the objective function gives the direction of the greatest increase; depending on whether the goal is to maximize or minimize the objective, one moves the system in the direction of the gradient or negative gradient, respectively. Therefore, the problem of determining which moves to make (in the inverse kinematics setting) can be based on this complementary optimization problem.

In the general case of comparing binding sites from distinct protein folds, there are no known rules to establish correspondences or the relative significance of the correspondences. The correspondences in SitesBase are between nearby atomic centers for atoms (in the binding sites) with the same element [37]. Methods such as SiteEngine [92] and Cavbase [89], pair up nearby atomic centers for atoms and pseudo atoms that are chemically important and have the same chemical type (hydrogen bond donors, hydrogen bond acceptors, π centers, and aliphatic points). Still others, including SimSite3D and SuMo [53], construct correspondences between computed chemical points that are nearby and share feature labels (e.g. hydrogen-bond acceptor). Based on published results and the tests in this dissertation, no one method has been shown to be clearly superior to any other (Chapter 3).

Therefore, given that surface and chemistry are important for site similarity searches (Chapter 4), the SimSite3D surface and chemical correspondences were selected as the features to optimize. In the interest of keeping the problem clear, the main goal (objective) is to minimize the ℓ_2 distance between the SimSite3D surface and chemical point correspondences. Protein stereochemical constraints can be modeled by adding penalty terms to the objective function. In particular, protein atoms should not have significant Van der Waals overlap, and one may desire to have the final joint angle configuration

(i.e. final protein conformation) be energetically favorable. Such an objective function is one example of an optimization method that can be used to automatically direct the movement of end effectors to reach given goals.

5.4 Protein Motions

The previous sections covered how to move protein atoms and where to move corresponding points to optimize a given objective function, but do not directly provide a connection between the two ideas. Therefore, we require an association between the molecular surface vertices and chemical points and their corresponding protein atoms. In SimSite3D, each vertex is modeled as being rigidly attached to its closest protein atom or bond, and each chemical point is assumed to be rigidly attached to its corresponding protein atom. This association is made so that each query vertex is considered as an end point in the IK formulation. The free dihedral angles in the amino acid side chains that contribute to the surface or chemical points form the set of joint angles in the IK formulation.

Using this association and the stated joint constraints, each vertex and chemical point has 3 columns in the Jacobian J (one for each dimension in $\mathbb{R}^3$), and each joint has one degree of freedom and has a corresponding row in J. The gradient of the objective function gives the direction of the greatest increase. The objective function is reduced by moving in the direction of the negative gradient. The change in the joint angles is found by matrix multiplication between $J^{\dagger}$ and the negative gradient. Since the approximations are linear, the result is only valid in a small neighborhood of the current values of the joint angles (joint configuration space). Therefore, only small moves are made in the joint configuration space. This process is repeated until the maximum number of iterations is reached or the method has converged.

	E_x	E_y	E_z
Joint angle for Lys CA-CB bond $(\chi_{i,1})$	$T_{A,x}$	$T_{A,y}$	$T_{A,z}$
Joint angle for Lys CB-CG bond $(\chi_{i,2})$	$T_{B,x}$	$T_{B,y}$	$T_{B,z}$
Joint angle for Lys CG-CD bond $(\chi_{i,3})$	$T_{C,x}$	$T_{C,y}$	$T_{C,z}$
Joint angle for Lys CD-CE bond $(\chi_{i,4})$	$T_{D,x}$	$T_{D,y}$	$T_{D,z}$

Table 12: Example of the part of a Jacobian block corresponding to an end effector (site map point) E and a lysine side chain (Figure 38). The columns E_x, E_y, and E_z correspond to the x, y, and z coordinates of the site point. The rows correspond to the joint angles (dihedral angles) of the Lys residue with residue number i. The values of row 1,2,3, and 4 are exactly the components of the tangent vector computed for panels A, B, C, and D, respectively, (Figure 38).

5.5 Computational Method

The presented outline of the method provides a conceptual overview of ArtSurf, but does not provide the implementation details necessary to reproduce results. The concepts are presented and implemented in a modular manner so that one concept may be modified without affecting the other concepts/modules. In this section, the data structures, numerical methods, and implementation details are presented and explained.

The protein side chain atomic positions and associated points (surface vertices and chemical points) require straightforward and efficient bookkeeping to keep the method understandable and computationally efficient. For each side chains, except isoleucine, there is a single chain of zero or more joints with rigid group of one or more heavy atoms at the end of the chain. Given the PDB naming of side chain atoms and that the joint chains are linear, it is clear which joint angles affect which side chain atoms. Given this fact and that the protein main chain is kept rigid, each mobile side chain and its associated points and atoms form a block in the Jacobian and all entries outside any block are zero. This means that one needs to store only the blocks, and block multiplication is used to help reduce the computational time.

The surface vertices and chemical points are assigned to move with their corresponding atom (one could think of this assignment as a rigid pseudobond between each point

and its corresponding atom).

- Each molecular surface vertex is assigned to the closest atom in the protein.
- If the closest atom is a joint, then the vertex is checked to see whether it lies above or below the plane defined by the atom's axis of rotation (the plane normal) and using the center of the atom as the point on the plane.
- If the vertex is below the plane, it is assumed that a rotation around that particular axis would not significantly affect the molecular surface of the protein at that point [4].
- Each chemical point is assigned to the atom from which the chemical point arose.

Based on this assignment, forces on the points can change the joint angles, and conversely, changes in joint angles will propagate to the points.

The numerical part of the implementation consists of solving the optimization and the IK problems. What remains, to complete the IK problem as presented, is to compute the Jacobian J and its pseudo inverse $J^{\dagger}$. Note that the presented method to solve the IK problem relies on a linear system of equations (as does least squares regression). The solution space of a linear system of equations can be problematic as it may contain a number of singularities or unstable points. Singularities are locations in space characterized by small changes in the input that produce relatively large changes in the computed solutions [30]. In the IK setting, this occurs when J is almost row rank deficient and exhibits itself as small changes in positions yield relatively large changes in joint angles [20]. A common solution is to use dampened least squares (regularization) to avoid singularities [8, 20]. That is, compute the pseudo inverse as $J^{\dagger} = (JJ^t + \lambda I)^{-1}J^t$, where I is the identity matrix of the same size as JJ^t, and λ is a small, positive constant. The inverse of the regularized square matrix $(JJ^t + \lambda I)$ is computed via LAPACK [5] using the Cholesky

4 The bond parallel to the axis of rotation is not rotating therefore surface points associated with the bond are fixed irrespective of changes in the dihedral angle

decomposition method. The blocks of J and the inverse of the square matrix are used to compute the pseudo inverse $J^{\dagger}$.

The objective function is to minimize the squared distance between the corresponding points which may be the surface points and/or the chemical points. Let V be the set of M vertices of the query surface, and let V' be the set of closest points on the dataset surface. Then, the vertices in V are variables and the points in V' are held constant (for the current iteration). The gradient of half of the squared difference in positions of the corresponding points is a vector $G = [v_{i,j} - v'_{i,j}]$ where $0 \leq i < M$ and $0 \leq j < 3$. A similar construct is used for the gradient H of half of the squared difference in the positions of the corresponding chemical points. One or both of the gradients are used to compute the change in position (i.e. $\Delta\mathbf{x}$). Angular constraints are imposed once the change in joint angles is computed from $J^{\dagger}G$ and/or $J^{\dagger}H$.

There are two angular constraints in the implementation: severe overlap of atoms within the same protein is not allowed and the maximum rotation of any joint is restricted to 5 degrees per iteration. Overlap of any two protein atoms within the same protein structure file are limited to five percent of the sum of the atoms' Van der Waals radii. There are two types of exceptions: atoms that can participate in a hydrogen bond are allowed to have a minimum distance of 2.5 Å ; those pairs of atoms that have greater initial overlap, as given in the original structure, are left undisturbed or have their overlap reduced if such a reduction helps to minimize the error between corresponding surface or chemical points. Overlap is handled by fixing all joints that could move any overlapping atoms.

Once the final changes in joint angles, $\Delta\mathbf{q}$, are computed, the changes much be applied to those objects, in the query site, that depend on the joint angles. The objects include: the hydrogen bond points and caps, the site surface vertices, and the side chain atoms. The joint angles, for a given side chain, are applied starting with the joint closest to the α-carbon and moving along the joint chain for each joint (e.g. for Lys the order is $\Delta\chi_1, \Delta\chi_2, \Delta\chi_3, \Delta\chi_4$).

In the implemented method, the goal is to minimize the ℓ_2 distance between corresponding molecular surface points and complementary hydrogen bond cap points. The gradient of the goal (objective function) gives the directions (vectors) to move the points to optimize the correspondences. Including the inverse kinematics representation of the protein causes the motions of the points to respect the protein's constraints by requiring that all moves be accomplished only through the allowed degrees of freedom (i.e. changes in joint/dihedral angles). Because the method is general, any reasonable objective function can be used provided that its derivative:

- is reasonably well behaved
- can be evaluate/estimated
- can be related directly or through the chain rule to changes in the joint angles.

Given the presented methods and our implementation of them, some preliminary results are now given.

5.6 Results

The preliminary results have been encouraging. However better analysis likely requires several known examples of non-homologous proteins that are known to bind the same ligand, but for which, the crystal structures differ somewhat due to binding site conformational changes Such a dataset would help to address whether the flexibility method and implementation is progressing in a helpful direction. To gauge the functionality of the method we first consider two datasets for which the protein backbone is in approximately the same conformation, near the binding site, for all proteins within each dataset. The assumption is that if one has two conformations of a binding site from the same protein such that the backbone atom positions are very similar then the shape and chemistry differences are primarily due to relative differences of the poses of the atoms in the side

chains. To this end, the effects of ArtSurf are tested on: five H. sapiens thrombin exo sites with different inhibitors bound, and ten Y. pestis HPPK pterin binding sites from a molecular dynamics trajectory.

Next, the results for a set of molecular dynamics (MD) snapshots with increasing main-chain binding site RMSD are presented to illustrate the combination of main chain motion and the refinement of flexible side chains. This set and the previously mentioned set of MD snapshots (protein coordinate files) are from MD trajectories provided by Su and Cukier [96]. These MD simulations show the pterin binding site of Y. pestis 6-hydroxymethyl-7,8-dihydropterin pyrophosphokinase (HPPK) as it undergoes low-energy conformational changes over time. Applying ArtSurf to selected snapshots will provide an example of side-chain refinement performance in a realistic case of sites with the same sequence that have undergone distinct main-chain and side-chain conformational changes.

