SANDIA REPORT

SAND2005-7984
Unlimited Release
Printed January 2006

Developing Algorithms for Predicting Protein-Protein Interactions of Homology Modeled Proteins

Diana C. Roe, Ken L. Sale, Shawn B. Martin, and Jean-Loup Faulon

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SANDIA REPORT
SAND2005-7984
Unlimited Release
Printed January 2006

Developing Algorithms for Predicting Protein-Protein Interactions of Homology Modeled Proteins

Diana C. Roe, Ken L. Sale, Shawn B. Martin, and Jean-Loup Faulon
Biosystems Research Department, 8321
Sandia National Laboratories
Livermore, California 94550

Abstract

The goal of this project was to examine the protein-protein docking problem, especially as it relates to homology-based structures, identify the key bottlenecks in current software tools, and evaluate and prototype new algorithms that may be developed to improve these bottlenecks. This report describes the current challenges in the protein-protein docking problem: correctly predicting the binding site for the protein-protein interaction and correctly placing the sidechains. Two different and complementary approaches are taken that can help with the protein-protein docking problem. The first approach is to predict interaction sites prior to docking, and uses bioinformatics studies of protein-protein interactions to predict theses interaction site. The second approach is to improve validation of predicted complexes after docking, and uses an improved scoring function for evaluating proposed docked poses, incorporating a solvation term. This scoring function demonstrates significant improvement over current state-of-the art functions. Initial studies on both these approaches are promising, and argue for full development of these algorithms.

Table of Contents

Figures

Tables

Executive Summary

The goal of this project was to examine the protein-protein docking problem, especially as it relates to homology-based structures, identify the key bottlenecks in current software tools, and evaluate and prototype new algorithms that may be developed to improve these bottlenecks. This report describes the current challenges in the protein-protein docking problem: correctly predicting the binding site for the protein-protein interaction and correctly placing the sidechains. Two different and complementary approaches are taken that can help with the protein-protein docking problem. The first approach is to predict interaction sites prior to docking, and uses bioinformatics studies of protein-protein interactions to predict theses interaction site. The second approach is to improve validation of predicted complexes after docking, and uses an improved scoring function for evaluating proposed docked poses, incorporating a solvation term. This scoring function demonstrates significant improvement over current state-of-the art functions. Initial studies on both these approaches are promising, and argue for full development of these algorithms.

Nomenclature

PDB	Protein Data Bank
CAPRI	Critical Assessment of prediction of interactions
Homology models	Protein structure prediction modeled from protein with similar sequence
Pose	Term used to describe the combined docked orientation and conformation
Decoy	Term used to described an incorrect docked pose
Rotamer	Sidechain conformation
FDPB	Finite difference Poisson-Boltzman
RMSD	Root mean square deviation
Vdw	Van der Waals interactions
Å	Angstrom units
Native	"correct" pose found in the X-ray crystal complex
Decoy	Pose with low score but not in native orientation (usually far away)

This page intentionally left blank.

Developing Algorithms for Predicting Protein-Protein Interactions of Homology Modeled Proteins

Background

As the complete genomes of numerous species are now being determined, the next major challenge in biology is to examine the proteins encoded in these genomes, and how the proteins interact with each other. A key component of this understanding is to model how proteins interact at the atomistic level. This is often referred to as the protein-protein "docking" problem, analogous with a ship docking into a port. By studying the structural details of how proteins interact, we can elucidate their functional properties. This can lead to important biological applications in the biodefense and pharmaceutical industries. For example, modeling how a protein toxin from a threat bioagent binds to its receptor protein in humans would provide a starting point for the rapid development of therapeutics, in the form of small molecules that block that protein-protein interaction. Furthermore, it is becoming increasingly possible to perform large-scale docking studies to predict protein-protein interactions, to understand cellular networks by predicting which proteins may bind to each other. These types of studies can complement and refine high-throughput predictions made using only sequence information of the proteins involved.

The Docking Problem

The protein-protein docking problem can be broken down into three components: predicting the site of interaction or binding site on each protein, predicting how the proteins will orient with respect to one another to bind at these sites of interaction, and predicting how the three-dimensional shape of each protein (the protein conformation) will modify itself upon binding. Although the docking problem is related to the protein folding problem, in practice the docking problem requires its own specialized algorithms. The focus in protein-protein docking is largely on solving the relative orientation of two interacting proteins, and the challenge is to identify how protein-protein binding domains (large surface "pockets" of interaction that do not necessarily resemble the packing motifs found within a protein) bind and form an active complex configuration. The protein-protein docking problem is solved for the trivial case when the three-dimensional structures of both proteins *are known in the binding conformation.* In this case it is a simple 3-D jigsaw problem. However, predicting how the proteins will interact when only the individual *un*bound structures are known is a difficult problem, because of changes in protein conformation upon binding (referred to as "induced fit"). This was evidenced in a recent CAPRI (Critical Assessment of **PR**ediction of Interactions) contest, an international competition to predict the final docking interaction of two proteins given their unbound 3-dimensional structures [1]. In this contest, in two out of seven of the protein-protein docking test cases, not even the binding site on each protein, much less the binding orientation between the two proteins, was correctly identified.

Docking to Homology Modeled Proteins

As the protein-protein docking problem is not yet solved, an even greater challenge is predicting protein interactions in proteins whose exact structure is uncertain. High resolution x-ray crystal structures do not exist for many proteins of biological interest, especially those from threat bioagents. Lower resolution models can be generated for many of these proteins, by looking at structures with similar amino acid sequences and assuming they will have similar three-dimensional structures, in a process called *homology modeling*. About 30% sequence identify is usually sufficient to generate a reasonable homology model. As there are currently over 34,000 protein crystal structures solved in the Protein Data Bank (PDB) and this number is growing rapidly [2], it is increasingly possible to generate homology structures for non-membrane target proteins. However docking to these homology models is an even greater challenge than docking to crystal structures, due to the large potential errors in the model structures, particularly in sidechain placement (see Figure 1.)

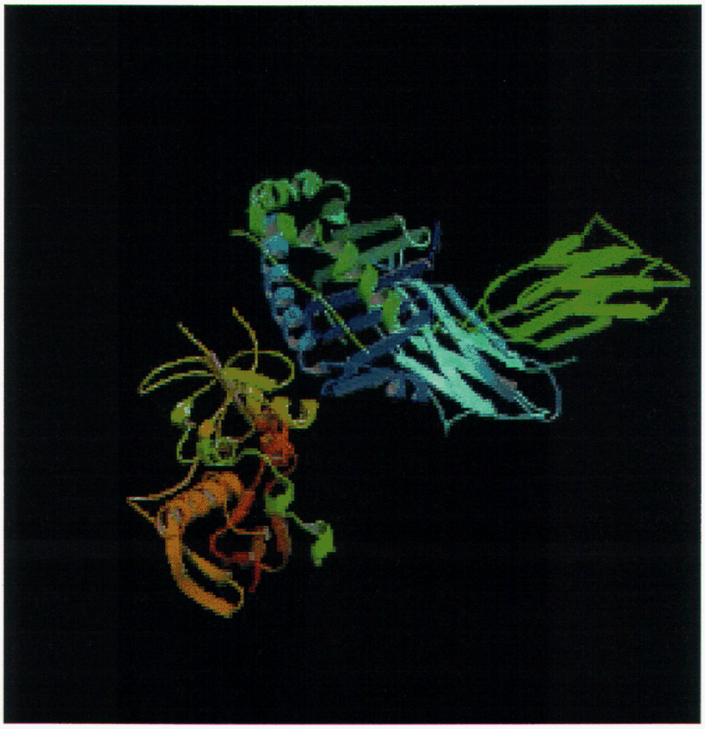


Figure 1. Crystal structure of a docked complex. The Mhc Class II Protein Hla-Dr1 (blue-green) is complexed with the superantigen sec3 protein toxin (red-orange).

Addressing Current Challenges in Protein-Protein Docking

Thus, the current biggest challenges in protein-protein docking are: correctly predicting the interaction site for the protein-protein binding, and correctly placing the protein sidechains, which is especially important with homology modeled structures where initial sidechain positions may be incorrect. Most sidechain placement strategies involve discretizing the possible conformations of the sidechains into a set of low energy "rotamers," which are usually determined from statistical analysis of the PDB structures [3]. The sidechain placement problem thus becomes a search problem through the set of all rotamers at each sidechain position. In this report we develop algorithms to address both the binding site identification problem and the sidechain placement problem. In the first part we develop a novel method to identify protein-protein interaction sites prior to docking, using constraints based on knowledge gleaned from protein-protein interaction databases. We show how this can correctly identify interacting surface patches in known protein complexes. In the second study, we develop and validate a new scoring function for evaluating protein complexes that incorporates a more physically realistic solvation term. We demonstrate significant improvement over current methods in discriminating correct binding orientations from a set of "decoy" orientations. This function could be used as a post-filter to re-rank docking predictions, or combined with a rotamer search strategy to identify correct sidechain positions. For the latter purpose, we also show how this function can be decomposed into a form that is a pairwise combination of sidechain rotamers, that can lend itself to more powerful and exhaustive search strategies such as Integer Programming [4].

This page intentionally left blank.

Algorithm for Predicting Protein-Protein Interacting Domains in Protein Complexes

Given the structures of the proteins involved in a protein complex, the protein – protein docking problem consists of modeling the structure of the complex by determining the most favored molecular interactions. In the common "lock and key" analogy, the protein docking problem is that of finding the part of one protein corresponding to the key and finding the part on the second protein corresponding to the lock. Even in the simplest case in which each protein is treated as a rigid body, this process is complicated by the enormous conformational space that must be searched. Here we propose a method for determining the most likely interacting domains in a protein complex, with the goal of reducing the size of the conformational search space and thus increasing protein–protein docking efficiency.

The approach taken for predicting protein domain interactions is taken from our recent work on predicting whether two proteins are likely to interact [5]. Here we extend that work to the prediction of the domains within a pair of interacting proteins that specify the protein–protein interaction.

