JOURNAL OF
CHEMICAL INFORMATION
AND MODELING

ARTICLE

pubs.acs.org/jcim

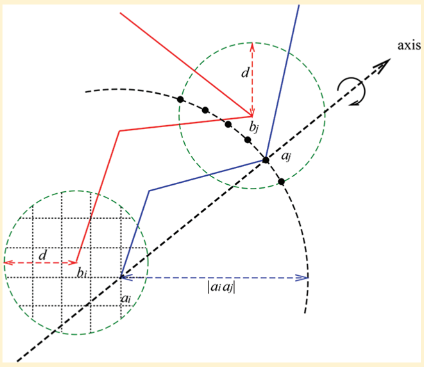
Protein−Protein Binding Sites Prediction by 3D Structural Similarities

Fei Guo,[†] Shuai Cheng Li,[‡] and Lusheng Wang*,[‡]

[†]School of Computer Science and Technology, Shandong University, Jinan 250101, Shandong, China

[‡]Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

ABSTRACT: Identifying the location of binding sites on proteins is of fundamental importance for a wide range of applications including molecular docking, de novo drug design, structure identification, and comparison of functional sites. In this paper, we develop an efficient approach for finding binding sites between proteins. Our approach consists of four steps: local sequence alignment, protein surface detection, 3D structure comparison, and candidate binding site selection. A comparison of our method with the LSA algorithm shows that the binding sites predicted by our method are somewhat closer to the actual binding sites in the protein−protein complexes. The software package is available at http://sites.google.com/site/guofeics/pro-bs for noncommercial use.



INTRODUCTION

Identifying the location of binding sites on proteins is of fundamental importance for a wide range of applications including molecular docking, de novo drug design, structure identification, and comparison of functional sites. Identifying interface between two interacting proteins provides important clues to the function of a protein and can reduce the search space required by docking algorithms to predict the structures of complexes. Here we develop a software package for identifying the binding sites between proteins.

Many methods have been proposed for identifying the location of binding sites on proteins. Bradford and Westhead[1] combine a support vector machine (SVM) approach with surface patch analysis to predict protein−protein binding sites. Chen et al.[2] develop a tool, 3D-partner, for inferring interacting partners and binding models. 3D-partner first utilizes IMPALA to identify homologous structures (templates) of a query protein sequence from a heterodimer profile library. The sequence profiles of those templates are then used to search for interacting candidates of the query from protein sequence databases by PSI-BLAST. Hsu et al.[3] develop a method for predicting helix−helix interaction from residue contacts in membrane proteins. They first predict contact residues from sequences. Their relationships are further predicted in the second step via statistical analysis on contact propensities and sequence and structural information. Li et al.[4] propose an approach for finding binding sites for groups of proteins. It contains the following steps: finding protein groups as bicliques of protein−protein interaction networks (PPI), identifying conserved motifs, and searching domain−domain interaction databases. Liu and Wang[5] extend the method of Li et al.[4] and consider comparing 3D local structures.

The LSA algorithm[6] uses the local 3D structure similarity to find binding sites in protein complexes. The algorithm first extracts the solvent accessible surface residues/atoms from the protein structures and constructs an undirected graph according to the distances between residues in the proteins. The problem of finding binding sites is transformed into the problem of finding maximum cliques on the graph.[7] Experiments show that this approach works reasonably well.[8,9]

In this paper, we develop an efficient approach for predicting protein−protein binding sites by directly comparing the 3D (three-dimensional) structures of the binding sites. If two proteins are bound together, the structures of them are complementary, and each atom of one binding site is geometrically near to some atoms of its binding partner. Therefore, the pairs of binding sites on the interface have complementary substructures. We predict the protein−protein binding sites via finding the complementary 3D substructures between pair of proteins.

We design a method which consists of four steps: local sequence alignment, protein surface detection, 3D structure comparison, and candidate binding site selection using a unified transformation. All four steps are served to finding a pair of candidate sites, where one site from each binding partner, which have complementary 3D structures. Experiments show that our method works well in practice.