5.6.1 H. sapiens thrombin exo sites

The following H. sapiens thrombin exo site binding sites were selected based on their diversity of inhibitors' 3D structure (shape):

- ANS-Arg-2EP-KTH, a thiazole containing inhibitor, (PDB: 1A4W)
- aeruginosine298-a (PDB: 1A2C),
- IH2, a non-electrophilic inhibitor with a cyclohexyl moiety at P1 (PDB: 1C4V)
- T87, a dual specific thrombin and factor XA inhibitor (PDB: 1G30)
- T15, an N-acetamidoimidazole with novel groups in P1 (PDB: 3C1K)

Thrombin is a relatively rigid protein that is formed by two distinct peptides. Therefore, SSM [60] was used to align the structures to 1TMB based on the longer peptide chain.

The rigidity of thrombin can be seen by the low pairwise RMSD values for the main chain atoms within 12.0 Å of the exo site (ignoring two small flexible loops) (Figure 39).

1a4w ANS-Arg-2EP-KTH	0.00	0.38	0.39	0.37	0.26
1a2c ENO-Leu-RPH-OAR	0.38	0.00	0.41	0.51	0.36
1c4v IH2	0.39	0.41	0.00	0.43	0.38
1g30 T87	0.37	0.51	0.43	0.00	0.39
3c1k T15	0.26	0.36	0.38	0.39	0.00

Figure 39: A distance matrix of the main-chain, pairwise, RMSD for five H. sapiens thrombin structures. The RMSD is computed with respect the residues within 12.0 Å of the exo site (ignoring the small flexible loops). Notice that the RMSD is generally less than 0.5 Å . It is easy to see that thrombin is relatively rigid as each RMSD is over 400+ atomic positions and with respect to each structure being aligned to 1TMB (i.e not necessarily the best pairwise alignments).

SimSite3D with ArtSurf was used to flex each query site so that the surface and chemical complementarity was increased between the query site and each dataset site. The dataset site for each structure was defined using the union of the volume of the inhibitors from all structures. The volume of each query site was determined by the corresponding structure and volume of its bound inhibitor. To separate the effects of ArtSurf from the sampling issues, the starting alignment for ArtSurf, for each pair of sites, was the alignment which minimized the main chain RMSD of the binding site. Since the terms of the objective function are also terms in the scoring function, it is not surprising that ArtSurf improves the score for each pair of binding sites (left matrix in Figure 42. The changes in side chain RMSD and score (Figure 42) are relatively small, such that, the changes in RMSD are of similar magnitude to crystallographic errors.

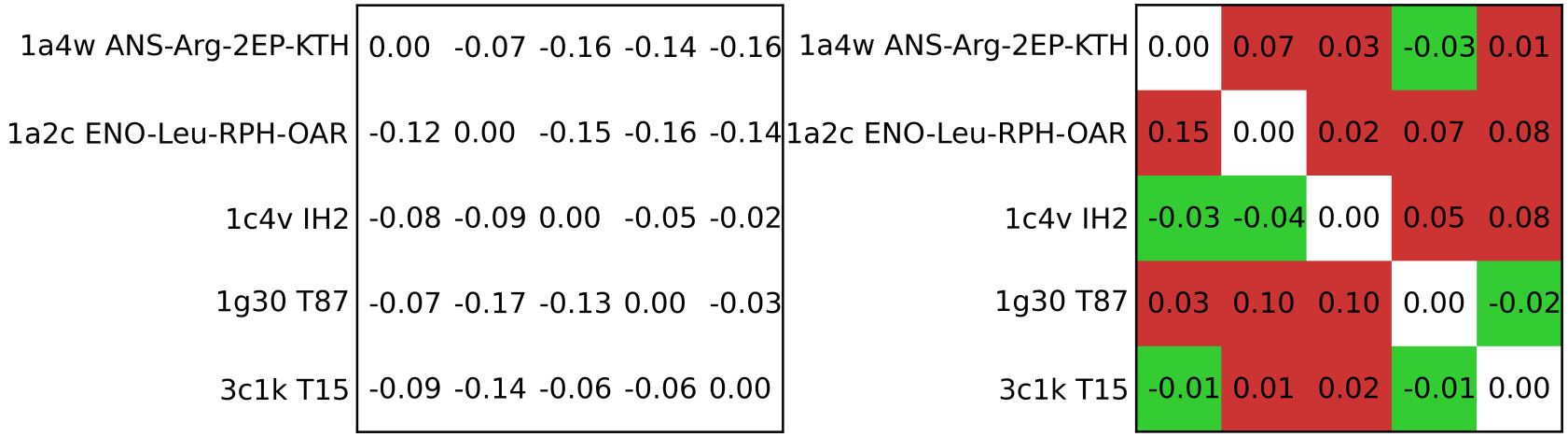


Figure 40: SimSite3D ArtSurf results for five H. sapiens thrombin exo sites with distinct inhibitors bound. Each row corresponds to a query site and each column to a dataset site. The matrix on the left shows the improvement in the site score before and after ArtSurf (the reference score is computed after aligning the sites and applying ICP but before ArtSurf). Note that a more negative score is more favorable. For the matrix on the right, each cell is the change in the RMSD (before and after ArtSurf) of the side chain atoms of those residues that ArtSurf could move. The RMSD is computed between the query site and the dataset site. The cells are colored green or red if the side chain RMSD decreased or increased, respectively, after using ArtSurf.

5.6.2 Y. pestis HPPK pterin binding sites

The starting set of molecular dynamics snapshots for the Yp HPPK pterin binding site contains 2999 snapshots. These snapshots correspond to one protein coordinate file for each picosecond of the molecular dynamics simulation. The residues near the binding site were selected using molecular graphics, and the residue numbers (in PDB 2qx0) are: 43-46, 54-56, 96, 98, 122-125. The upper triangular pairwise, main-chain RMSD matrix (distance matrix) was computed for each pair of snapshots and with respect to the binding site residues. The snapshots were clustered by a hierarchical method using average link clustering and the distance matrix. The ten snapshots for this dataset were selected by considering all clusters in the hierarchy that had exactly ten snapshots and taking the cluster with the minimum average binding site RMSD (with respect to the main-chain atoms of thirteen binding site residues).

1741ps	.00	.40	.44	.35	.36	.38	.37	.41	.46	.51
1468ps	.40	.00	.28	.34	.27	.35	.36	.39	.35	.36
1536ps	.44	.28	.00	.37	.32	.35	.38	.37	.38	.34
2850ps	.35	.34	.37	.00	.32	.38	.33	.36	.37	.41
1602ps	.36	.27	.32	.32	.00	.35	.34	.40	.36	.35
1539ps	.38	.35	.35	.38	.35	.00	.36	.29	.35	.39
1600ps	.37	.36	.38	.33	.34	.36	.00	.37	.38	.44
1490ps	.41	.39	.37	.36	.40	.29	.37	.00	.29	.42
1542ps	.46	.35	.38	.37	.36	.35	.38	.29	.00	.44
1452ps	.51	.36	.34	.41	.35	.39	.44	.42	.44	.00

Figure 41: A distance matrix of the main-chain, pairwise, binding site RMSD for 10 snapshots from an molecular dynamics simulation of Yp HPPK. The sites were aligned pairwise using a least squared fit of the N, CA, C, O atoms of the 13 binding site residues. The RMSD (LSE error) of each fit is recorded in this matrix (the unit is Å). Notice that the RMSD is generally less than 0.5 Å .

In this test, the same protein is used for each site, but the relative poses of the side chain atoms differ between the snapshots. The method of applying ArtSurf was the same as used to compute the previous set of results. Once again it can be seen that ArtSurf always decreases the site score (a more negative score is more favorable), and has an almost negligible effect on the binding site side chain RMSD.

1741ps	.00	.16	-.01	-.03	.18	.02	.00	.16	.04	.18
1468ps	.20	.00	.13	.08	.03	-.00	.12	-.15	.05	.18
1536ps	.13	.18	.00	.13	.25	-.01	.17	.42	.05	.12
2850ps	.23	.27	.13	.00	.31	.32	.25	-.02	.27	.10
1602ps	.08	.08	.22	.28	.00	.14	.28	.03	.13	.15
1539ps	.06	-.01	.11	.28	.02	.00	.00	.07	.06	-.10
1600ps	-.11	-.04	-.02	.05	.09	-.07	.00	-.01	.03	-.23
1490ps	.11	.10	.15	.26	.08	.05	.33	.00	.10	.11
1542ps	.07	.02	.07	.08	.03	.14	.16	.13	.00	-.03
1452ps	-.08	.00	.06	.02	.04	.04	-.01	.03	.08	.00

Figure 42: SimSite3D ArtSurf results for 10 Yp HPPK MD snapshots with low main chain, binding site RMSD. Each row corresponds to a query site and each column to a dataset site. The matrix on the left shows the improvement in the site score before and after ArtSurf (the reference score is computed after aligning the sites and applying ICP but before ArtSurf). Note that a more negative score is more favorable. On the right, each cell is the change in the RMSD (before and after ArtSurf) of the side chain atoms of those residues that ArtSurf could move. The RMSD is computed between the query site and the dataset site. The cells are colored green or red if the side chain RMSD decreased or increased, respectively, after using ArtSurf.

5.6.3 Y. pestis MD Snapshots with Increasing Main-Chain Differences

The two sets of molecular dynamics snapshots for the Yp HPPK pterin binding site contains 2999 snapshots each [96]. These snapshots correspond to one protein coordinate file for each picosecond of molecular dynamics simulation. One simulation used a traditional MD method and the other simulation used a Hamiltonian replica exchange method [96]. The first snapshot of the traditional MD method was taken to be the reference coordinates. A histogram was used to partition the traditional MD snapshots into bins of 0.25 Å binding site main-chain RMSD, with respect to the reference coordinates, in the range of $[0.0, 2.0]$ Å . Any traditional MD snapshots with greater than 2.0 Å RMSD were ignored. The set of Hamiltonian replica exchange snapshots were partitioned into bins of 0.25 Å binding site main-chain RMSD in the range of $[2.0, 4.0]$ (snapshots with RMSD outside of that range were ignored). For each bin, the snapshot nearest the leading edge was selected as the representative for that bin. All bins except for the first two had at least one snapshot giving 14 snapshots plus the reference coordinates for a total of 15 coordinate files.