Methods

Support Vector Machines

SVMs are classifiers (there are also SVMs that perform regression [6]) that are described thoroughly by their inventor [7]. SVMs are very adaptable and have been applied successfully to a wide variety of problems. Recently, there has been interest in the application of SVMs to biological problems such as classification of gene expression data [8], homology detection [9] and prediction of protein-protein interaction [10], as well as many additional problems. For an introduction to SVMs, see Burges [11] and Cristianini and Shawe-Taylor [12]. To describe an SVM precisely, suppose our data are given as pairs $\{(x_i, y_i)\} \subset \square^{n} \times \{\pm 1\}$ {(xi, yi)}. In other words, suppose our data consist of two classes (1 and −1, or in our case binding and nonbinding protein pairs). Using this notation an SVM assumes the form $f(\mathbf{x}) = \sum \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$, where $f : \square^{n} \to \square$ is a decision function (**x** belongs to class 1 if $f(\mathbf{x})$ is greater than some threshold *t*, or to class −1 otherwise), $k : \square^{n} \times \square^{n} \to \square$ is a kernel function, otherwise known as a dot product in some vector space, and the constants *b* and α_i are obtained by solving a quadratic programming problem [11]. The threshold *t* is typically 0, although it may be varied to obtain classifiers that are more or less accurate on positive predictions. SVMs have several advantages over other classifiers although we do not discuss them here. Instead, we refer to Vapnik [7] and Bennett and Campbell [13], among others. To implement the SVMs in this study we used the SVMlight algorithm [14] with a custom kernel based on signature.

Signature Molecular Descriptor

Classifying protein domains as either interacting domains or non-interacting domains requires the use of a suitable method for encoding protein sequence information in an appropriate vector space. In this case, the challenge was to represent variable length protein domains as vectors containing the information necessary to best predict whether two protein domains are likely to interact.

The sequence information coding strategy used here was to use a specific instance (height 1) of the more general signature molecular descriptor [15-18] as described by Martin *et al.* [5]. The molecular signature is formulated as a function $s:\{\text{variable length amino acid sequence}\} \to F$ defined by $s(A)=\sum \sigma_i z_i$, where A is an amino acid sequence, $\mathbf{z}_i$ is a basis vector in the signature space $F \cong \square^N$ and σ_i is the number of occurrences of $\mathbf{z}_i$ in A. A signature consists of an amino acid and its neighbors, and the signature space consists of all possible signatures. A height 1 signature consists of a root (the middle letter) and its two immediate neighbors—one amino acid residue in either direction, ordered alphabetically.

As an example, consider the six-letter amino acid sequence LVMTTM. The height 1 signatures correspond to the four trimers in this sequence: LVM, VMT, MTT and TTM. Thus, the signatures corresponding to the four trimers are V (LM), M(T V), T (MT) and T (MT), so that s(LVMT TM) = V (LM)+M(T V)+2T (MT). Notice that MTT and TTM generate the same signature (due to symmetry) and therefore contribute two occurrences to the sum $s(A)=\sum \sigma_i z_i$. This is actually a very simple usage of signature, and that signature can be extended to handle longer subsequences (height 2, 3, . . .), as well as non-linear sequences. In fact, signature was originally developed to describe molecules in cheminformatics. Recently, however, signature has also been used successfully in applications to HIV protease-1 peptide prediction [17] and inverse design of LFA-1/ICAM-1 peptides [18]). Signature has the following advantages over other descriptors [15, 16]: (1) signature has been shown to be competitive with other descriptors (e.g. Molconn-Z) in terms of deriving quantitative structure–activity relationships and predicting various properties; (2) signature is canonical in the sense that it can often be used to derive other descriptors (in the case of amino acid sequences we note that descriptors such as hydrophobicity [used in Bock and Gough [10]] are encompassed using signature with a smaller alphabet; (3) signature encodes information about structure as well as sequence by keeping track of neighborhoods. Thus, signature is information rich, and, in particular, enables the solution of inverse problems.

Signature has been a useful descriptor in the past, and has the important practical advantage for us that it provides a vector representation of an amino acid sequence. We exploit this fact to develop a signature-based SVM for use in the prediction of protein–protein interaction.

Signature Kernel

As mentioned previously, signature can be used to obtain vector representations of variable length amino acid sequences. This is, of course, the minimum requirement for the application of an SVM to our problem. It is more computationally efficient, however, if we use the fact that SVMs do not actually require the storage/use of these very long sparse vectors. SVMs only require dot products between the vectors. Since this dot product is the composition of the standard Euclidean dot product with the signature function, we call it the signature kernel. The signature kernel is given as *k*: {variable length amino acid sequences}$^2 \rightarrow \square$, where $k(A,B)=s(A)\cdot(B)$.

Here, we note that our kernel is very similar to the string kernels in Leslie *et al.* [9]. For us, the only real difference is that we used a symmetric formulation, while Leslie *et al.* do not, primarily due to problem domain. In fact, the string kernels in Leslie *et al.* could potentially improve the performance of our algorithm, as these kernels allow comparisons of sequences with mismatches. On the other hand, mismatches in subsequences of length 3 would be of negligible value, and while we could have used longer subsequences, we did not find this to be necessary. In fact, we did some experiments (on the H. pylori data) where we tried longer subsequences: we found that these subsequences did not provide any improvement in performance and therefore discontinued their use. In the end, we focused more on the product signature formulation (discussed in the next section), and chose to use our simpler string kernel based on signature. The signature kernel elegantly combines signature with SVMs and thus incorporates all of the advantages of the two methods. Most importantly, the SVMs allow the use of a very large number of signatures (up to 100 k in the applications we consider later). This would be unfeasible with other methods [e.g., multilinear regression, as used previously in Visco *et al.* [15]; Faulon *et al.* [16, 17]; Churchwell *et al.* [18].

Product Signature

The signature kernel SVM, as presented above, will only work with data points that consist of a single amino acid sequence. This is a problem since our data points are, in fact, pairs (protein–protein pairs) of amino acid sequences. To overcome this difficulty, we must define signature for pairs of amino acid sequences, and we must also provide a kernel that gives the inner product between two protein–protein pairs. To define signature for protein–protein pairs, we use the notion of a tensor product between vectors (this can be found in many standard texts on linear algebra). For our purposes, we define the tensor product between $\mathbf{a}=(a_1,\cdots,a_n)^{\mathrm{T}}\in\square^n$ and $\mathbf{b}=(b_1,\cdots,b_n)^{\mathrm{T}}\in\square^m$ to be $\mathbf{a}\otimes\mathbf{b}=(a_1b_1,a_1b_2,\ldots,a_1b_m,a_2b_1,\ldots,a_nb_m)^{\mathrm{T}}\in\square^{nm}$, and we observe that the entries in $\mathbf{a}\otimes\mathbf{b}$ are the same as the entries contained in the outer product $\mathbf{ab}^{\mathrm{T}}$. Using this definition, the signature for pairs, or the signature product, $s\otimes s$: {amino acid sequences}$^2 \rightarrow F\otimes F\cong\square^{N^2}$, is taken to be $s\otimes s(A,B)-s(A)\otimes s(B)$. Using this definition of signature product, we can now apply an SVM to our problem by specifying a kernel $\tilde{k}:(F\otimes F)\times(F\otimes F)\rightarrow\square$ that gives a dot product in the signature product space.

We use the standard Euclidean inner product so that the signature product kernel is defined by $\tilde{k}\left(\left(A,B\right),\left(C,D\right)\right)=\left(s\left(A\right)\otimes s\left(B\right)\right)\cdot\left(s\left(C\right)\otimes s\left(D\right)\right)$.

We are now in a position to apply an SVM to the problem of predicting protein–protein interaction. Computationally, however, there is one final obstacle to overcome. Specifically, the use of the signature product effectively squares the complexity of the calculation. This is easily seen when we observe that $F\cong\square^{N}$ so that $F\otimes F\cong\square^{N^2}$. In reality, the complexity does not increase according to N but rather according to the lengths of the amino acid sequences involved. Nevertheless, the computational complexity is squared, and this causes a problem that must be addressed if we are to process large datasets.

Fortunately, there is a simple way to fix this problem. If we write (for clarity) $\mathbf{a}=s\left(A\right)$, $\mathbf{b}=s\left(B\right)$, $\mathbf{c}=s\left(C\right)$ and $\mathbf{d}=s\left(D\right)$ a = s(A), then we can see that

$$\begin{aligned}\tilde{k}\left(\left(A,B\right),\left(C,D\right)\right)&=\left(s\left(A\right)\otimes s\left(B\right)\right)\cdot\left(s\left(C\right)\otimes s\left(D\right)\right)\\&=\text{trace }\left(\left(\mathbf{ab}^{\mathrm{T}}\right)\left(\mathbf{cd}^{\mathrm{T}}\right)^{\mathrm{T}}\right)\\&=\text{trace }\left(\mathbf{ab}^{\mathrm{T}}\mathbf{dc}^{\mathrm{T}}\right)\\&=\left(\mathbf{b}^{\mathrm{T}}\mathbf{d}\right)\text{trace}\left(\mathbf{ac}^{\mathrm{T}}\right)\\&=\left(\mathbf{b}^{\mathrm{T}}\mathbf{d}\right)\left(\mathbf{a}^{\mathrm{T}}\mathbf{c}\right)\\&=k\left(A,C\right)k\left(B,D\right)\end{aligned}$$

where trace (**X**) is the sum of the diagonal elements of a square matrix **X**, and k is the signature kernel. Equation (1) shows that to compute the signature product kernel of two protein–protein pairs, we need only to compute the signature kernel between combinations of the individual proteins. Thus, we have removed the squared computational complexity.

Symmetric Signature Product

We can obtain an additional improvement in the signature product by enforcing symmetry in the protein–protein order. In other words, we can make a protein pair (*A, B*) equivalent to protein pair (*B, A*). This symmetry is easily achieved by defining the symmetric signature product $\Gamma\left(A,B\right)=s\left(A\right)\otimes s\left(B\right)+s\left(V\right)\otimes s\left(A\right)$. The associated symmetric signature product kernel is then $\gamma\left(\left(A,B\right),\left(C,D\right)\right)=2\left(k\left(A,C\right)k\left(B,D\right)+k\left(A,D\right)k\left(B,C\right)\right)$.

Normalized Signature Product

Next, we can use a normalized dot product to compensate for potential differences in the length of the amino acid sequences involved in our calculations. In particular, a normalized version of the signature kernel can be implemented as $k\left(A,B\right)/\sqrt{\left(k\left(A,A\right)k\left(B,B\right)\right)}$. This kernel extends directly to the signature product kernel and is only slightly more complicated in the case of the symmetric signature product.

Other Adjustments

As mentioned previously, SVMs are very flexible, and we found that we could occasionally achieve minor (2%) improvements in performance by adjusting kernels or using preprocessing techniques. In particular, we found that preprocessing by removal of signatures occurring only once in the dataset occasionally resulted in better performance. This improvement was used in the case of the yeast SH3 data. We also found that using a Gaussian kernel in combination with the product signature kernel could result in better performance. In particular, we used the kernel $\exp\left[-\gamma\left(\tilde{k}(A,A)-2\tilde{k}(A,B)+\tilde{k}(B,B)\right)\right]$, where $\tilde{k}$ is the product signature as described previously, and γ was chosen to be 0.5.