METHODS

Given two protein structures, our task is to find the binding sites between proteins. Our method contains four steps. Step 1: we do local sequence alignment at the atomic level to get the aligned segments, which contain some gaps. These aligned *segments* (pairs of subsequences) may or may not have similar 3D structures. However, the unaligned segments are unlikely to

Received: May 9, 2011
Published: November 12, 2011

have similar 3D structures. The subsequent steps are to further prune the unlikely candidates from the results of step one. It is observed that most of the atoms on the binding sites are at the surface of protein. Hence, we have designed Step 2. Step 2: among the aligned segments obtained in Step 1, we filter out all aligned *subsegments* (part of aligned segment), which have no more than 2/3 of its atoms on the surface. The unfiltered subsegments are considered as the *surface* subsegments. In addition, we require that each subsegment has at least 15 atoms. The length of 15 atoms comes from observations with the data set of 24 complexes, where all the segments in the actual binding sites have more than 15 atoms. Step 3: for the unfiltered subsegments from Step 2, we perform rigid transformations on them. If there exist transformations such that at least 2/3 of atom pairs in the subsegments are within distance d (a given parameter), we keep these subsegments with similar 3D structures for further process. When performing the rigid transformations, we ensure that the protein 3D structures have no overlaps. This step will output a set of candidate sites; each of them is associated with a rigid transformation. Step 4: we select a rigid transformation obtained in Step 3 that can "match" the maximum number of candidate binding sites. Details of those four steps are presented in the following subsections.

Step 1: Local Sequence Alignment. The method begins with two proteins and compares the proteins at the sequence level. In PDB format files, each residue (amino acids) is represented in the traditional order of atom records N, CA, C, O, followed by the side chain atoms (CB, CG1, CG2 ...) in order first of increasing remoteness and then branch.[10] The whole protein sequence of residues can be translated into a sequence of atoms based on this representation.

In the results, we show that the numbers of residues in the pairs of binding sites are similar. This step is to discover the pairs of subsequences, one subsequence from each protein, such that two subsequences contain almost the same number of atoms. In addition, we prefer that an atom can only be aligned to the same type of atoms. This is solely to reduce the search space. We can handle the case that an atom matches to the difference types of atoms, by enlarging the distance threshold d of Step 3 and Step 4.

The sequences of binding sites between two proteins are usually conserved at the atomic level. We use the standard Smith-Waterman local sequence alignment algorithm[11] to find a set of aligned segments (pairs of subsequences with high similarity). The matching scores are defined as follows. A higher score denotes a preferred match, and a lower score denotes an unpreferred match. A matched pair of same atom type contributes a score of 1; a matched pair of different atom types results in a score $-\infty$, that is, we do not allow such cases. In addition, a match between an atom and a space (or gap) has a score of -2.

We present details here. For two sequences P_1 and P_2, an alignment of P_1 and P_2 can be obtained by (1) inserting spaces into the two sequences P_1 and P_2 such that the two resulting sequences with inserted spaces P_1' and P_2' have the same length and (2) overlap the two resulting sequences P_1' and P_2'. The score of the alignment is the sum of the scores for all the columns, where each column has a pair of letters (including spaces) and for each pair of letters there is a predefined similarity score. Note that there are many ways to insert spaces into two given sequences. The similarity between two sequences is defined as the highest score over all alignments between two sequences.

Now, we present the local sequence algorithm. The input of the algorithm consists of *two protein sequences* P_1 and P_2 and a *score scheme*.