SimSite3D with ArtSurf was used to flex each query site so that the surface and chemical complementarity was increased between the query site and each dataset site. The PDB structure of Yp HPPK (2QX0) was aligned to the reference structure and the pterin ligand PH2 (from the aligned coordinates of 2QX0) was used to define the binding site volume for the query sites. The dataset sites were defined using a 6.0 Å radius sphere centered at the center of the pterin ring system. To separate the effects of ArtSurf from the sampling issues, the starting alignment for ArtSurf for all sites was the alignment of the rigid backbone of the protein. The results in terms of the change in score and flexible sidechain RMSD vary little from the results from the previous two test datasets. One item of note is that, for this test dataset, ArtSurf does improve scores on the test dataset more than scores between the test dataset and the 140 diverse structures in the normalization dataset (Figure 43).

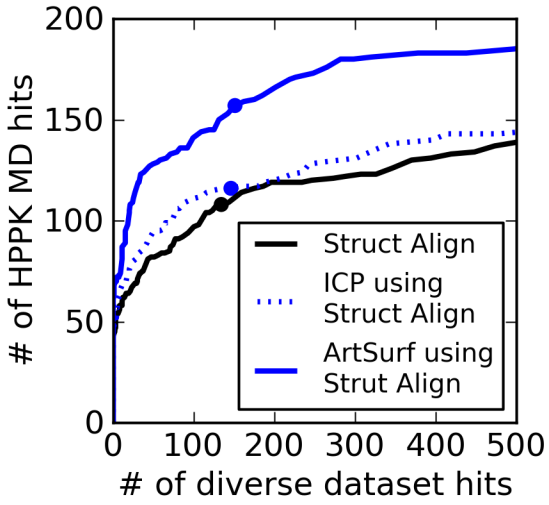


Figure 43: ROC-like curves for the ability of SimSite3D to discriminate between hits within the MD HPPK with increasing main chain RMSD test dataset and between that test dataset and the 140 diverse structures. The point on each curve denotes the location where the score threshold is 1.5 standard deviations better than the mean score (on the 140 diverse structures). The initial alignment is given by the backbone (core) alignment of the coordinates. The scoring of the initial alignment is given by the black curve. After applying ICP to the initial alignments, the results are the dashed blue curve. Application of ArtSurf yields the solid blue curve. Notice that, on this dataset, the use of ArtSurf allows for about 40 more hits (out of a maximum of 225) from the test dataset with little increase in normalization dataset hits.

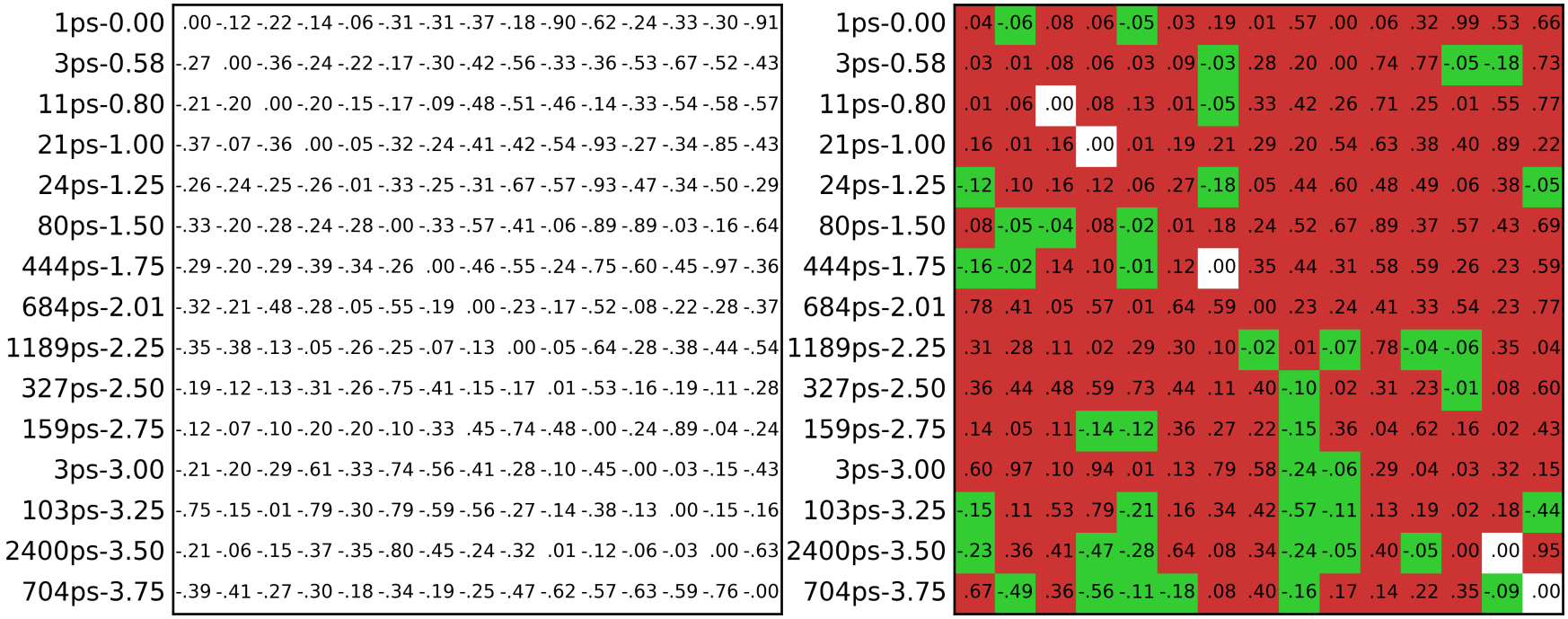


Figure 44: SimSite3D ArtSurf results for 15 Yp HPPK MD snapshots with increasing main chain, binding site RMSD with respect to the first snapshot (1ps-0.0). Each row corresponds to a query site and each column to a dataset site. The matrix on the left shows the improvement in the site score before and after ArtSurf (the reference score is computed after aligning the sites and applying ICP but before ArtSurf). Note that a more negative score is more favorable. On the right, each cell is the change in the RMSD (before and after ArtSurf) of the side chain atoms of those residues that ArtSurf could move. The RMSD is computed between the query site and the dataset site. The cells are colored green or red if the side chain RMSD decreased or increased, respectively, after using ArtSurf.

5.7 Discussion

ArtSurf has been applied to several test datasets. Based on these results the changes in atomic positioning resulting from applying ArtSurf are quite small as the changes in flexible side chain RMSD are on the order of the crystallographic error of relative side chain placement. The small changes are due to at least one of several considerations. The sites in the test datasets are from the same protein (structure and sequence). The surface meshes and chemical caps are not recomputed at any iteration, and the goal is to minimize the correspondence distance between the query surface points and the dataset surface. Therefore, if, as an example, a phenyl ring is rotated by some significant amount in one structure relative to another structure, it is unlikely that they will be planar after ArtSurf converges since the mesh surfaces of the two phenyl rings will be significantly different. However, we do not as yet know of a better method (than computing side chain atomic RMSD before and after ArtSurf) to assess the accuracy of ArtSurf. The last item is the handling of the overlap of binding site should be studied in greater detail. Currently, if two or more atoms overlap by five percent or more, their corresponding joints (those joints that affect the atoms' positions) are held fixed. The reason for this is it is not trivial to robustly and elegantly handle overlap for the cases where more than two atoms overlap and multiple joints affect the positions of the overlapping atoms.

Already at this stage, one can see that ArtSurf does help in the discrimination between within test dataset hits and hits between the test dataset query sites and the 140 diverse proteins. For this reason alone, investing additional resources in ArtSurf and like methods is likely to provide great benefits to protein-ligand structural methods. In addition, there are other protein-ligand structural methods beside binding site comparisons which can benefit from optimizing an objective function subject to protein dihedral angles.

5.8 Conclusion

We have shown the ability to implement low-energy motions in binding sites using the ArtSurf algorithm and implementation that improves the shape and chemical match between two binding sites via small rotations of dihedral bonds. At the present, the utility of ArtSurf needs to be further proved on sets of binding sites that are known to bind similar ligands and for which the given crystal structures are in different conformations. These datasets are difficult to construct since a prerequisite is to have a method (not necessarily automatic) to select protein structures from different families and in different conformations that are known to bind the same ligand such that the proteins exhibit similar chemical and shape interfaces when such a ligand is bound. Some examples achieve their similar ligand binding by using water molecules (present in one structure, absent in the other) to recognize small molecules, and an ideal set of test cases would avoid this complexity.

Once suitable examples or datasets are assembled, it is likely that ArtSurf can be further developed to address the flexible binding site comparison problem. At the present one of the issues which should be addressed is that the correspondences used to direct the changes in side-chain dihedral angles might be too local as they are capped at 1.5 Å . In some instances binding sites have a conserved, long, flexible side chain such as Lys or Glu. Because the atoms at the end of such side chains can have relative displacements much greater than 1.5 Å , consideration of other methods (than a strictly distance dependent method) to establish chemical and surface point correspondences is needed. One possibility to test is the hypothesis that side chains rooted in similar positions of the binding site correspond to one another. Then, ArtSurf could aim to optimize their match in surface chemistry. Such heuristics would circumvent the tendency of ArtSurf to match wrong side chains between two binding sites in the case where the surface points for complementary side chains are farther than 1.5 Å apart.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

The problem of comparing protein-ligand binding sites, and a computational software toolkit to address that problem was presented. Throughout the research and implementation of the method a number of discoveries were made.

It is clear that both chemical and surface complementarity are necessary for binding sites to bind ligands with similar shape and chemistry (Chapter 4). In many cases, a rigid refinement, using the surface and chemical point correspondences, of the best scoring alignment (for two binding sites) results in a more accurate alignment and a better assessment of the degree of similarity of the two sites. However, more detailed representations of the volume of space where ligand polar atoms would form hydrogen bonds with protein polar atoms did not result in improved alignment scoring or discrimination between significant test dataset hits and significant hits from a set of 140 binding sites from diverse proteins (Chapter 4). On the other hand, using the cap representation of polar volumes and the molecular surface of the binding sites did see slight improvement in alignment accuracy for ICP of rigid alignments. Based on these results and our current understanding of protein-ligand interactions there are several areas that should be explored:

- determining and modelling critical binding site water molecules
- small rotatable groups on ligands (e.g. hydroxyl groups)
- better methods of determining the chemical similarities and differences (when compared with maximizing the overlap of chemical points).