Structural Comparisons

Crystal structures of the complex of cyclophilin A with the N-terminal domain of HIV-1 capsid (PDB ID: 1AK4) and the histocompatibility complex (PDB ID: 1AGD) were taken from the Protein Data Bank (PDB). Our sequence based predictions of the interacting domains were compared to crystal structures by mapping the predicted interacting domains onto the structure of the complex using the visual molecular dynamics (VMD) program (Humphrey *et al.*, 1996).

Results

Training

The support vector machine was trained on the human database of interacting proteins (DIP). The DIP database catalogues experimentally determined interactions among proteins, and combines information from a variety of experimental sources to create a single, consistent set of protein – protein interactions. The human DIP contains 898 proteins and 1379 documented protein – protein interactions based on 1998 distinct experiments.
Since databases of protein domain interactions are unavailable, we trained the SVM on the full-length sequences of interacting protein pairs. Protein pairs for which no documented protein – protein interactions exist were assumed to be protein pairs that do not interact and were used as negatives in the SVM training.

The SVM model was validated using ten-fold cross-validation in which 90% of the data are used to create the model. The model is then tested using the remaining 10% of the data. This procedure is repeated 10 times and the average accuracy is calculated over the 10 cross-validations. The average accuracy, [defined as $\left(\text{True}^{+}+\text{True}^{-}\right)/\left(\text{True}^{+}+\text{True}^{-}+\text{False}^{+}+\text{False}^{-}\right)$] for predicting protein – protein interactions in the human DIP was 73.1%. This accuracy compares favorably to the 70 - 80% accuracy previously reported for H. Pylori, human and mouse datasets [5].

Testing

To test whether the SVM built from full-length sequences is capable of predicting interactions at the protein domain level, we extracted the sequence from a set of 20 human protein complexes for which the X-ray structure was available in the protein data bank (PDB).

Each sequence was then split into domains of width *N* amino acids spaced *M* residues apart by sliding a window of length *N* across the sequence, offsetting each window by *M* residues. Pairs of the resulting pairs of *N*-mers were then scored using the SVM trained protein interaction classifier.

Human Cyclophilin A Bound to N-Term Domain of HIV-1 Capsid

Each of the two proteins of the complex of human cyclophilin A and the N-terminal domain of HIV-1 capsid was split into domains of 30 amino acids with a one amino acid offset, and all possible pairs of 30 amino domains were scored using the SVM trained on the human DIP. The scores for the possible interacting domains between human cyclophilin A and the N-term domain of HIV-1 capsid are presented graphically Figure 2, in which the color range runs from dark red (highest score) to cyan (lowest score).

The maximum in this plot occurs at position (75, 77), which corresponds to the interaction between residues 75 to 105 of human cyclophilin A and residues 77 to 107 of the N-terminus of HIV-1 capsid. The dark red region around this point indicates that the most likely interactions occur among several domains adjacent to these two domains. Figure 3 shows the domains corresponding to this region mapped onto the three dimensional structure of the complex. This mapping of the predicted interacting domains onto the crystal structure shows that the predicted domains are larger than expected under the assumption that the most important domain interactions are those in which atoms are in very close proximity. Thus, the signature product may either be picking up longer range interactions or portions of these domains may be the result of false positive predictions at the protein domain level.

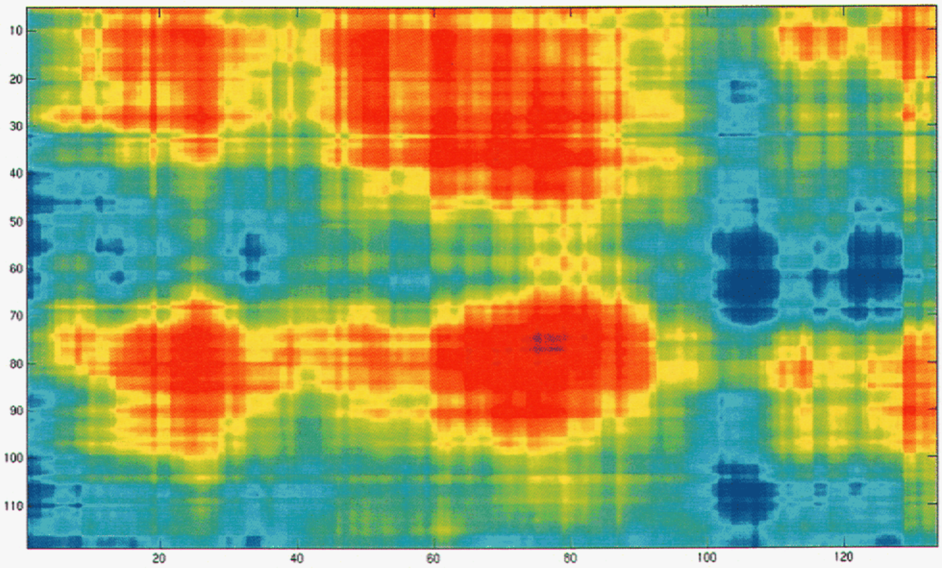


Figure 2. Colormap of interaction scores for cyclophilin A with HIV-1 capsid. The interaction scores for 30 amino acid long domains taken from cyclophilin A (x-axis) and the N-terminal domain of HIV-1 capsid (y-axis). Dark red corresponds to the highest scoring domain pairs and cyan corresponds to the lowest scoring domain pairs.

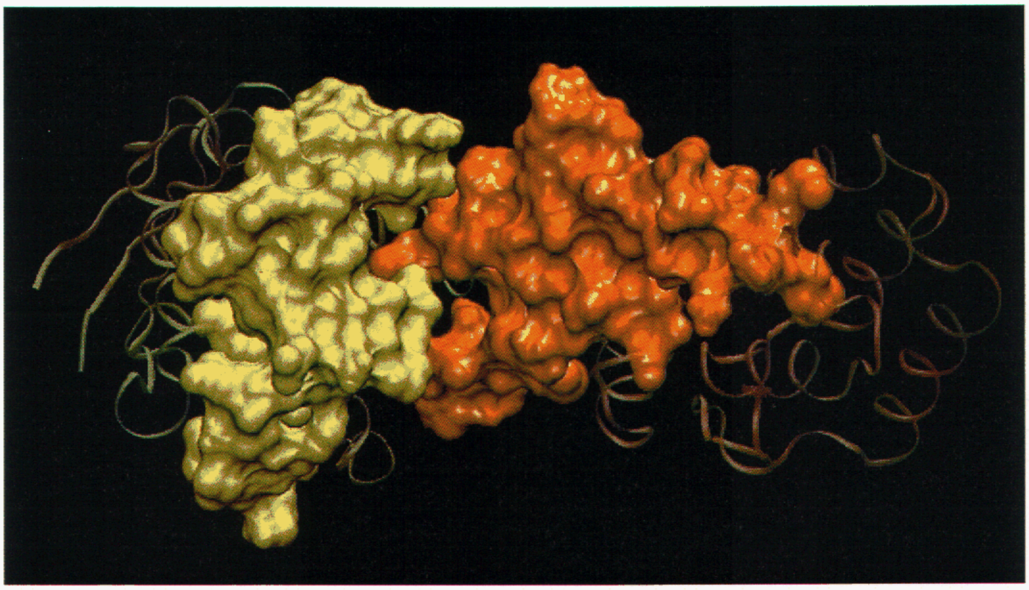


Figure 3. Mapping of the predicted interacting domains from human cyclophilin A (yellow) and the N-terminus of HIV-1 capsid (orange) onto the X-ray structure of the complex (PDB ID: 1AK4).

Human Histocompatibility Complex

We performed the same analysis on the human histocompatibility complex using our SVM trained on the human DIP. The predicted interacting domains for the histocompatibility complex (1AGD) are shown in Figure 4. As with the human cyclophilin A, the predicted interacting domains are larger than what might be expected, but the domains in direct contact appear to have been recovered. Again, the signature product may in fact be predicting larger interacting domains, because it may be picking up longer range interactions.

Discussion

Our goal in extending the signature product approach from predicting protein – protein interactions to predicting the interactions at the protein domain level was to reduce the size of the conformational search space during protein docking. Based on the results shown in Figure 3 and Figure 4, our predictions are capable, at a minimum, of determining the most likely large interacting surfaces. In the context of the protein – protein docking problem, this is valuable information in that it serves to reduce the size of the conformational search by eliminating large faces of the two proteins that are very unlikely to interact significantly.

The fact that the predicted interacting domains are larger than those that are indicated by measures such as changes in solvent accessible surface area (ASA) upon complex formation or inter-domain distances, suggests that the signature product accounts for other important longer range interactions such as electrostatics. Alternatively, the larger number and larger size of predicted interacting domains may indicate a propensity for the descriptor to produce false positive interactions.

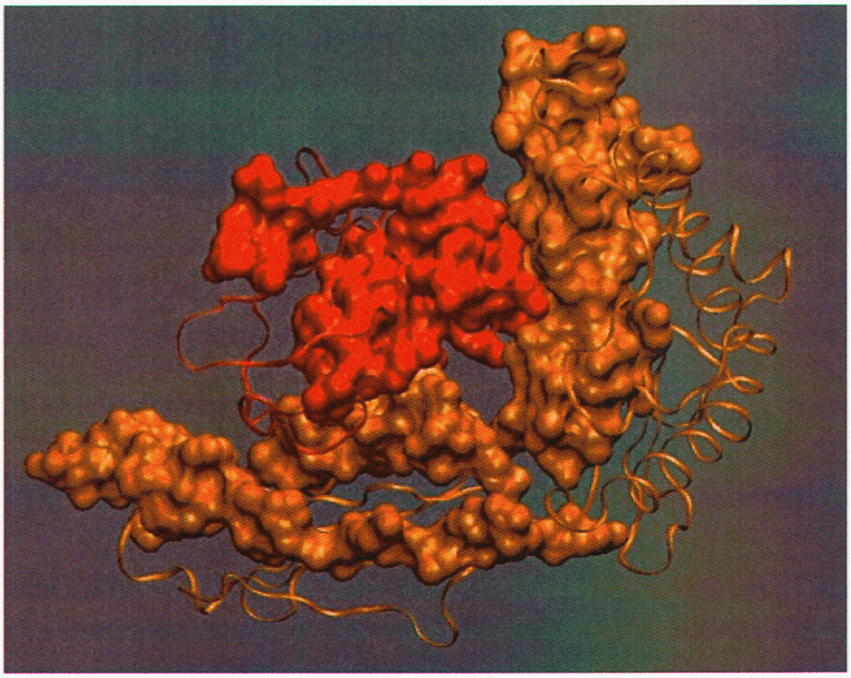


Figure 4. Mapping of the predicted interacting domains from human histocompatibility complex onto the X-ray structure of the complex (PDB ID: 1AGD).