1daa_63_A	N E20	CA E20	C E20	O E20	CB E20	CG E20	CD E20	N D21	CA D21	C D21	O D21	CB D21	CG D21		N R22
1daa_65_B	N D143	CA D143	C D143	O D143	CB D143	CG D143		N I144	CA I144	C I144	O I144	CB I144	CG I144	CD I144	N K145
1daa_63_A	CA R22	C R22	O R22	CB R22	CG R22	CD R22	CZ R22	N G23	CA G23	C G23	O G23		**N** **Y24**	**CA** **Y24**	**C** **Y24**
1daa_65_B	CA K145	C K145	O K145	CB K145	CG K145	CD K145		N S146	CA S146	C S146	O S146	CB S146	**N** **L147**	**CA** **L147**	**C** **L147**
1daa_63_A	**O** **Y24**	**CB** **Y24**	**CG** **Y24**	CD1 Y24	CD2 Y24	CE1 Y24	CE2 Y24	**N** **Q25**	**CA** **Q25**	**C** **Q25**	**O** **Q25**	**CB** **Q25**	CG Q25	CD Q25	**N** **F26**
1daa_65_B	**O** **L147**	**CB** **L147**	**CG** **L147**	CD1 L147	CD2 L147			**N** **N148**	**CA** **N148**	**C** **N148**	**O** **N148**	**CB** **N148**	CG N148		**N** **L149**
1daa_63_A	**CA** **F26**	**C** **F26**	**O** **F26**	**CB** **F26**	**CG** **F26**	**CD1** **F26**	CE1 F26	**N** **G27**	**CA** **G27**	**C** **G27**	**O** **G27**			N D28	CA D28
1daa_65_B	**CA** **L149**	**C** **L149**	**O** **L149**	**CB** **L149**	**CG** **L149**	**CD1** **L149**		**N** **L150**	**CA** **L150**	**C** **L150**	**O** **L150**	CB L150	CG L150	N G151	CA G151
1daa_63_A	C D28	O D28	CB D28	CG D28	N G29	CA G29	C G29	O G29		N V30	CA V30	C V30	O V30	CB V30	CG1 V30
1daa_65_B	C G151	O G151			N A152	CA A152	C A152	O A152	CB A152	N V153	CA V153	C V153	O V153	CB V153	CG1 V153
1daa_63_A	CG2 V30	N Y31	CA Y31	C Y31	O Y31	CB Y31	CG Y31	CD1 Y31	CD2 Y31	CE1 Y31	CE2 Y31	CZ Y31			
1daa_65_B	CG2 V153	N L154	CA L154	C L154	O L154	CB L154	CG L154	CD1 L154	CD2 L154						

Figure 1. Aligned segments by the local sequence alignment algorithm: the bold columns are the actual binding sites.

The traditional local sequence alignment algorithm finds a subsequence α of P_1 and a subsequence β of P_2 such that the similarity between α and β is maximized. Here we want to find all (nonoverlapping) pairs of subsequences with a similarity score of at least x. For our purpose, we set $x = 15$ throughout the paper.

For a sequence P, we use $P[i]$ to indicate the i-th letter of P. Let n and m be the lengths of P_1 and P_2, respectively. In the algorithm, we use an $n \times m$ matrix W, where each cell $W[i,j]$ is the highest similarity score between a subsequence of P_1 ending at position $P_1[i]$ and a subsequence of P_2 ending at position $P_2[j]$. We can compute $W[i,j]$ as follows

$$W[i,j] = \max \begin{cases} 0 \\ W[i-1,j-1] + s(P_1[i],P_2[j]) \\ W[i-1,j] + s(P_1[i],\Delta) \\ W[i,j-1] + s(\Delta,P_2[j]) \end{cases}$$

where Δ represents the space, and $s()$ is the similarity score between two letters (including possibly a space) in a column. We define the *score scheme*, as follows: (1) $s(P_1[i],\Delta) = s(\Delta,P_2[j]) = -2$; (2) if $P_1[i] = P_2[j]$, $s(P_1[i],P_2[j]) = 1$; (3) if $P_1[i] \neq P_2[j]$, $s(P_1[i],P_2[j]) = -\infty$. The initial values are $W[0,j] = 0$ and $W[i,0] = 0$ for $i = 0,1,...,n$ and $j = 0,1,...,m$. After all the cells in W are computed, we find the cell with the maximal value $W[i,j]$, and use the standard backtracking process to find the pair of subsequences α and β for P_1 and P_2, respectively. After obtaining the pair of subsequences with the highest similarity score, we can find the pair of subsequences with the second highest similarity score that has no overlap with the previous reported pair(s). We repeat this process until all the pairs with a similarity score of at least 15 are reported.

The local sequence alignment algorithm outputs a set of aligned segments. An example of the output local alignment is given in Figure 1, where the reported segment is long, and it contains the actual binding sites between two proteins (see the bold columns). Next, we will focus on the columns with identical atoms and ignore the rest of columns in the following steps. Some residues may not be resolved experimentally in proteins. We exclude the segments from our solution when the residue numbers of the atoms are not consecutive.