The flexible surface and chemical matching method (ArtSurf) does perform as intended, but the motions are limited due to the current method of determining surface and chemical correspondences. In fact, ArtSurf rarely makes the score worse for any alignment of any two binding sites because only those motions that improve the site score are kept. The reason is that the objective function is based on the two main terms used in the site alignment and similarity scoring function. An open problem for computational binding site comparisons is defining which chemical groups in two binding sites should correspond well for those cases where the two sites do not have significant sequence or structural similarities. This problem is also challenging for experienced structural biologists, and hypotheses such as "side chains with similar alpha carbon locations correspond to each other" will need to be tested. To our knowledge, there is not an existing break through method that performs significantly better than SimSite3D when comparing binding sites from otherwise unrelated proteins that bind the same small molecule.

After the ArtSurf algorithm and implementation has matured beyond its current ability to make small dihedral rotations to improve surface chemistry (or other labeled surface) matching, it is expected to have applications to other flexible matching problems that have coupled dihedral rotations.

6.2 Future Work

A number of important questions remain in the context of comparing protein-ligand binding sites. We have seen anecdotal evidence that careful consideration of water molecules

in binding sites would help to compare sites from otherwise unrelated proteins that bind the same small molecule. In addition, there are numerous examples of drug design pockets where one or more water molecules are known to be conserved (i.e. function as part of the protein). Finally, the modeling of water molecules is a known, challenging problem in computational protein chemistry, and when properly addressed results in models that are more reflective of experimental observations.

The comparison of binding sites as presented in this dissertation ignores a major area of existing knowledge, namely the well studied field of protein-ligand interactions. Throughout most of the research it was assumed that an advantage of binding site comparison methods is that they are not restricted to protein structures that have bound ligands. However, it is likely that for two specific questions the protein-ligand interactions could be used to improve the method.

- Are there any protein structures that have a binding site similar to my query site and have a complementary ligand bound?
- Are there any protein structures with binding sites similar to my query site and can they bind the molecule bound in my query site?

A hypothesis is that using both the protein and ligand information will better direct ArtSurf motions, and result in more accurate answers to the above two questions. In addition, the area of protein-ligand scoring functions is more established than site comparison methods, and the knowledge of protein-ligand interactions might be more helpful than was thought at the start of this research. Therefore, data fusion is expected to produce more accurate binding site similarity scores for those questions where one has both protein and ligand data.

The flexible surface matching method can be improved in several key areas. The overlap of atoms could be handled in a more graceful and/or careful manner than stopping all movements of those atoms which have significant overlap. Modelling the flexibility of

proteins' backbones may allow for more realistic binding site motions and greater flexibility for those binding sites which are affected by main chain motions. The bond networks within proteins should be modeled as being energetically favorable to form and unfavorable to break. The modeling of protein backbone flexibility subject to intra-protein hydrogen bonds could be performed similar to the methods of ROCK [65].

A major boon for designing binding site comparison methods would be the existence of at least one substantial dataset of binding sites from otherwise unrelated proteins that bound similar small molecules and that have binding sites with significantly similar shape and chemistry even when one ignores water molecules. The emphasis here is on datasets for which the shape and chemical similarities are much more pronounced than for the test datasets presented in this dissertation. Of course, more than one such dataset would be desirable so that the designed methods would have good generalization (i.e. perform well on protein folds not in the dataset). Based on the datasets presented in this dissertation, it is not clear how to refine ArtSurf for the general problem of flexible binding site comparisons. One path forward is to address (one at at time) a number of known limitations and clearly document the results.

A more specific question that appears to now be solvable is "find those protein-ligand binding sites, such that, the given query site can bind the molecules in those sites". The reason is the bound ligands provide additional information. A known problem is ligand fragments that have hydroxyl groups. An hydroxyl group can act as a hydrogen bond acceptor or donor (or both), and the hydrogen atom and lone pairs of electrons can rotate on a circle with respect to the position of the oxygen atom. In short, this means that a hydrogen bond acceptor (donor) atom from two otherwise unrelated proteins that bind the same ligand that contains a hydroxyl group (e.g. estradiol) can have the acceptor atoms at opposing locations with respect to current models for comparing protein ligand binding sites. This issue and others could be addressed by using ArtSurf to optimize the query site with respect to the dataset ligand in the place of or in conjunction with

the dataset binding site. In fact, one current unknown is how to optimize the overlay of the hydrogen bonding groups of two protein structures. In particular, for proteins that are otherwise unrelated but bind the same small molecule, maximizing the overlap of hydrogen bonding regions (points, caps, volumes, etc.) does not necessarily optimize the two structures with respect to the bound ligands. Besides addressing the binding site comparison problem, the ArtSurf framework can be readily applied to the refinement of solutions to the protein-ligand docking problem.

A particular advantage of ArtSurf, as implemented, is all of the degrees of freedom can be adjusted slightly during one timestep, and the motions are coordinated. Thus, several groups of atoms might be moved to produce a better refinement of a docking that could not be refined with methods that attempt to move one atom to its current best position each timestep In theory, the objective function can be as detailed or reductionist as one might desire. A major drawback of ArtSurf, for high throughput methods, is the cost of initialization and the need to recompute feature correspondences at each timestep. At the present, ArtSurf was not necessarily designed for computational efficiency and the run time for flexible refinement for one alignment of two binding sites (in the test datasets) is on the order of 1-10 seconds. The computation of one timestep is similar to that of ICP since the main computational burden is computing the point correspondences at each timestep. Note that spatial partitioning is used in SimSite3D and reduces the computational time by approximately 100-fold over a simple method that checks all possible point correspondences. One possible method to reduce this computational cost further is to use d2-trees [67] that use an adaptive grid to approximate the squared distance between an arbitrary point and a given surface.

APPENDICES

Appendix A

Root Mean Square Differences (RMSD)

In many applications it is desirable to compute the average error present in the alignment of two objects. In protein science one would like to gauge the quality of the superposition or alignment of structures. The most commonly used metric is the ℓ^2 norm of the differences in the positions of corresponding features from the same or similar objects (in proteins this is typically atomic positions). That is, given m point correspondences, let (x_i, y_i) for $i \in 0, 1, 2, \cdots m-1$ be the point correspondences, then $\ell^p = \left(\sum_{i=0}^{m}(x_i - y_i)^p\right)^{1/p}$ The ℓ^2 norm is used because it is easy to compute and its first derivative is smooth (i.e. it is in C^1). Apparently it is too cumbersome to call this metric "the ℓ^2 error". Thus, in some fields this metric is called the Root Mean Square Differences or RMSD. In statistical learning fields this metric is generally termed Root Mean Square Error or RMSE [42].

Appendix B

SimSite3D Documentation

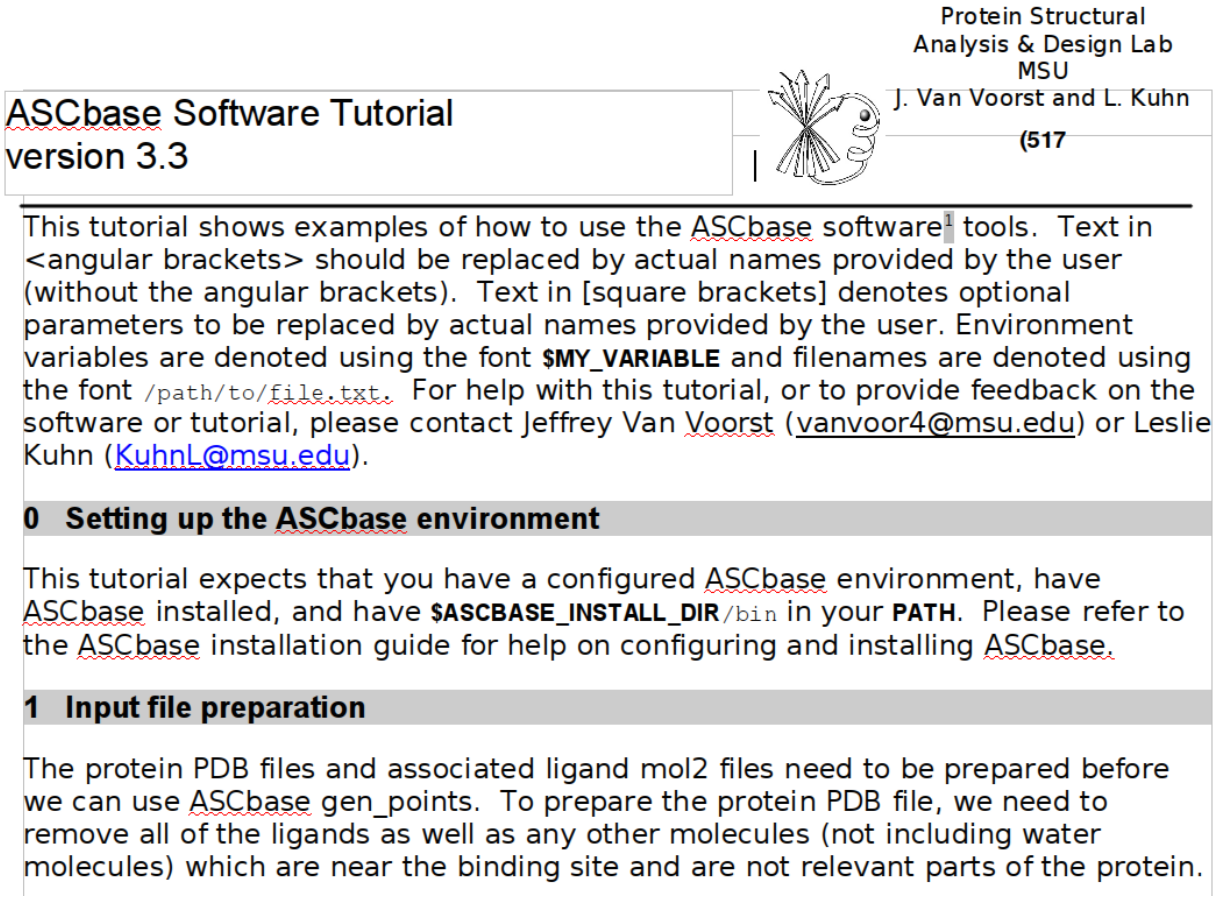
SimSite3D has been developed with users in the foreground and includes: the SimSite3D software toolkit, a short tutorial, examples of site maps and searches, an installation guide, and a user guide. Our goal is to release SimSite3D as soon as possible under the GPL-2 software license. Currently SimSite3D is approximately 50,000 lines of C++ and Python code (all of the code was written since January, 2006). There are a few requirements for the C++ code to compile in it current form: a gcc compiler, the math library, the popt library, a LAPACK library, and the scandir() function. The Python code contains a number of useful scripts that augment and extend the C++ interface. There are Python wrappers that allow access to the main C++ modules using Boost.Python.