These arguments point to two important points that should be made regarding this approach to predicting interacting protein domains. First, the SVM was trained using full-length amino acid sequences but was applied to sequence segments containing only 30 amino acids. This approach was necessitated by the lack of a convenient experimentally verified data set of protein – protein interactions compiled at the protein domain level. The consequence of this disconnect between the protein level training set and the domain level predictions may in part explain the tendency to predict large disconnected interacting domains. Second, validation of predicted results given the structure of a protein complex is difficult. Determining the important interacting domains from the 3D structure of the complex is not as straightforward as one might expect. Clearly, the residues in direct contact (van der Waals interactions) are important contributors; however, the important longer range interactions such as electrostatic interactions are not as easy to determine.

Nevertheless, as discussed, the ability to predict only the interacting "sides" or large domains of the proteins in a complex provides a means for greatly reducing the computational expense involved in docking proteins. Accurate prediction of the most important domain or amino acid level interactions will require the creation of an experimentally validated domain level training set and more accurate methods for determining the interacting domains from solved structures of protein complexes.

Validation of a Novel Scoring Function for Protein-Protein Complexes: Significant Improvement over Existing Scoring Schemes

Introduction

Although there are now a number of successful protein-protein docking programs [19-26], the problem is far from solved. This was most notable in the CAPRI contest, a blind contest for predicting protein-protein complexes based on unbound crystal structures, where in rounds 1 & 2 [1], there were no predictions even close to the correct binding orientation for 2 out of 7 of the target proteins. Even in more recent rounds 3-5, no single program could correctly identify the overall orientation for all 9 of the targets [27]. Several approaches have been taken to address this problem by post-filtering docked orientations, to identify correctly docked poses from "decoys": combining multiple scoring functions [28], surface patch approaches [29, 30, 31], and analyzing docking energy landscapes [22]. To evaluate scoring and post-filtering functions, one can examine their ability to discriminate correctly docked structures from mis-docked (or "decoy") complexes.

For example, in a recent paper on the rosettaDock program [20], researchers analyzed a benchmark set of 54 structurally unique protein complexes [32]. They created a set of 1000 "decoys" for each complex, starting from the bound backbone conformations of the proteins (but allowing sidechains to vary for each complex), and running them through their docking protocol. While many of the native complexes were correctly identified with this scheme, in 10 out of 54 of the structures the program was were unable to distinguish the native binding orientation from one of the decoys. They published their entire decoy set to challenge the community to develop better scoring schemes (http://graylab.jhu.edu/docking/decoys/). Our goal was to develop a more physically accurate scoring scheme, and evaluate it on these same decoy sets. This scoring scheme could be used as a complementary scheme to post-filter docking results, or could be directly incorporated into a docking algorithm such as rosettaDock, which has a separate module for sidechain placement and refinement of docking poses.

Many scoring functions have been developed over the years for evaluating protein-protein interactions, including free-energy based methods [21], empirical methods [20] and statistical potentials [33-35]. For purposes of speeding the time of the calculation, most of the existing force-field based functions employ a simple approximation for their solvation term based on atomic surface areas [36, 37] or atomic contact energies [38]. To create a more accurate model of binding energy, we developed a free-energy based scoring function that incorporated a more physical solvation term based on a finite difference Poisson-Boltzman (FDPB) model, considered the "gold-standard" for implicit solvation calculations. We used the ZAP Toolkit [39] to calculate the solvation energy, and incorporated it into an AMBER 95 force-field [40]. Unique to the ZAP FDPB calculation is the use of atom-centered Gaussians, which is both significantly faster than original FDPB implementations, making it tractable for use in docking scoring functions, and also solves many of the known problems with the FDPB approach with respect to discrete edge effects.

The results are both more physically realistic and more numerically stable [41]. Zap has been successfully added to both CHARMM and AMBER molecular dynamics packages [42]. We incorporated the Zap toolkit into the PDock program, for use with our AMBER-based scoring scheme in docking calculations. PDock is a Sandia software tool for docking implementing algorithms from DOCK 4.0 [43] and Sandia's Coliny [44] optimization suite. We validated the scoring scheme in PDock against the rosettaDock "decoy" test set and showed considerable improvement on discriminating native from decoy structures, correctly identifying all native complexes, as compared to 44 out of 54 from the rosettaDock function.

Along with its potential use as a final step in protein-protein docking, we wanted to see if our PDock scoring function could be useful for optimizing the sidechain positions in a docking orientation, to possibly replace the sidechain refinement step in a protein-protein docking code or as a stand-alone code. Sidechain rotamers are a set of low energy conformations of each sidechain, usually taken from a statistical analysis of all structures in the PDB [3]. In the sidechain optimization problem, the goal is to search among all rotamers of all sidechains in a protein-protein interface for the combination of rotamers that has the lowest energy. There are a number of search methods that can be used to identify the lowest combination of rotamers (for an overview of search methods see [45]), ranging from random search methods, such as *monte carlo* or genetic algorithms, to more powerful and exhaustive searches such as the pairwise-self-consistant field method (SCMF), dead-end elimination (DEE), and integer programming. These last methods require that a scoring function be decomposable into a form that can be treated as a pairwise combination of sidechain rotamers. We developed a pairwise decomposition for the PDock scoring function, so that we could take advantage of the powerful integer programming algorithms available at Sandia such as PICO combinatorial optimization library [46]. We validated that the pairwise formulation gave the same energy as the original implementation, and can be used for these search strategies.

Methods

PDock Scoring Function

The form for our scoring scheme is as follows:

$$(1) \qquad \Delta G_{bind} = \Delta G_{free,complex} - \Delta G_{free,prot1} - \Delta G_{free,prot2} = \Delta G_{free,complex} - \Delta G_{free,prot1+prot2}$$

Where $\Delta G_{complex}$ is the free energy of the complex and $\Delta G_{prot1+prot2}$ is the free energy of the unbound proteins. The individual terms of the scoring function are as follows:

$$(2) \qquad \Delta G_{free} = \Delta G_{vdw} + \Delta G_{es} + \Delta G_{solv} + \Delta G_{cav}$$

Where ΔG_{vdw} is the non-bonded van der Waals energy, ΔG_{solv} is the solvation free energy, calculated using the FDPB approach, ΔG_{es} is the electrostatic energy, and ΔG_{cav} is the cavitation free energy.

Substituting equation (2) into equation (1) gives:

$$(3)\quad \begin{aligned}\Delta G_{bind} &= \Delta G_{vdw,complex} + \Delta G_{es,complex} + \Delta G_{solv,complex} + \Delta G_{cav,complex} \\ &\quad - \Delta G_{vdw,prot1+prot2} - \Delta G_{es,prot1+prot2} - \Delta G_{solv,prot1+prot2} - \Delta G_{cav,prot1+prot2}\end{aligned}$$

In the final form for comparing protein-protein docking poses, we approximate that unbound proteins are in the same conformation (or energetically comparable conformations) as the bound state conformation. In this case the *intra*molecular components of the van der Waals energy and electrostatic energies for the complex cancels the contribution from the unbound proteins, and we calculate only the *inter*molecular energy terms—$\Delta G_{vdw,interact}$ and $\Delta G_{es,interact}$ (a practice commonly used in scoring functions for ligand-protein docking calculations). Note: we neglect $\Delta G_{cav,\ prot1+prot2,}$ as this cancels when comparing different orientations of the same two proteins.

$$(4)\quad \Delta G_{bind} = \Delta G_{vdw,interact} + \Delta G_{es,interact} + \Delta G_{solv,complex} + \Delta G_{cav,complex} - \Delta G_{solv,prot1+prot2}$$

Equation (4) can be used to compare multiple different docked poses of two proteins and discern which pose is most likely to be the actual biological one, assuming that the intramolecular interactions of the docked poses have previously been optimized. When optimizing a single docked pose of two proteins, we need to include the *intra*molecular terms for the flexible portions of the two proteins (e.g. for the sidechains that are allowed to rotate during the optimization) $\Delta G_{vdw,flex\ and}$ $\Delta G_{es,flex}$.

$$(5)\quad \begin{aligned}\Delta G_{bind} &= \Delta G_{vdw,interact} + \Delta G_{es,interact} + \Delta G_{solv,complex} + \Delta G_{cav,complex} - \Delta G_{solv,prot1+prot2} + \\ &\quad + \Delta G_{vdw,flex} + \Delta G_{es,flex}\end{aligned}$$

The van der Waals parameters and radii and the partial charges for the coordinates are from the parm96 set of the Amber8 force-field [40, 47], using united atom parameters for the carbon atoms from the DOCK4.0 parameter set [43]. "United charges" were generated for the non-polar hydrogens by adding their partial charges back into their attached carbon atoms. United atom parameters were used because we have found in practice with ligand-protein docking that these parameters "soften" the scoring function, so that it converges more easily and consistently upon minimization. We used the same charges and radii for the calculation of the FDPB solvation term, which was calculated by calling the "zap_binding()" macro from the ZAP library, interfaced to the PDock program. We calculated the cavitation term separately from the ZAP PB term, using the SURF program [48]. The dielectric was set to ZAP defaults—2.0 inside the proteins and 78.5 outside.

Finally, when minimizing a sidechain conformation or protein pose, for scoring with equations (4) or (5), we use a fast evaluation function where we replace the solvation and electrostatic terms with a electrostatic term with a distance-dependent dielectric of 4.0, and also calculate the *intra*molecular electrostatic and van der Waals terms for the flexible portions of the proteins.

$$(6)\quad \Delta G_{bind} = \Delta G_{vdw,\mathrm{interac}t} + \Delta G_{\mathrm{es,interact}} + \Delta G_{vdw,flex} + \Delta G_{es,flex}$$

Optimizations were performed using the coliny patternsearch optimizer [44] with a convergence set to 0.001, maximum evaluations set to 1000, and exploratory_move adaptive.

Once optimized we re-scored with equation (4) or (5) as appropriate.