Step 2: Identifying Surface Atoms. The interface residues of two proteins are necessarily surface residues. Inspired by the

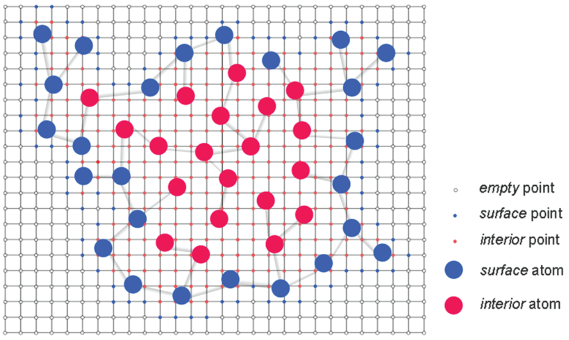


Figure 2. A 2D example of the grid points and protein atoms labeling *interior*, *surface*, or *empty*.

work in LIGSITE^{csc},[12] we propose the following method to identify the surface atoms of a protein. First, the protein is projected onto a cubic grid in the Euclidean space. The grid size is 1 Å. Second, grid points are classified as *interior*, *surface*, or *empty* points as follows: (i) A grid point is classified as a *protein* point if it is within distance 2 Å of an atom in the protein. (ii) A grid point is classified as an *empty* point if it is not a *protein* point. (iii) A *protein* grid point is classified as an *interior* grid point if all its six neighboring grid points are *protein* grid points. (iv) A *protein* grid point is classified as a *surface* grid point if at least one of its six neighboring grid points is *empty* grid point. Finally, an atom in the protein is a *surface* atom if it is within distance 1.5 Å of a *surface* grid point, otherwise it is classified as an *interior* atom. Figure 2 gives an example in 2D, where a *protein* grid point is labeled as *interior* if it has all four neighbors as *protein* grid points.

For an aligned segment output by the local sequence alignment, we consider all its subsegments containing at least 15 matched pairs of atoms. For such a subsegment, if both sequences on this subsegment have at least 2/3 of its atoms (matched and unmatched atoms) as the surface atoms, we then treat such a subsegment as a candidate site for further processing in the next step. This 2/3 is obtained from our observation of the data, and the details are presented in the Results section.

Step 3: Computing Rigid Transformations to Match Candidate Sites. For any candidate sites obtained after Step 2, we further test if the pair of 3D structures can match well on such a site. For alignment $\mathcal{A}$, we only consider the columns containing a pair of matched (identical) atoms. Precisely, we will find the set of subsegments in alignment $\mathcal{A}$ using the following rule: there exists a rigid transformation such that for at least 2/3 of atom pairs in the subsegment, the distance between each pair of (identical) atoms is at most d, where d is a parameter given by the user. This requires us to solve the following protein 3D structure matching problem:

Input: A sequence alignment $\mathcal{A}$, where each position in the alignment has two identical atoms (ignoring columns with a space), the 3D coordinate of each atom in the alignment, and a threshold d.

Goal: Find a set of subsegments of alignment $\mathcal{A}$, such that for each output subsegment there exists a rigid transformation t and at least 2/3 of atom pairs are within distance d under t.

The protein 3D structure matching problem can be solved in several ways. Here we extend the new transformation method by Guo and Wang[13] which is a faster version of the method by Li et al.[14] to solve the problem. The method can compute a rigid transformation, such that the distance between each pair of atoms is at most $(1 + \varepsilon)d$, where $\varepsilon = 0.1$ is a parameter to control the accuracy of the transformation.