Several versions of SimSite3D have been installed at Pfizer. SimSite3D is one of the tools, at Pfizer, which are integrated into pipeline pilot[1]. In addition, the results of a SimSite3D search can be viewed in molecular graphics both in a Pfizer proprietary molecular graphics tool and in PyMOL using our prototype PyMOL plugin. Therefore, SimSite3D has the potential to be used by many of the scientists in the drug research areas at Pfizer.

1 Pipeline pilot is a way for users to connect programs graphically and to pipe output of one program as the input of another, etc. An example of a similar program, using graphics, for dynamic systems modeling is Stella

B.1 SimSite3D tutorial

The SimSite3D tutorial covers the steps that a user would follow to create a query (or one dataset) site map. These steps assume the user already has protein-ligand structure of interest. The steps include: converting the ligand from PDB to mol2 format, generating the site map based on the ligand volume and protein shape and chemistry, and verifying that the site map was created correctly.



Protein Structural
Analysis & Design Lab
MSU
J. Van Voorst and L. Kuhn
(517

ASCbase Software Tutorial
version 3.3

This tutorial shows examples of how to use the ASCbase software[1] tools. Text in <angular brackets> should be replaced by actual names provided by the user (without the angular brackets). Text in [square brackets] denotes optional parameters to be replaced by actual names provided by the user. Environment variables are denoted using the font **$MY_VARIABLE** and filenames are denoted using the font `/path/to/file.txt`. For help with this tutorial, or to provide feedback on the software or tutorial, please contact Jeffrey Van Voorst (vanvoor4@msu.edu) or Leslie Kuhn (KuhnL@msu.edu).

0 Setting up the ASCbase environment

This tutorial expects that you have a configured ASCbase environment, have ASCbase installed, and have **$ASCBASE_INSTALL_DIR**`/bin` in your **PATH**. Please refer to the ASCbase installation guide for help on configuring and installing ASCbase.

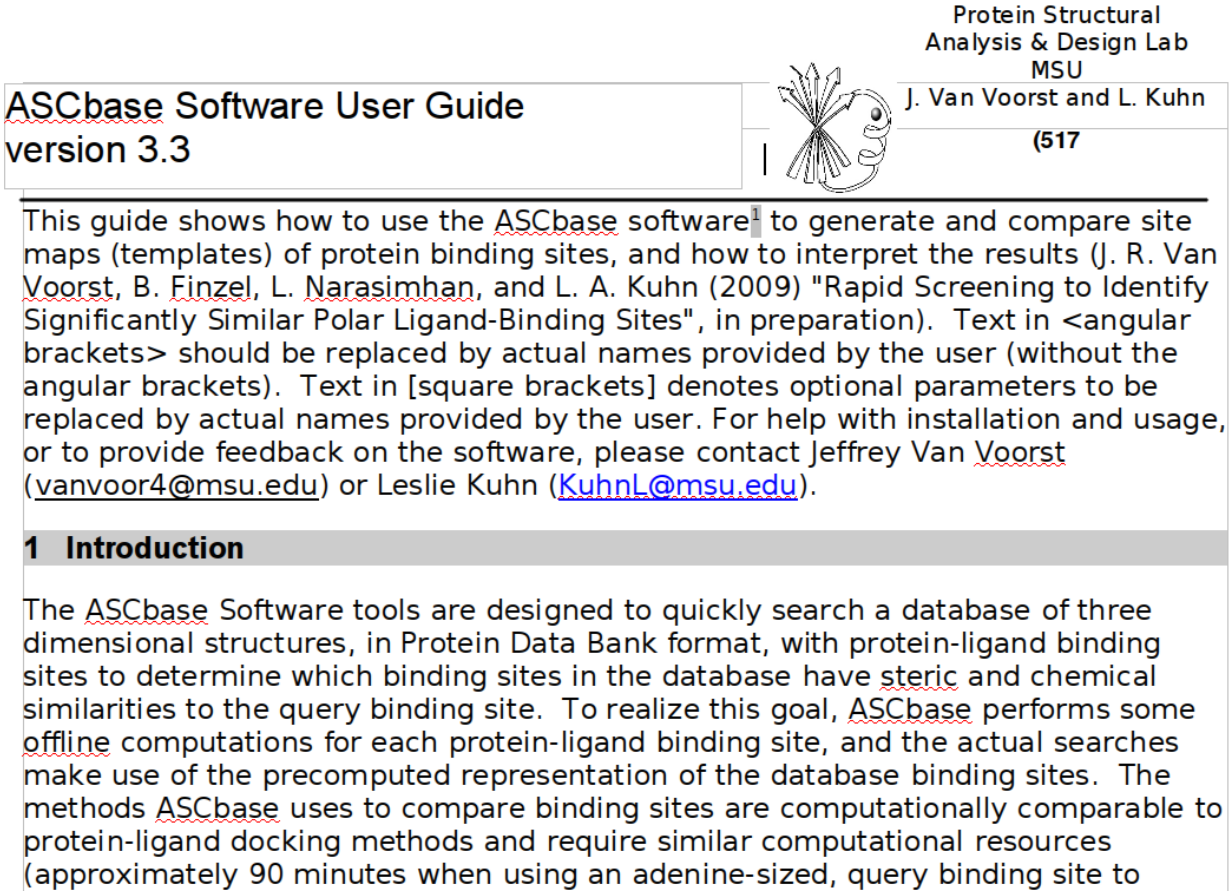
1 Input file preparation

The protein PDB files and associated ligand mol2 files need to be prepared before we can use ASCbase gen_points. To prepare the protein PDB file, we need to remove all of the ligands as well as any other molecules (not including water molecules) which are near the binding site and are not relevant parts of the protein.

Figure 45: An excerpt from the SimSite3D tutorial document. Note that this documentation was prepared for Pfizer and the name of SimSite3D within Pfizer is ASCbase.

B.2 SimSite3D User Guide

The SimSite3D user guide covers all of the options for SimSite3D with respect to creating site maps and searches.



Protein Structural
Analysis & Design Lab
MSU
J. Van Voorst and L. Kuhn
(517

ASCbase Software User Guide
version 3.3

This guide shows how to use the ASCbase software[1] to generate and compare site maps (templates) of protein binding sites, and how to interpret the results (J. R. Van Voorst, B. Finzel, L. Narasimhan, and L. A. Kuhn (2009) "Rapid Screening to Identify Significantly Similar Polar Ligand-Binding Sites", in preparation). Text in <angular brackets> should be replaced by actual names provided by the user (without the angular brackets). Text in [square brackets] denotes optional parameters to be replaced by actual names provided by the user. For help with installation and usage, or to provide feedback on the software, please contact Jeffrey Van Voorst (vanvoor4@msu.edu) or Leslie Kuhn (KuhnL@msu.edu).

1 Introduction

The ASCbase Software tools are designed to quickly search a database of three dimensional structures, in Protein Data Bank format, with protein-ligand binding sites to determine which binding sites in the database have steric and chemical similarities to the query binding site. To realize this goal, ASCbase performs some offline computations for each protein-ligand binding site, and the actual searches make use of the precomputed representation of the database binding sites. The methods ASCbase uses to compare binding sites are computationally comparable to protein-ligand docking methods and require similar computational resources (approximately 90 minutes when using an adenine-sized, query binding site to

Figure 46: The beginning of the SimSite3D user guide contains an introduction to SimSite3D. The user guide specifies the purpose and design parameters of SimSite3D.

2 Input file preprocessing

Ligand files

All ligands should be removed from the PDB files. Ligands should be saved in separate coordinate files in the Tripos mol2 format. If a PDB file contains multiple ligands, they can be handled by creating a mol2 file for each ligand and a corresponding PDB file for each binding site. As an example, if a PDB file `z123.pdb` contains the ligands ATP and ADP, one option is to create the ligand files `z123_ATP_1.mol2` and `z123_ADP_1.mol2` and create 2 identical copies of the prepared PDB file named `z123_ATP_p.pdb` and `z123_ADP_p.pdb`. (Note: one technique to save disk space is to have 1 prepared PDB file and make links to it that are named using the ASCbase file naming conventions). Please refer to the "ASCbase file naming convention" section of this user guide for the file naming conventions.

Ligands corresponding to protein-ligand binding sites

The hydrogen atoms of the ligand are ignored when determining the volume of the binding site.

Ligands corresponding to virtual screening hits

Ligands that have been docked into a target protein require partial charges and polar hydrogen atoms to enable identification of hydrogen bonds and salt bridges by the protein-ligand scoring functions, but not by search_sitemap's scoring function for comparing binding sites (site maps). Polar hydrogen atoms and charges may be assigned using a tool such as Molcharge in QuACPAC (freely available to academic users from Open Eye Software, Santa Fe, NM; http://www.eyesopen.com).

Figure 47: An excerpt from the SimSite3D user guide that describes how and why to convert ligands to a different file format (i.e. mol2) and the conditions for when partial charges are required for ligand atoms.

Site map generation command and arguments for one site map

The intended use of `gen_points` is to generate query site maps or temporary site maps. To generate a site map database, we recommend using the python script, `auto_gen_sitemaps.py`. To simplify file management, the recommended location to create query and temporary site maps is the directory from which you intend to run searches.

The methods of defining the site map volumes of interest are by a ligand volume bounding box or by a sphere. When using the bounding box option, a site map can have a large volume. This is because the bounding box is defined by the minimum and maximum coordinates of the input ligand in the *x,y,z* axis directions of the PDB file, which generally do not align well with the major and minor axes of the ligand.

To generate a site map using a ligand bounding box:

gen_points -p <XXXXXX_p.pdb> -l <XXXXXX_l.mol2> [<XXXXXX_s.csv>]

Ex: gen_points -p 1eqm_ADP_p.pdb -l 1eqm_ADP_l.mol2 [1eqm_ADP_s.csv]

In the above example, the program will first look for `1eqm_ADP_p.pdb` and `1eqm_ADP_l.mol2` in your current working directory. If a file is not found locally, gen_points will look for the protein in the **$ASCBASE_DBASE_PROTS** directory and the ligand in the **$ASCBASE_DBASE_LIGS** directory. If no output path is given, the generated site map points file, XML file, atoms file and rad file will be written to the **$ASCBASE_OUPUT_DIR** in the file named `XXXXXX_s.pdb`, `XXXXXX_s.csv`, `XXXXXX_a.pdb`, `XXXXXX_rad.pdb` respectively. Note: if given, the output file name must follow the `XXXXXX_s.csv` naming convention (please see section "ASCbase file naming convention").

Figure 48: The beginning of the section in the SimSite3D user guide on how to create a site map (query and dataset sites are created in the same manner).