Pairwise Decomposition of the Pdock Scoring Function

We developed a pairwise decomposable version of the PDock scoring function that could be used in searches of sidechain rotamers to optimize a given docking pose. The coulomb electrostatic and van der Waals terms are already pairwise, as their functions depend only on interactions between two atoms. The solvation term is not, as it is inherently multibody. Several groups have looked at how to approximate a solvation term into a pairwise function [49, 50], although in these cases they were approximating solvation with atomic surface area potentials, and developing pairwise approximations for calculating surface area, whereas our function has a poisson-boltzman term in it as well as a surface-area based cavitation term. However, many of the ideas from these pairwise decompositions can be applied to our solvation function. Both pairwise functions for calculating surface area treat surface as a single sidechain approximation that is simply a summation over all individual sidechains, plus a correction term that is a summed over sidechain pairs.

(7) $$A_{pairwise} = A_{\sin gle\_sidechain\_sum} + A_{pair\_sidechain\_sum}$$

In Street and Mayo's approach [49], the sidechain correction factor estimates the sidechain "overlap," based on a parameterized scaling factor times the difference between the surface area of sidechain pairs and the individual sidechains, all taken in the presence of the backbone.

(8a) $$A_{\sin gle\_sidechain\_sum} = \sum_i A_i$$

(8b) $$A_{pair\_sidechain\_sum} = -s\sum_i \sum_{j<i} (A_i + A_j - A_{ij})$$

Here A_i is the exposed surface area of sidechain i alone, and A_{ij} is the surface area of sidechains i and j. The double sum term is an "overlap" term, subtracting a portion of the area of overlap of the sidechains, with the parameterized scaling factor s, which had 3 terms depending on location of the sidechain in the protein. This model was improved by Zhang *et al.* [49, 50], who used "generic sidechains" to replace sidechains in all positions protein backbone outside of the rotamer i (or i,j for pair terms) in the calculation. The generic sidechains consisted of either 1 or 3 spheres with a parameterized location and radius. They used the same single sidechain approximation as 7a (with generic sidechains at all other positions), and a pairwise correction term as:

(9) $$A_{pair\_sidechain\_sum} = -s\sum_i \sum_{j \neq i} (A_i - A_{ij})$$

which represents the difference in area of sidechain i in the presence of j versus in the presence of a generic sidechain. They parameterized the scaling factor and found that s=1 worked best.

For our solvation function, we approximate both the FDPB solvation term and the cavitation terms (based on surface area), and so we calculate energy (E) instead of surface area (A), but the functional form stays analogous. Instead of generic sidechains, we used alanines to replace sidechains on the protein backbone, which are similar to the single sphere that was found to be effective in the generic side chain approach. Since we are only interested in sidechains at the protein-protein interface, we only sum over sidechains at this interface, and hence only replace these interface sidechains with alanines in the protein backbone. We refer to this construct as the "scaffold." We calculate the solvation energy of the scaffold alone, the scaffold with its interface sidechains placed back individually(E_i) or in pairs (E_{ij}). Our final function has a single sidechain approximation with a pairwise correction term as follows:

$$\text{(10a)} \quad E_{\text{single\_sidechain\_sum}} = scaffold + \sum_i (E_i - scaffold)$$

$$\text{(10b)} \quad E_{pair\_sidechain\_sum} = \sum_i \sum_{j<i} (E_{ij} + scaffold - E_i - E_j)$$

We note that in this form, we can ignore the pairwise correction term for any given sidechains i and j, and it will still sum up correctly to approximate the overall solvation energy. To save calculation time, we take a quick evaluation of the interaction between sidechain rotamers at positions i and j, using the van der Waals term and an electrostatic term in a distance-dependent dielectric of 4.0, which is a faster calculation, and only calculate the pair term if the absolute value of the interaction is above a cutoff.

Results

Comparison of PDock Scoring Function to rosettaDock on Decoy Dataset

We evaluated our scoring function on the rosettaDock decoy set, focusing on the 10 pathological test cases for which rosettaDock failed to distinguish correctly docked complexes from decoys. For cases where rosettaDock performed well, our scoring function did as well, as seen in the 1ACB example in Appendix 1. In Appendix 1, (and summarized in Table 1) we plotted score versus root-mean square deviation (RMSD) for all these test cases. For cases where a scoring function is working, we expect to see a scoring "funnel" where the native and near-native decoys (rmsd<2Å) all score considerably lower than the remaining decoys.

This is the case for 1ACB, included for illustrative purposes. For the 10 pathological test cases, not only do we not see a scoring "funnel" in the rosettaDock scores, but in all but the 2KAI case, there are few or no near-native decoys generated. In the 2KAI example, there are many low-rmsd decoys, but there are several high-rmsd decoys scoring considerably lower. As this could be either a sampling problem with the algorithm, or a scoring problem, we looked at the scores for two types of native complexes – the native crystal minimized using the rosettaDock scoring function, and the native crystal structures with sidechains repacked and re-minimized 50 times, which is equivalent computationally to the decoys. Both of these are highlighted in the graphs in Appendix 1. In both sets of native complexes, the score was considerably higher (worse) than the decoys, implying the problem was primarily with the scoring scheme.

Table 1. Comparison of rosettaDock vs PDock's ability to distinguish native from decoy conformations.

Name of PDB Complex	rosettaDock Score		PDock Score
	Number decoys (rmsd>2Å) Scoring Below Native Min[1]	Number Decoys Scoring (rmsd >2 Å) Below Repacked Min[2]	Number Decoys (>2 Å) Scoring Below Native Min[1]
1ACB	0	437	0
1BVK	989	230	0
1EO8	123	222	0
1FQ1	981	979	0
1GLA	51	154	0
1JHL	265	131	0
1MDA	error	640	0
1WEJ	8	269	0
2KAI	0	566	3
2VIR	984	56	0
3HHR	55	50	0

[1]Crystal structure of complex minimized with corresponding scoring scheme.
[2]Crystal structure of complex with sidechains re-packed and minimized 50 times.

We compared this against PDock score for the native and decoys. For the PDock native, we performed minimization of the native structure to resolve any vdw clashes as described in methods for equation 6, and then re-scored with the final evaluation function (equation 4). We also re-scored all the decoys using the PDock scoring scheme in equation 4, and compared to the score of the minimized native complex structure (note: minimizing the decoys structures did not affect their PDock score, presumably as they had previously been optimized in the rosettaDock with a scoring scheme that included van der Waals and electrostatic potentials). In all but 1 case (2KAI), the native complex scored considerably lower than all the decoys (see Appendix 1). In the case of 2KAI, three decoys were found with scores lower than the native complex using the 2.0 Å cutoff (Table 1), but all three of these decoys were within 2.5 Å of the native score. And the plot rms vs PDock score for 2KAI does show a scoring "funnel," albeit shallower than many of the other test cases. We note that the positive scores for the PDock score are because we are not subtracting the cavitation term for the unbound proteins ($\Delta G_{cav,\, prot1+prot2}$), as it should be the same for all poses.

Validation of Pairwise Decomposition Scheme

To validate the pairwise decomposition of our scoring scheme, we looked at a number of protein complexes and calculated the pairwise solvation score using equation 10 summed over all interface sidechains, versus the solvation portion of the PDock score calculated for the entire complex at once. We also looked at the sensitivity of this formulation to optimization functions.

Specifically, we optimized the sidechain rotamers, simultaneously in the case of the entire complex, and individually in the case of the pairwise score, allowing each sidechain torsional angle to rotate a maximum of ± 10 degrees. Optimizations were performed using the coliny patternsearch optimizer [44] with a convergence set to 0.001, maximum evaluations set to 1000. The results are summarized in Tables 2-4. In Table 2 we show the errors in approximating the solvation score as a simple addition of single sidechain scores (equation 10a only). In Table 3 we show how the pair term (adding equation 10b to 10a), corrects most of this error. In Table 4 we test compare the scores while allowing the optimization of the sidechain torsions.

Table 2: Comparison of PDock score vs. summation of single approximation of PDock solvation score with no (pairwise) correction term.

Name of PDB Complex	PDock Solvation Score	Summation of Single PDock Solvation Scores	Absolute Difference
1BVK	378.8	377.9	1.0
1EO8	494.0	494.5	0.4
1FQ1	571.8	571.9	0.1
1GLA	631.4	631.5	0.0
1JHL	368.2	369.0	0.9
1MDA	627.0	626.0	1.0
1MLC	353.6	356.4	2.8
1WEJ	370.2	379.0	8.8
2KAI	307.6	307.5	0.1
2VIR	496.9	495.0	1.9
3HHR	467.9	464.7	3.2

Table 3. Comparison of PDock score vs. summation of pairwise approximation of PDock solvation score with correction term.

Name of PDB Complex	PDock Solvation Score	Summation of Pairwise PDock Solvation Scores	Absolute Difference
1ACB	319.9	319.9	0.0
1BVK	378.8	378.2	0.6
1EO8	494.0	493.3	0.8
1FQ1	571.8	571.7	0.0
1GLA	631.4	631.3	0.1
1JHL	368.2	368.4	0.2
1MDA	627.0	627.5	0.5
1MLC	353.6	351.6	2.1
1WEJ	370.2	369.5	0.7
2KAI	307.6	306.0	1.5
2VIR	496.9	496.9	0.0
3HHR	467.9	467.4	0.5

Table 4. Comparison of PDock score vs pairwise PDock score with torsional optimization of sidechains. Sidechains allowed to rotate ±10 degrees, optimization procedure as described for equation 6 in methods.

Name of PDB Complex	PDock Solvation Score	Summation of Pairwise PDock Solvation Scores	Absolute Difference
1ACB	319.6	320.5	0.9
1BVK	373.6	371.8	1.9
1EO8	492.1	489.5	2.6
1FQ1	569.7	567.4	2.3
1GLA	626.3	626.2	0.1
1JHL	362.8	363.7	0.9
1MDA	625.2	625.2	0.1
1MLC	342.8	343.5	0.7
1WEJ	359.5	362.2	2.7
2KAI	305.1	304.0	1.0
2VIR	490.1	490.1	0.1
3HHR	455.3	453.2	2.1

Discussion

That the PDock scoring function performed much better than the rosettaDock function is striking, especially since the PDock function is a first-principal based scoring scheme, and the rosettaDock one is based on logistical fitting of a parameterized function to a variety of complexes, including 38 out of 54 of the very complexes used in their study! Additionally, the PDock scoring scheme uses a simplified set of only 4 unique terms, whereas the rosettaDock function includes 11 functional terms.