4 Comparing site maps using ASCbase search_sitemaps

The ASCBase tool search_sitemaps aligns and ranks database site maps according to their shape and chemistry match to a given query site map. Search_sitemaps identifies an optimal (rigid) orientation for each database site map relative to the query site map, determines the fragment of the database ligand in the query pocket and outputs statistics about the dockings.

Docking

Site map matches are based upon testing each set of neighboring triplets of site map points in the database site map for the ability to match a triplet of site map points in the query, requiring that the two triplets (triangles) have similar side lengths (within a preset tolerance), matching chemistry labels (acceptor, donor, donor/acceptor or hydrophobic) at all three points, and have similarly oriented polar groups (the dot product between the vectors representing the favored hydrogen-bonding direction for the two matched polar site map points needs to be positive). All such triangle matches (dockings) are tested between the two site maps.

Orientation/Alignment Scoring

For a pair of query and database site maps, each saved triangle match is scored according to shape and chemistry. Pseudo code of the scoring method for a query,database site map pair is:

- For each saved triangle match:
  1. Initialize the match print to be false (zero) for each query site map point
  2. Initialize the ligand fragment binary string to be true (1) for each ligand atom
  3. Transform the database site map into the reference frame of the query site map (the transformations are computed the docking step).
  4. For each point in the query site map:

Figure 49: The beginning of the section in the SimSite3D user guide on the use of the search program.

Search results data file (*.out)

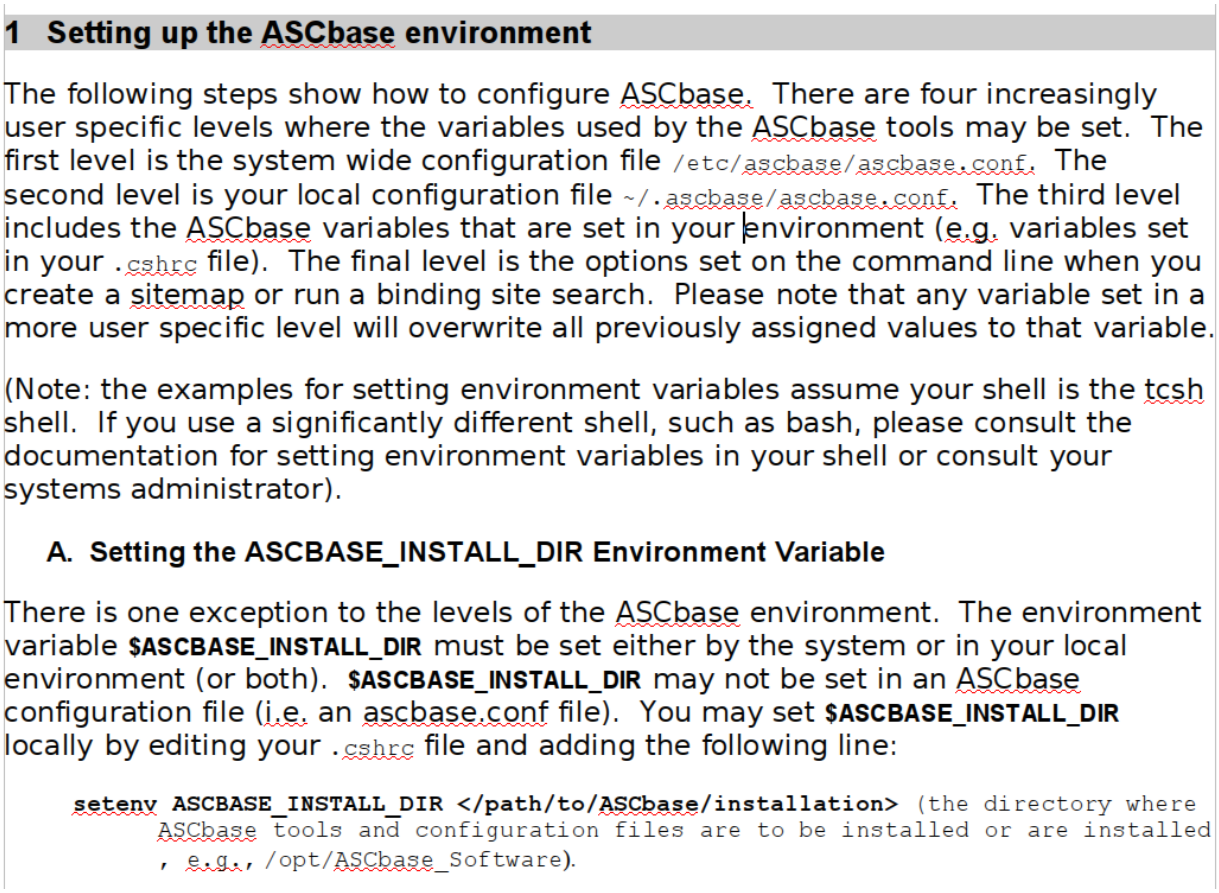
The first 20 or so lines of the results file represent a small comment header to help reproduce and/or compare results from one run to the next. Each line in the .out file that is not whitespace nor has the # or % character in the first column is a score record and contains the following information:

1) name of the ligand fragment file
2) normalized score of this orientation of the database site map with respect to the query (see Scoring section, above)
3) 3x3 rotation matrix written row-wise for operations on column vectors
4) 3x1 translation vector which is to be applied after the rotation
5) A binary string representing the match print of the hit site map to the query site map
6) A binary string denoting the database ligand atoms present in the ligand fragment
7) external score of database ligand, for this orientation, with respect to the query protein (if an external scoring method is not set or explicitly requested to be omitted, this column will also be omitted). Please refer to the section "External Protein-Ligand Scoring Functions" for an explanation of external scoring functions.

If no hits were found, the results file will not have any score records. If up to **N** scores for each database site map are requested, the number of records for each database site map will depend on how many hits for that particular site map were better than the score threshold and will be at most **N**.

Figure 50: The section in the SimSite3D user guide which gives the file format for the search results and describes what is in each field.

B.3 SimSite3D Install Guide



1 Setting up the ASCbase environment

The following steps show how to configure ASCbase. There are four increasingly user specific levels where the variables used by the ASCbase tools may be set. The first level is the system wide configuration file /etc/ascbase/ascbase.conf. The second level is your local configuration file ~/.ascbase/ascbase.conf. The third level includes the ASCbase variables that are set in your environment (e.g. variables set in your .cshrc file). The final level is the options set on the command line when you create a sitemap or run a binding site search. Please note that any variable set in a more user specific level will overwrite all previously assigned values to that variable.

(Note: the examples for setting environment variables assume your shell is the tcsh shell. If you use a significantly different shell, such as bash, please consult the documentation for setting environment variables in your shell or consult your systems administrator).

A. Setting the ASCBASE_INSTALL_DIR Environment Variable

There is one exception to the levels of the ASCbase environment. The environment variable **$ASCBASE_INSTALL_DIR** must be set either by the system or in your local environment (or both). **$ASCBASE_INSTALL_DIR** may not be set in an ASCbase configuration file (i.e. an ascbase.conf file). You may set **$ASCBASE_INSTALL_DIR** locally by editing your .cshrc file and adding the following line:

setenv ASCBASE_INSTALL_DIR </path/to/ASCbase/installation> (the directory where ASCbase tools and configuration files are to be installed or are installed , e.g., /opt/ASCbase_Software).

Figure 51: The section in the SimSite3D install guide describes how to setup one's Linux environment to run SimSite3D.

SimSite3D loads a few data files when the C++ programs are used. The Python interface must be in a user's PYTHONPATH for the Python interpreter to load the SimSite3D Python modules. These values must be in a user's environment for SimSite3D to function correctly.

2 Building and installing the ASCbase Software tools

For details on custom installations, please see the Appendix section II.

A. Installing the C++ programs (Required)

After setting the **$ASCBASE_INSTALL_DIR** environment variable to the desired installation directory, a default installation of ASCbase software can be done automatically by running the install.py script (from the ASCbase source directory – ASCbase_Software_3.3). The install script will install ASCbase in **$ASCBASE_INSTALL_DIR** and will install the tools in the directory **$ASCBASE_INSTALL_DIR**/bin:

```
./install.py  (The preceding "./" is important for the system to locate the
      file, even when it is in your working directory.)
```

NOTE: If you are planning to build ASCbase Software on a system where the default for GCC is to build 64 bit executables and you desire to build 32 bit executables, you can achieve this by editing the file configure.ac (in the ASCbase_Software_3.3 directory) and adding "-m32" to the CFLAGS line. To support building Python extension modules, ASCbase Software must be compiled using position independent code (gcc flag "-fPIC"). If you choose to remove this option, you will be unable to build the Python modules and use most of the included Python progams (of course the Python install script will not be affected).

Use gzip and tar to unpack the scoring normalization dataset files, data/diverse_sitemaps.tgz and data/diverse_ligands.tgz, into **$ASCBASE_DBASE_SITES** and **$ASCBASE_DBASE_LIGS** respectively.

Figure 52: The section in the SimSite3D install guide describes how to build the C++ programs in the SimSite3D toolkit.

The C++ programs in SimSite3D are easy to compile and install. The GNU automake tools are used to configure the parameters to the source code and makefiles based on the environment and user input.

File naming convention

The ASCbase protein, ligand and sitemap file naming scheme is:

- XXXXXX_p.pdb for proteins (e.g., `HIV-pr_K41R_V64I_0200758_LIG_1_p.pdb`)
- XXXXXX_l.mol2 for ligands (e.g., `HIV-pr_K41R_V64I_0200758_LIG_1_l.mol2`)
- Sitemaps files generated by ASCbase corresponding to the ligand binding site (e.g., `LIG_1`) in the given protein, where XXXXXX is the Pfizer crystal structure name
  - XXXXXX_s.csv is the sitemap labels file (e.g., `HIV-pr_K41R_V64I_0200758_LIG_1_s.csv`)
  - XXXXXX_s.pdb contains the sitemap interaction points
  - XXXXXX_a.pdb holds the atoms from XXXXXX_p.pdb which have a corresponding sitemap interaction point (in XXXXXX_s.pdb)
  - XXXXXX_rad.pdb contains those residues from the protein (XXXXXX_p.pdb) that have at least one heavy atom within 4.5 Å of any atom in the ligand (XXXXXX_l.mol2.)
- Ligand files generated by ASCbase during a search have the following naming
  - XXXXXX_NNNNN_l.mol2 is the ligand from the database structure XXXXXX in the reference frame of the query pocket given by the Nth hit from XXXXXX's pocket to the query pocket; where NNNNN is N written as 5 digits (zero padded) (e.g, `HIV-pr_K41R_V64I_0200758_LIG_1_00012_f.mol2`)
  - XXXXXX_NNNNN_f.mol2 is the fragment(s) of the database ligand (from XXXXXX); that is the portion of XXXXXX_NNNNN_l.mol2 that "fit" in the query's pocket.