Not only does PDock distinguish native from decoy complexes, but in most cases it strongly distinguishes them with a large relative difference in energies to the next lowest non-nativelike (i.e. rmsd > 2.0Å) decoy energy value (see Appendix 1). To evaluate whether the very low scores PDock generates for the native structures are an artifact due to starting with x-ray crystal structures, we looked at the values in the individual terms of the scoring function of the native versus the decoys. We looked to see if any term, in particular the van der Waals, was out of proportion in the crystal complexes versus the decoys, which could happen if the rosettaDocking procedure did not sufficiently optimize for van der Waals interactions. We did not find any single term dominating the energy values, or outside the range of what is found in the decoy set (i.e. there were many decoys with van der Waals score equal or better than the crystal score). Nor could any single term discern correctly docked from mis-docked structures by itself, such as van der Waals alone or electrostatics alone. Further, in cases where rosettaDock did generate near native structures, such as 1ACB, we found that some of these structures have a lower PDock score than the minimized crystal complex structure, implying there is no bias in the score for the crystal complex itself.

We believe the two factors that help generate such a strong predictive function in PDock are the use of a highly accurate FDPB function for evaluation of solvation terms, and the use of "united atom" parameters for the vdw portion of the scoring scheme, which softens the interaction boundaries, avoiding a lot of noise in the vdw potential during minimization. We realize that this is not a complete demonstration for the use of the PDock scoring scheme in generating docking poses, for although we have demonstrated we can distinguish incorrect poses generated from the rosettaDock function, it is possible the PDock function could generate additional mis-docked poses that would be difficult to distinguish from the native poses. However, we do believe that because of the overwhelming ability of this function to correctly discern docked from mis-docked poses, at the very least the PDock scoring scheme could be used as a complementary post-evaluation function of docked poses generated from rosettaDock, and may possibly be useful as a replacement in the sidechain refinement module.

For using our scoring function in refining docked structures with flexible sidechains and/or backbone structures, we would have to include the *intra*molecular van der Waals and electrostatic terms for the flexible portions of the proteins, at least to minimize and identify low energy conformations of sidechain rotamers. Although we tested the scoring scheme without including these intramolecular terms (equation 4) in discerning docked from "decoy" structures, we note that these poses already had been optimized for intramolecular energy, as the rosettaDocking function included intramolecular terms in its minimization procedure. The scoring function in equation 5, which includes intramolecular terms, can be used with many sidechain optimization algorithms.

We also developed a pair-wise approximation to our scoring function (equation 10) that can be used with more exhaustive sidechain rotamer search strategies, such as integer programming and dead-end elimination techniques. We validated the pair-wise approximation for the solvation score, by comparing the pair-wise score summed over the entire protein-protein interface, to the solvation PDock score on the entire complex. (We did not need to validate the vdw and coulomb components, as their functions are mathematically pair-wise). We tested the summation of individual sidechain contributions to solvation score (equation 10a alone) in Table 2, and saw that our pairwise correction (equation 10b added to 10a) improves the error in this approximation considerably.

We showed that our pair-wise approximation for solvation is reasonably robust to allowing small torsional optimization of the sidechain rotamers, increasing the search space covered using discrete sidechain rotamers to ±10 degrees for each torsional angle in that rotamer. As the PDock scoring function has been shown to be very predictive as a post-filter of docking complexes, the next step is to test its ability to within a docking protocol, in particular in sidechain refinement using either the functional form with intramolecular terms (equation 5) or the pair-wise functional form (equation 10a,10b) as appropriate for the rotamer search strategy employed.

This page intentionally left blank.

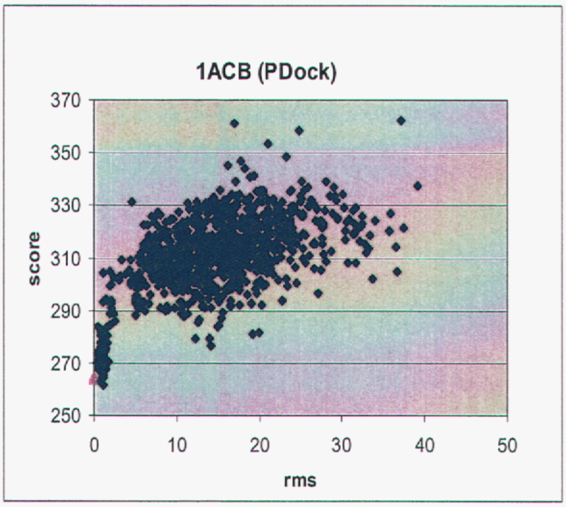
References

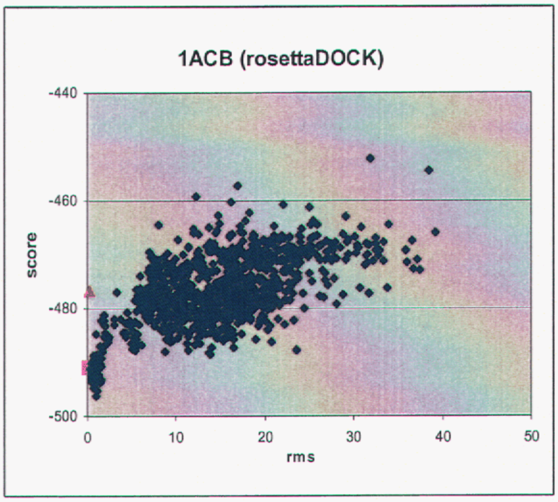
1. Mendez, R., et al. Proteins, 2003. **52**(1): p. 51-67.
2. Berman, H.M., et al. Nucleic Acids Research, 2000. **28**: p. 235-242.
3. Dunbrack, R.L. and M. Karplus. Journal of Molecular Biology, 1993. **230**(2): p. 543-574.
4. Kingsford, C.L., B. Chazelle, and M. Singh. Bioinformatics (Oxford), 2005. **21**(7): p. 1028-1036.
5. Martin, S., D. Roe, and J.L. Faulon. Bioinformatics (Oxford), 2005. **21**(2): p. 218-226.
6. Smola,A.J. and Schölkopf,B. (1998) A tutorial on support vector regression. *NeuroCOLT Technical Report NC-TR-98 030*, Royal Holloway College University of London, UK.
7. Vapnik,V. (1998) Statistical Learning Theory. Wiley Interscience, New York.
8. Furey, T.S., et al. Bioinformatics (Oxford), 2000. **16**(10): p. 906-914.
9. LESLIE, C. **15**: p. 1441.
10. Bock, J.R. and D.A. Gough. Bioinformatics (Oxford), 2001. **17**(5): p. 455-460.
11. Burges,C.J.C. and Smola,A.J. (eds), *Advances in Kernel Methods–Support Vector Learning*. MIT Press, Cambridge, MA, pp. 169–184.
12. Cristianini,N. and Shawe-Taylor,J. (2000) An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, UK.
13. Bennett,K.P. and Campbell,C. (2000) Support vector machines: hypeor hallelujah. *ACM SIGKDD Explorations*, 2, 1-13.
14. Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C.J.C. and Smola,A.J. (eds), *Advances in Kernel Methods–Support Vector Learning*. MIT Press, Cambridge, MA, pp. 169–184.
15. Visco, D.P., et al. Journal of Molecular Graphics & Modelling
221st National Meeting of the American-Chemical-Society; April 1-5, 2001; San Diego, California, 2002. **20**(6): p. 429-38.
16. Faulon, J.L., D.P. Visco, and R.S. Pophale. Journal of Chemical Information and Computer Sciences, 2003. **43**(3): p. 707-20.
17. Faulon, J.L., C.J. Churchwell, and D.P. Visco. Journal of Chemical Information and Computer Sciences, 2003. **43**(3): p. 721-34.
18. Churchwell,C.J., Rintoul,M.D., Martin,S., Visco,D., Kotu,A., Larson,R.S., Sillerud,L.O., Brown,D.C. and Faulon,J.L. (2004) The signature molecular descriptor. 3. Inverse quantitative structure–activity relationship of ICAM-1 inhibitory peptides. *J. Mol. Graph. Model.*
19. Fernandez-Recio, J., M. Totrov, and R. Abagyan. Proteins, 2003. **52**(1): p. 113-117.
20. Gray, J.J., et al. Journal of Molecular Biology, 2003. **331**(1): p. 281-299.
21. Camacho, C.J. and D.W. Gatchell. Proteins, 2003. **52**(1): p. 92-97.
22. Fernandez-Recio, J., M. Totrov, and R. Abagyan. Journal of Molecular Biology, 2004. **335**(3): p. 843-865.
23. Li, L., R. Chen, and Z.P. Weng. Proteins, 2003. **53**(3): p. 693-707.
24. Smith, G.R., M.J.E. Sternberg, and P.A. Bates. Journal of Molecular Biology, 2005. **347**(5): p. 1077-1101.
25. Schneidman-Duhovny, D., et al. Proteins Structure Function and Bioinformatics, 2005. **60**(2): p. 224-231.

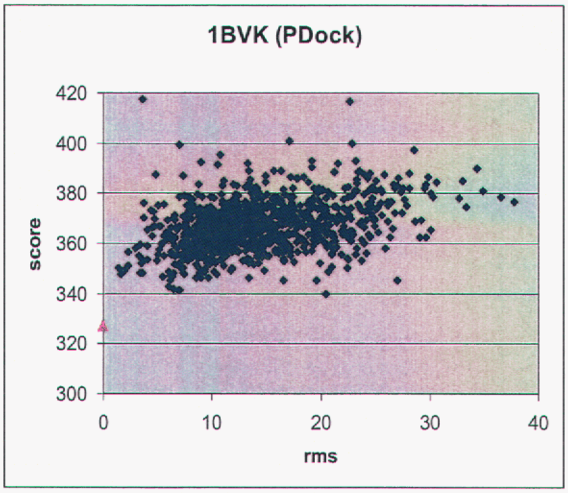
26. Wang, C., O. Schueler-Furman, and D. Baker. Protein Science, 2005. **14**(5): p. 1328-1339.
27. Mendez, R., et al. Proteins Structure Function and Bioinformatics, 2005. **60**(2): p. 150-169.
28. Murphy, J., et al. Proteins, 2003. **53**(4): p. 840-854.
29. Bradford, J.R. and D.R. Westhead. Bioinformatics (Oxford), 2005. **21**(8): p. 1487-1494.
30. Fernandez-Recio, J., et al. Proteins Structure Function and Bioinformatics, 2005. **58**(1): p. 134-143.
31. Bordner, A.J. and R. Abagyan. Proteins Structure Function and Bioinformatics, 2005. **60**(3): p. 353-366.
32. Chen, R., et al. Proteins, 2003. **52**(1): p. 88-91.
33. Jiang, L., et al. Proteins, 2002. **46**(2): p. 190-196.
34. Zhang, C., et al. Journal of Medicinal Chemistry, 2005. **48**(7): p. 2325-2335.
35. Moont, G., H.A. Gabb, and M.J.E. Sternberg. Proteins, 1999. **35**(3): p. 364-373.
36. Mendes, J., et al. Journal of Computer-Aided Molecular Design, 2001. **15**(8): p. 721-740.
37. Liang, S.D. and N.V. Grishin. Proteins Structure Function and Bioinformatics, 2004. **54**(2): p. 271-281.
38. Camacho, C.J., et al. Proteins, 2000. **40**(3): p. 525-537.
39. Nicholls, A., *Zap Toolkit*, Openeye Scientific Software.
40. Cornell, W.D., et al. Journal of the American Chemical Society, 1995. **117**(19): p. 5179-5197.
41. Grant, J.A., B.T. Pickup, and A. Nicholls. Journal of Computational Chemistry, 2001. **22**(6): p. 608-40.
42. Prabhu, N.V., P.J. Zhu, and K.A. Sharp. Journal of Computational Chemistry, 2004. **25**(16): p. 2049-64.
43. Ewing, T.J.A., et al. Journal of Computer-Aided Molecular Design, 2001. **15**(5): p. 411-428.
44. Hart, W.E., *The Coliny Optimization Library*. 2004.
45. Voigt, C.A., D.B. Gordon, and S.L. Mayo. Journal of Molecular Biology, 2000. **299**(3): p. 789-803.
46. PICO. *http://www.cs.sandia.gov/~caphill/proj/pico.html*
47. Amber8. *http://amber.scripps.edu*
48. Varshney, A. and F.J. Brooks, *Fast Analytical Computation of Richard's Smooth Molecular Surface*. 1993, UNC at Chapel Hill: Chapel Hill, NC.
49. Street, A.G. and S.L. Mayo. Folding & Design, 1998. **3**(4): p. 253-258.
50. Zhang, N.G., C. Zeng, and N.S. Wingreen. Proteins Structure Function and Bioinformatics, 2004. **57**(3): p. 565-576.
51. Gray, J.J. *http://graylab.jhu.edu/docking/decoys/*