Figure 53: An excerpt of the section describing the SimSite3D data file naming convention. The naming convention is very useful as it allows one to know what is in a file without having to open/view it.

A file naming convention was agreed upon between our group and our collaborators at Pfizer. This convention is not strictly required by the main programs, but many of the Python utilities depend on it to automatically parse file names and retrieve protein and ligand coordinate files.

1. Install the Boost.Python interface to ASCbase.
2. Append the installation location of ASCbasePy to your **$PYTHONPATH** variable.
3. Install numpy if it is not present on your system.
4. Install Pmw if it is not present on your system.
5. Get one of the later (0.99 or later) versions of the PyMOL sources. Build and install the PyMOL modules. (NOTE: the modules are only need by the plugin and seem to work seamlessly with the incentive builds of PyMOL in our lab).
6. Find the `pmg_tk/startup` directory and copy the ASCbase PyMOL module from your **$ASCBASE_INSTALL_DIR**/ASCbasePy/PyMOL_plugins directory to the `pmg_tk/startup` directory.
7. Test the plugin by loading an ASCbase results file and query site map.
8. If the plugin fails to initialize, remove it from the `pmg_tk/startup` directory so that PyMOL can be used.

Figure 54: One method to install the PyMOL plugin to load SimSite3D hits from a .out file into the PyMOL molecular graphics program.

There are a number of hurdles to installing the PyMOL plugin to load SimSite3D hits into the PyMOL molecular graphics viewer. These hurdles are due primarily to two issues: some of the SimSite3D Python utilities are required, and there are several ways to install PyMOL. The SimSite3D Python utilities are required because PyMOL does not have its own methods to load SimSite3D results and site map files. PyMOL can be installed either using system libraries, or using its own version of Python and dependencies. Because of these complications, it is difficult to foresee complications which were not seen on our lab machines.

B.4 Remarks

In this appendix section, we briefly covered the work that went into developing SimSite3D and creating its documentation. This work is important because we expressly intend to distribute SimSite3D, and documentation is one of the main reason why users

tend to quickly discard freely available software tools. Since a substantial amount of documentation already exists, it is much easier to refine it and add pertinent details.

BIBLIOGRAPHY

Bibliography

[1] A. Aitken, "On least squares and linear combination of observations," *Proc. Roy. Soc. Edinb.*, vol. 55, pp. 42–48, 1934.

[2] I. L. Alberts, N. P. Todorov, and P. M. Dean, "Receptor flexibility in de novo ligand design and docking," *J. Med. Chem.*, vol. 48, no. 21, pp. 6585–6596, Oct. 2005.

[3] S. F. Altschul and W Gish, "Local alignment statistics," *Methods in Enzymology*, vol. 266, pp. 460–480, 1996.

[4] S. F. Altschul, W Gish, W Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *JMB*, vol. 215, no. 3, pp. 403–410, Oct. 1990.

[5] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.

[6] N. Andrusier, R. Nussinov, and H. J. Wolfson, "FireDock: fast interaction refinement in molecular docking," *Proteins: Structure, Function, and Bioinformatics*, vol. 69, no. 1, pp. 139–159, 2007.

[7] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223 –230, Jul. 1973.

[8] P. Baerlocher, "Inverse kinematics techniques for the interactive posture control of articulated figures," Ph. D. Ecole Polytechnique Federale de Lausanne, 2001.

[9] J. A. Barker and J. M. Thornton, "An algorithm for constraint-based structural template matching: application to 3D templates with statistical analysis," *Bioinformatics*, vol. 19, no. 13, pp. 1644–1649, Sep. 2003.

[10] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide protein data bank," *Nat. Struct. Mol. Biol.*, vol. 10, no. 12, p. 980, Dec. 2003.

[11] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucl. Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000.

[12] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *ACM Comput. Surv.*, vol. 17, pp. 75–145, 1 Mar. 1985.

[13] P. Besl and H. McKay, "A method for registration of 3-D shapes," *TPAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[15] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM TOG*, vol. 1, no. 3, pp. 235–256, 1982.

[16] C. Brandon and J. Tooze, *Introduction to Protein Structure*, 2nd. Garland, 1998.

[17] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching," *PNAS*, vol. 103, no. 5, pp. 1168–1172, 2006.

[18] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, "CHARMM: a program for macromolecular energy, minimization, and dynamics calculations," *J. Comp. Chem.*, vol. 4, pp. 187–217, 1983.

[19] B. R. Brooks, C. L. B. III, A. D. M. Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M York, and M. Karplus, "CHARMM: the biomolecular simulation program," *J. Comp. Chem.*, vol. 30, no. 10, pp. 1545–1614, 2009.

[20] S. R. Buss and J. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics Tools*, vol. 10, no. 3, pp. 37–49, 2005.

[21] D. Colbry and G. Stockman, "The 3DID face alignment system for verifying identity," *Image and Vision Computing*, vol. 27, no. 8, pp. 1121–1133, Jul. 2009.

[22] P. Cozzini, G. E. Kellogg, F. Spyrakis, D. J. Abraham, G. Costantino, A. Emerson, F. Fanelli, H. Gohlke, L. A. Kuhn, G. M. Morris, M. Orozco, T. A. Pertinhez, M. Rizzi, and C. A. Sotriffer, "Target flexibility: an emerging consideration in drug discovery and design," *J. Med. Chem.*, vol. 51, no. 20, pp. 6237–6255, Oct. 2008.

[23] P. Craven and G. Wahba, "Smoothing noisy data with spline functions," *Numerische Mathematik*, vol. 31, pp. 377–403, 1979.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE CVPR*, vol. 1, 2005, 886–893 vol. 1.

[25] I. W. Davis and D. Baker, "RosettaLigand docking with full ligand and receptor flexibility," *JMB*, vol. 385, no. 2, pp. 381–392, Jan. 2009.

[26] Z. Deng, C. Chuaqui, and J. Singh*, "Structural interaction fingerprint (SIFt): a novel method for analyzing Three-Dimensional ProteinLigand binding interactions," *J. Med. Chem.*, vol. 47, no. 2, pp. 337–344, 2004.

[27] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd. New York: Wiley, 2001.

[28] B. S. Duncan and A. J. Olson, "Shape analysis of molecular surfaces," *Biopolymers*, vol. 33, no. 2, pp. 231–238, 1993.

[29] D. Eisenberg and A. D. McLachlan, "Solvation energy in protein folding and binding," *Nature*, vol. 319, no. 6050, pp. 199–203, Jan. 1986.

[30] S. P. Ellis, "Instability of least squares, least absolute deviation and least median of squares linear regression, with a comment by stephen portnoy and ivan mizera and a rejoinder by the author," *Statistical Science*, vol. 13, no. 4, pp. 337–350, Nov. 1998.

[31] T. Fan, G. Medioni, and R. Nevatia, "Recognizing 3-D objects using surface descriptions," *TPAMI*, vol. 11, no. 11, pp. 1140–1157, 1989.

[32] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[33] H. J. Feldman and P. Labute, "Pocket similarity: are alpha carbons enough?," *J. Chem. Inf. Model.*, vol. 50, no. 8, pp. 1466–1475, Aug. 2010.

[34] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[35] R. Fisher, "Dispersion on a sphere," *Proc. Royal Soc. London A*, vol. 217, pp. 295–305, 1953.

[36] R. S. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *Computational Science & Engineering, IEEE*, vol. 4, no. 4, pp. 42–49, Oct. 1997.

[37] N. D. Gold and R. M. Jackson, "A searchable database for comparing ProteinLigand binding sites for the analysis of StructureFunction relationships," *J. Chem. Inf. Model.*, vol. 46, no. 2, pp. 736–742, Mar. 2006.

[38] G. H. Golub, M. Heath, and G. Wahba, "Generalized Cross-Validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979.

[39] D. S. Goodsell, G. M. Morris, and A. J. Olson, "Automated docking of flexible ligands: applications of autodock," *Journal of Molecular Recognition*, vol. 9, no. 1, pp. 1–5, 1996.

[40] J. Greer and B. L. Bush, "Macromolecular shape and surface maps by solvent exclusion," *PNAS*, vol. 75, no. 1, pp. 303–307, Jan. 1978.

[41] M. J. Hartshorn, M. L. Verdonk, G. Chessari, S. C. Brewerton, W. T. M. Mooij, P. N. Mortenson, and C. W. Murray, "Diverse, High-Quality test set for the validation of ProteinLigand docking performance," *J. Med. Chem.*, vol. 50, no. 4, pp. 726–741, Feb. 2007.

[42] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.

[43] P. Hawkins, A. Skillman, and A. Nicholls, "Comparison of Shape-Matching and docking as virtual screening tools," *J. Med. Chem.*, vol. 50, no. 1, pp. 74–82, Jan. 2007.

[44] M. S. Head, *What works now and what do we need?*, Fairmont Chateau Whistler, Apr. 2010.

[45] J. Heringa and P. Argos, "Strain in protein structures as viewed through nonrotameric side chains: II. effects upon ligand binding," *Proteins: Structure, Function, and Bioinformatics*, vol. 37, no. 1, pp. 44–55, 1999.

[46] L. Holm, C. Ouzounis, C. Sander, G. Tuparev, and G. Vriend, "A database of protein structure families with common folding motifs.," *Protein Science*, vol. 1, no. 12, pp. 1691–1698, Dec. 1992.

[47] L. Holm and C. Sander, "Mapping the protein universe," *Science*, vol. 273, no. 5275, pp. 595–602, Aug. 1996.

[48] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, Apr. 1987.

[49] T. Hurst, "Flexible 3d searching: the directed tweak technique," *J. Chem. Inf. Comput. Sci.*, vol. 34, pp. 190–196, 1994.

[50] J. A. Ippolito, R. S. Alexander, and D. W. Christianson, "Hydorgen bond stereochemistry in protein structure and function," *J. Mol. Biol.*, vol. 215, no. 3, pp. 457–471, 1990.

[51] R. M. Jackson and M. J. E. Sternberg, "Protein surface area defined," *Nature*, vol. 366, no. 6456, p. 638, Dec. 1993.

[52] A. Jagannathan, "Segmentation and recognition of 3D point clouds within graph-theoretic and thermodynamic frameworks," PhD thesis, Northeastern University, Boston, MA, 2005.