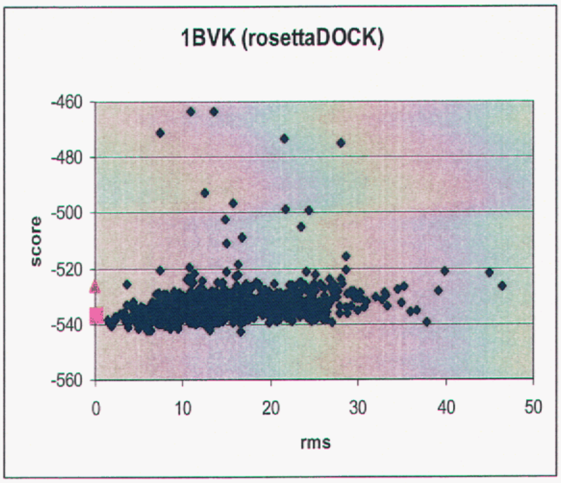
Appendix 1: Comparison of PDock Score Versus rosettaDock Score for 11 Protein-Protein Complexes

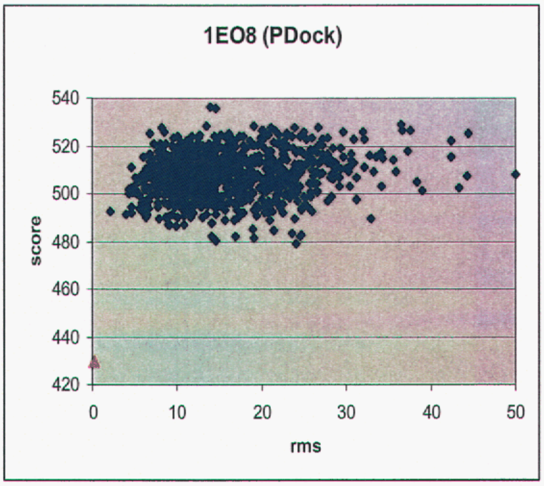
Here is a complete plot of the set of 10 pathological complexes from the rosettaDock decoy collection, plus one correctly predicted complex (1ACD) for illustration, comparing rosettaDock scoring scheme with the PDock scoring scheme. The RosettaDock scores and were taken directly from the web site [51]. For each complex we plot the score vs the rmsd of that decoy from the native complex. The title of each complex is the name of the complex in PDB [2]. The scores for the crystallographic complexes are highlighted as follows: red diamonds for the minimized native complex, and magenta square [rosettaDock score only] for the minimized native with sidechains repacked and minimized 50 times using the rosettaDock protocol. We note that the positive scores for the PDock score are because we are not subtracting the cavitation term for the unbound proteins, as it is the same for all poses. As seen in these plots, the PDock score always has the lowest score for the minimized native complex (or close native-like "decoys"), whereas the rosettaDock scoring function is unable to distinguish the crystallographic conformation from the decoys in these cases.

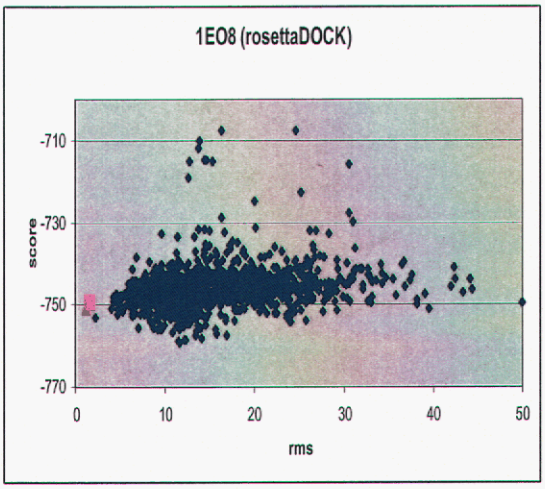


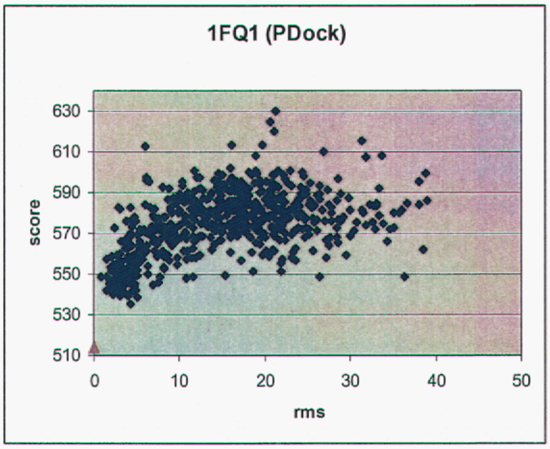


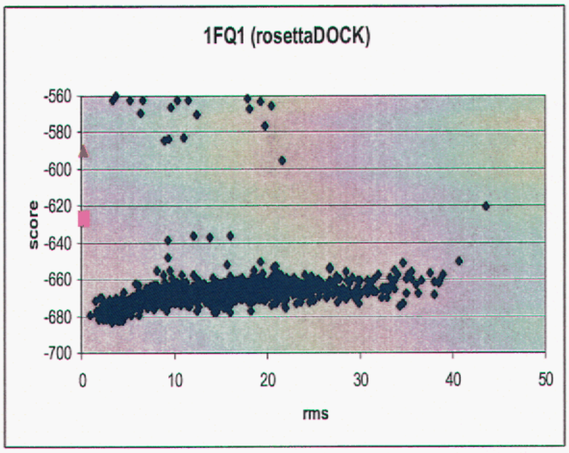


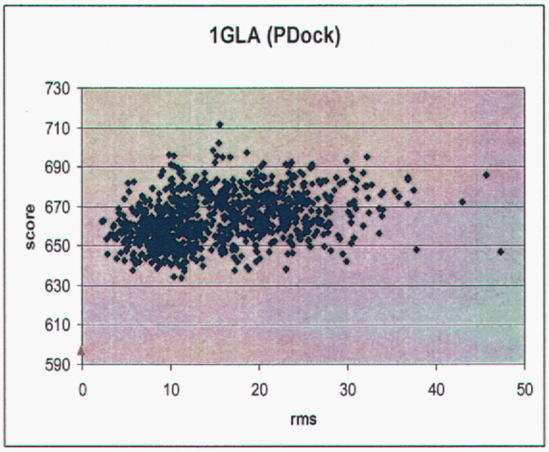


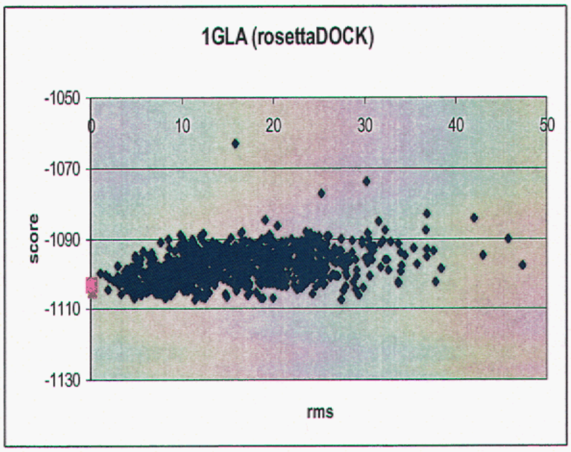

1EO8 (PDock)
score
540
520
500
480
460
440
420
0
10
20
30
40
50
rms

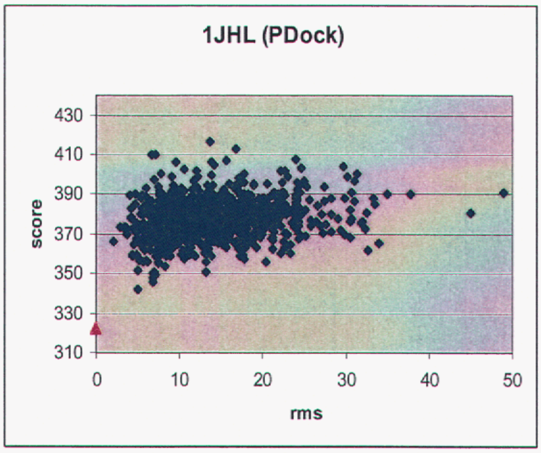

1EO8 (rosettaDOCK)
score
-710
-730
-750
-770
0
10
20
30
40
50
rms

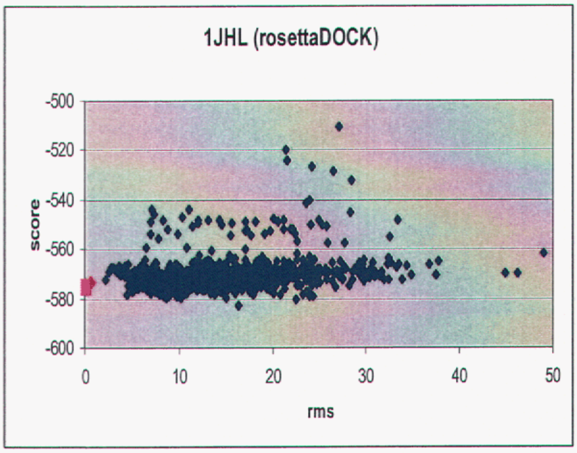

1FQ1 (PDock)
score
630
610
590
570
550
530
510
0
10
20
30
40
50
rms

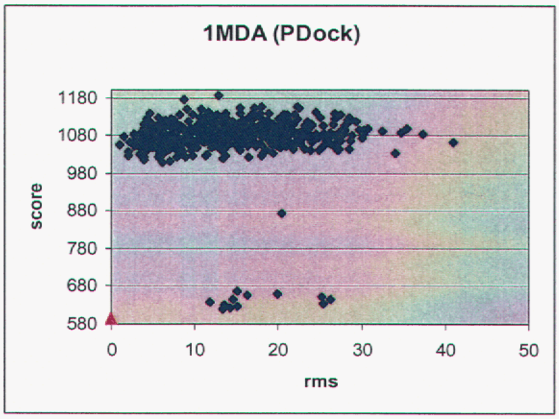

1FQ1 (rosettaDOCK)
score
-560
-580
-600
-620
-640
-660
-680
-700
0
10
20
30
40
50
rms

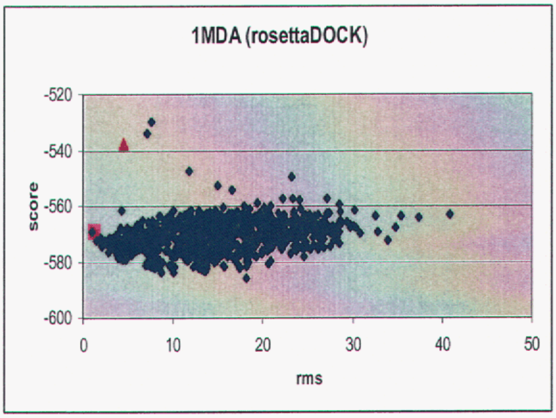

1GLA (PDock)
score
730
710
690
670
650
630
610
590
0
10
20
30
40
50
rms

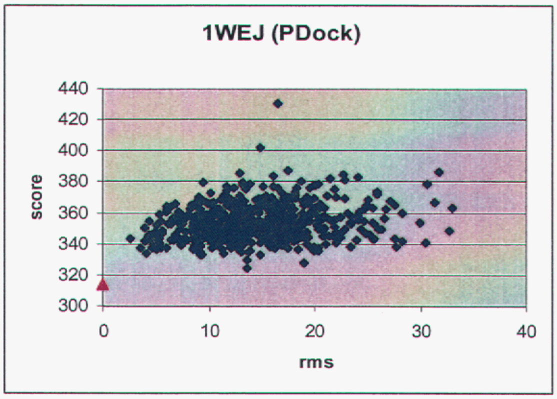

1GLA (rosettaDOCK)
score
-1050
-1070
-1090
-1110
-1130
0
10
20
30
40
50
rms

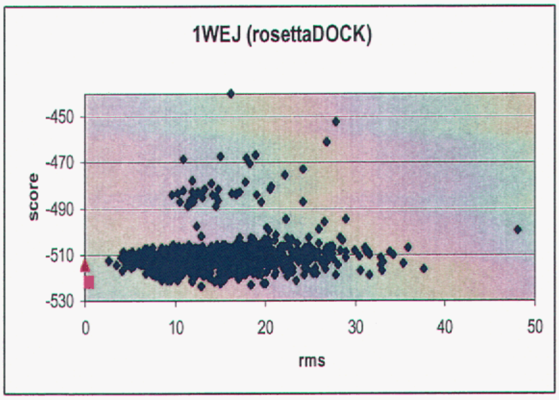

1JHL (PDock)
score
430
410
390
370
350
330
310
0
10
20
30
40
50
rms

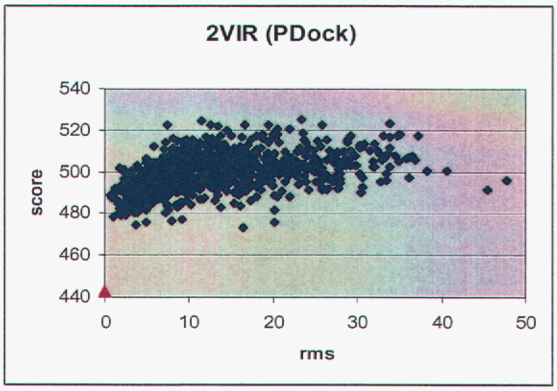

1JHL (rosettaDOCK)
score
-500
-520
-540
-560
-580
-600
0
10
20
30
40
50
rms

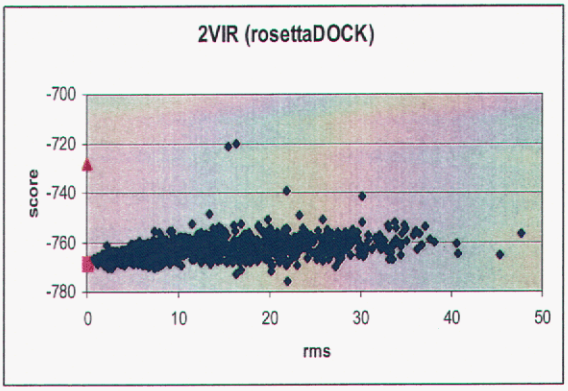

1MDA (PDock)
score
1180
1080
980
880
780
680
580
0
10
20
30
40
50
rms

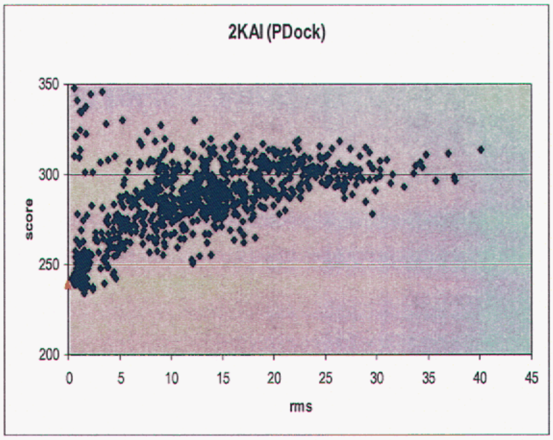

1MDA (rosettaDOCK)
score
-520
-540
-560
-580
-600
0
10
20
30
40
50
rms

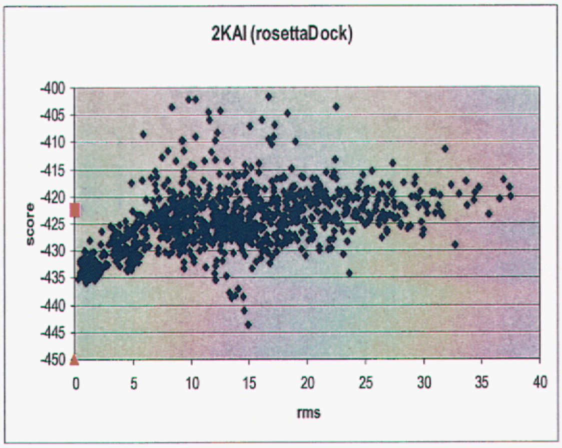

1WEJ (PDock)
score
440
420
400
380
360
340
320
300
0
10
20
30
40
rms

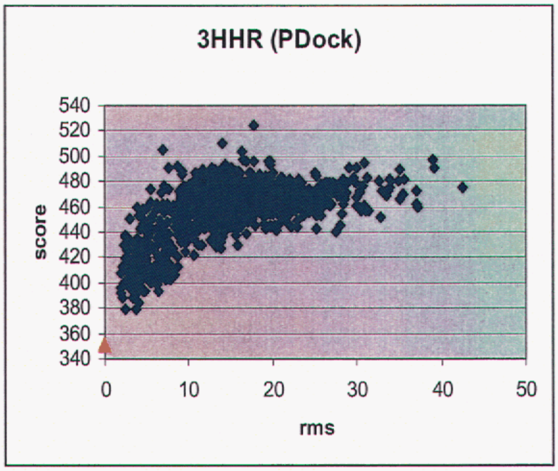

1WEJ (rosettaDOCK)
score
-450
-470
-490
-510
-530
0
10
20
30
40
50
rms

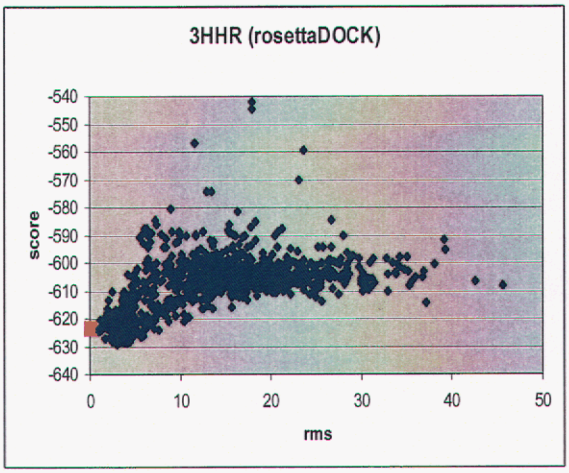

2VIR (PDock)
score
540
520
500
480
460
440
0
10
20
30
40
50
rms


2VIR (rosettaDOCK)
score
-700
-720
-740
-760
-780
0
10
20
30
40
50
rms


2KAI (PDock)
score
350
300
250
200
0
5
10
15
20
25
30
35
40
45
rms


2KAI (rosettaDock)
score
-400
-405
-410
-415
-420
-425
-430
-435
-440
-445
-450
0
5
10
15
20
25
30
35
40
rms


3HHR (PDock)
score
540
520
500
480
460
440
420
400
380
360
340
0
10
20
30
40
50
rms


3HHR (rosettaDOCK)
score
-540
-550
-560
-570
-580
-590
-600
-610
-620
-630
-640
0
10
20
30
40
50
rms

Distribution

1	MS 0310	Shawn Martin, 1412
1	MS 0672	Lyndon Pierson, 5616
1	MS 1110	William Hart, 1415
1	MS 1413	Jean-Loup Faulon, 8333
2	MS 9292	Diana Roe, 8321
1	MS 9292	Ken Sale, 8321
1	MS 9292	Malin Young, 8321
2	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0323	Donna Chavez, LDRD office, 1030