[53] M. Jambon, A. Imberty, G. Delage, and C. Geourjon, "A new bioinformatic approach to detect common 3D sites in protein structures," *Proteins: Structure, Function, and Genetics*, vol. 52, no. 2, pp. 137–145, 2003.

[54] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, "Development and validation of a genetic algorithm for flexible docking," *JMB*, vol. 267, no. 3, pp. 727–748, Apr. 1997.

[55] W. L. Jorgensen, "Rusting of the lock and key model for Protein-Ligand binding," *Science*, New Series, vol. 254, no. 5034, pp. 954–955, Nov. 1991.

[56] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the OPLS All-Atom force field on conformational energetics and properties of organic liquids," *JACS*, vol. 118, no. 45, pp. 11 225–11 236, Jan. 1996.

[57] L. Kavraki, "Protein inverse kinematics and the loop closure problem," *Connexions Web site*, Jun. 2007.

[58] P. Koehl and M. Delarue, "Mean-field minimization methods for biological macromolecules," *Current Opinion in Structural Biology*, vol. 6, no. 2, pp. 222–226, Apr. 1996.

[59] A. Kouranov, L. Xie, J. de la Cruz, L. Chen, J. Westbrook, P. E. Bourne, and H. M. Berman, "The RCSB PDB information portal for structural genomics," *Nucleic Acids Research*, vol. 34, pp. D302–D305, Jan. 2006.

[60] E. Krissinel and K. Henrick, "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Acta D*, vol. 60, pp. 2256–2268, Dec. 2004.

[61] L. A. Kuhn, "Strength in flexibility: modeling side-chain conformational change in docking and screening," in *Structure-Based Drug Discovery*, London: Royal Society of Chemistry, 2008, pp. 177–187.

[62] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin, "A geometric approach to Macromolecule-Ligand interactions," *JMB*, vol. 161, no. 2, pp. 269–288, 1982.

[63] R. A. Laskowski, J. D. Watson, and J. M. Thornton, "Protein function prediction using local 3D templates," *JMB*, vol. 351, no. 3, pp. 614–626, 2005.

[64] B. Lee and F. M. Richards, "The interpretation of protein structures: estimation of static accessibility," *JMB*, vol. 55, no. 3, pp. 379–400, 1971.

[65] M. Lei, M. I. Zavodszky, L. A. Kuhn, and M. F. Thorpe, "Sampling protein conformations and pathways," *J. Comp. Chem.*, vol. 25, no. 9, pp. 1133–1148, 2004.

[66] C. Lemmen and T. Lengauer, "Computational methods for the structural alignment of molecules," *Journal of Computer-Aided Molecular Design*, vol. 14, no. 3, pp. 215–232, Mar. 2000.

[67] S. Leopoldseder, H. Pottmann, and H. Zhao, "The dˆ2-tree: a hierarchical representation of the squared distance function," Institute of Geometry, Mar. 2003.

[68] A. Li and R. Nussinov, "A set of van der waals and coulombic radii of protein atoms for molecular and solvent-accessible surface calculation, packing evaluation, and docking," *Proteins: Structure, Function, and Genetics*, vol. 32, no. 1, pp. 111–127, 1998.

[69] J. Liang, C. Woodward, and H. Edelsbrunner, "Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design," *Protein Science*, vol. 7, no. 9, pp. 1884–1897, 1998.

[70] A. Lubotsky, R. Philips, and P. Sarnak, "Hecke operators and distributing points on the sphere i," *Comm. Pure Appl. Math*, vol. XXXIX, S149–S138, 1986.

[71] A. Mademlis, P. Daras, D. Tzovaras, and M. Strintzis, "On 3D partial matching of meaningful parts," in *IEEE ICIP*, vol. 2, 2007, pp. 517–520.

[72] G. McGaughey, R. Sheridan, C. Bayly, J. Culberson, C. Kreatsoulas, S. Lindsley, V. Maiorov, J. Truchon, and W. Cornell, "Comparison of topological, shape, and docking methods in virtual screening," *J. Chem. Inf. Model.*, vol. 47, no. 4, pp. 1504–1519, Jul. 2007.

[73] J. C. Mitchell, "Sampling rotation groups by successive orthogonal images," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, p. 525, 2008.

[74] D. L. Mobley and K. A. Dill, "Binding of Small-Molecule ligands to proteins: "What you see" is not always "What you get"," *Structure*, vol. 17, pp. 489–498, 2009.

[75] E. W. Myers and W. Miller, "Optimal alignments in linear space," *Computer applications in the biosciences : CABIOS*, vol. 4, no. 1, pp. 11 –17, Mar. 1988.

[76] R. Najmanovich, J. Kuttner, V. Sobolev, and M. Edelman, "Side-chain flexibility in proteins upon ligand binding," *Proteins: Structure, Function, and Bioinformatics*, vol. 39, no. 3, pp. 261–268, 2000.

[77] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *JMB*, vol. 48, no. 3, pp. 443–453, Mar. 1970.

[78] C. Olson, "A probabilistic formulation for hausdorff matching," in *IEEE CVPR*, 1998, pp. 150–156.

[79] L. Pauling and R. B. Corey, "The pleated sheet. a new layer configuration of polypeptide chains," *PNAS*, vol. 37, pp. 251–256, 1951.

[80] L. Pauling, R. B. Corey, and H. R. Branson, "The structure of proteins: two Hydrogen-Bonded spiral configurations of the polypeptide chain," *PNAS*, vol. 37, pp. 235–240, 1951.

[81] S. Pellegrini, K. Schindler, and D. Nardi, "A generalization of the ICP algorithm for articulated bodies," in *British Machine Vision Conference*, 2008.

[82] R. Penrose, "A generalized inverse for matrices," *Proc. Cambridge Phil. Soc.*, vol. 51, pp. 406–413, 1955.

[83] W. Rudin, *Principles of Mathematical Analysis*, 3rd. New York: McGraw-Hill, Inc., 1976.

[84] E. Saber, Y. Xu, and A. M. Tekalp, "Partial shape recognition by sub-matrix matching for partial matching guided image labeling," *Pattern Recognition*, vol. 38, no. 10, pp. 1560–1573, Oct. 2005.

[85] E. B. Saff and A. B. J. Kuijlaars, "Distributing many points on a sphere," *Mathematical Intelligencer*, vol. 19, pp. 5–14, 1997.

[86] C. Sander and R. Schneider, "Database of homology-derived protein structures and the structural meaning of sequence alignment," *Proteins: Structure, Function, and Genetics*, vol. 9, no. 1, pp. 56–68, 1991.

[87] M. F. Sanner, A. J. Olson, and J. Spehner, "Reduced surface: an efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, no. 3, pp. 305–320, 1996.

[88] R. F. Sarraga, "Algebraic methods for intersections of quadric surfaces in GMSOLID," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 2, pp. 222–238, May 1983.

[89] S. Schmitt, D. Kuhn, and G. Klebe, "A new method to detect related function among proteins independent of sequence and fold homology," *JMB*, vol. 323, no. 2, pp. 387–406, Oct. 2002.

[90] V. Schnecke and L. Kuhn, "Virtual screening with solvation and ligand-induced complementarity," *Perspectives in Drug Discovery and Design*, vol. 20, no. 1, pp. 171–190, Dec. 2000.

[91] A. Shulman-Peleg, R. Nussinov, and H. J. Wolfson, "Recognition of functional sites in protein structures," *JMB*, vol. 339, no. 3, pp. 607–633, Jun. 2004.

[92] —, "SiteEngines: recognition and comparison of binding sites and proteinprotein interfaces," *Nucleic Acids Research*, vol. 33, no. Web Server issue, W337–W341, Jul. 2005.

[93] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *JMB*, vol. 147, no. 1, pp. 195–197, Mar. 1981.

[94] A. Stark, S. Sunyaev, and R. B. Russell, "A model for statistical significance of local similarities in structure," *JMB*, vol. 326, no. 5, pp. 1307–1316, Mar. 2003.

[95] F. Stein and G. Medioni, "Structural indexing: efficient 3-D object recognition," *TPAMI*, vol. 14, no. 2, pp. 125–145, 1992.

[96] L. Su and R. I. Cukier, "Hamiltonian replica exchange method study of escherichia coli and yersinia pestis HPPK," *J. Phys. Chem. B*, vol. 113, no. 50, pp. 16 197–16 208, Dec. 2009.

[97] M. E. Tonero, M. I. Zavodszky, J. R. V. Voorst, L. He, S. Arora, S. Namilikonda, and L. A. Kuhn, "Effective scoring functions for predicting ligand binding mode," *in preparantion*, 2011.

[98] I. Tun, E. Silla, and J. Pascual-Ahuir, "Molecular surface area and hydrophobic effect," *Protein Engineering*, vol. 5, no. 8, pp. 715 –716, Dec. 1992.

[99] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[100] D. Voet and J. G. Voet, *Biochemistry*, 3rd. Wiley, 2004.

[101] G. L. Warren, C. W. Andrews, A. Capelli, B. Clarke, J. LaLonde, M. H. Lambert, M. Lindvall, N. Nevins, S. F. Semus, S. Senger, G. Tedesco, I. D. Wall, J. M. Woolven, C. E. Peishoff, and M. S. Head, "A critical assessment of docking programs and scoring functions," *J. Med. Chem.*, vol. 49, no. 20, pp. 5912–5931, Oct. 2006.

[102] C. Welman, "Inverse kinematics and geometric constraints for articulated figure manipulation," Masters of Science, Simon Fraser University, 1993.

[103] W. J. Wilbur and D. J. Lipman, "Rapid similarity searches of nucleic acid and protein data banks," *PNAS*, vol. 80, no. 3, pp. 726 –730, Feb. 1983.

[104] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, Oct. 1996.

[105] A. Yershova, S. Jain, S. M. LaValle, and J. C. Mitchell, "Generating uniform incremental grids on SO(3) using the hopf fibration," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 801–812, 2009.

[106] M. I. Zavodszky and L. A. Kuhn, "Side-chain flexibility in proteinligand binding: the minimal rotation hypothesis," *Protein Science*, vol. 14, no. 4, pp. 1104–1114, Apr. 2005.

[107] M. I. Zavodszky, P. C. Sanschagrin, L. A. Kuhn, and R. S. Korde, "Distilling the essential features of a protein surface for improving protein-ligand docking, scoring, and virtual screening," *Journal of Computer-Aided Molecular Design*, vol. 16, no. 12, pp. 883–902, Dec. 2002.

[108] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, 2003.