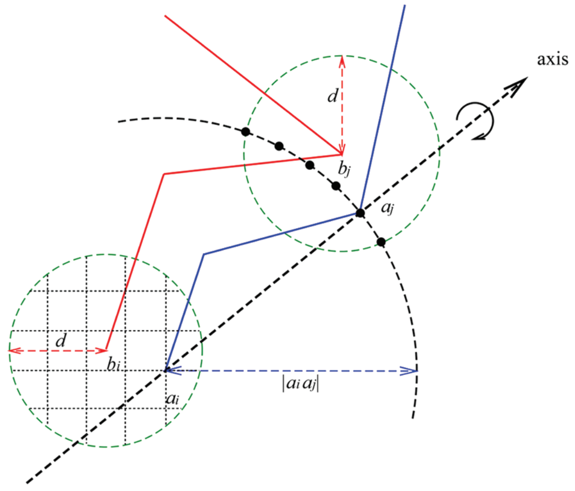


Figure 3. The steps to obtain a rigid transformation: (1) put a_i at one of the $O((1/\varepsilon)^3)$ grid points in the ball centered at b_i with radius d. (2) put a_j at a grid point on the intersection of the sphere centered at a_i with radius $|a_ia_j|$ and the ball centered at b_j with radius d. (See the dark grid points.) There are at most $O((1/\varepsilon)^2)$ grid points. (3) use a_i and a_j as the axis of rotation.

For completeness, we explain some details concerning the algorithm for solving the protein 3D structure matching problem. The task here is to find a set of subsegments of alignment $\mathcal{A}$, such that for each output subsegment there exists a rigid transformation t and at least 2/3 of atom pairs in the subsegment under t are within distance d. There are at most $|\mathcal{A}|^2$ possible subsegments. For each of the possible subsegments, we assume that we know the two ends i and j (the left most and the right most positions) of the subsegment in the alignment. Let a_i and b_i be the two atoms in column i of alignment $\mathcal{A}$ and a_j and b_j be the two atoms in the column j of $\mathcal{A}$. To see if there exists a rigid transformation t such that at least 2/3 of atom pairs in the subsegment are within distance d under t, we fix the 3D coordinates of the second protein $(b_i...b_j)$, and try to use a rigid transformation t to move the first protein $(a_i...a_j)$. There are an infinite number of possible rigid transformations. Here we use a discrete version. In each of the three directions in the 3D space, we divide the distance d into $1/\varepsilon$ units, i.e., we use a grid of size $d \times \varepsilon$ in the 3D space. We try to put a_i to a grid point within a distance at most d to b_i. There are $O((1/\varepsilon)^3)$ possible grid points within distance d to b_i. After fixing the position of a_i at a grid point, the possible position of a_j under the rigid transformation is on the intersection of the sphere centered at a_i with a radius $|a_ia_j|$ and the ball centered at b_j with radius d. See Figure 3. Again, we only consider at most $O((1/\varepsilon)^2)$ possible grid points on the sphere. After we fix the positions of a_i and a_j under t, we can use a_i and a_j as the axis to rotate the first protein and get the 3D coordinates of other points between a_i and a_j. Here we use 1° as the step size to try 360° and find a rotation such that at least 2/3 of atom pairs in the subsegment are within distance d. The Steps to find a rigid transformation are summarized in Figure 3.

Testing the Overlap of Proteins in 3D Space. When computing the rigid transformation, we also require that the proteins (as molecules) do not overlap under the rigid transformation in 3D space. For each rigid transformation that can match the 3D structures of the candidate sites, we test if the proteins have overlap in 3D space under such a transformation as follows:

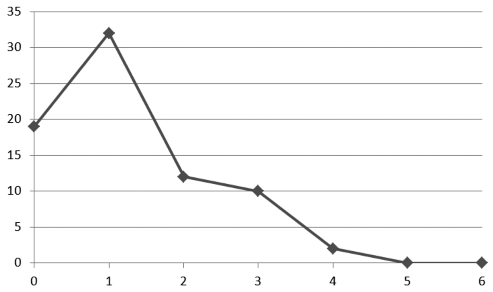


Figure 4. The size differences of binding sites vs the numbers of binding sites.

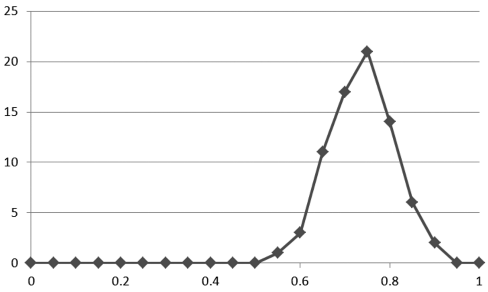


Figure 5. The percentage of surface atoms on the actual interfaces of 24 complexes.

1 Construct the grid in 3D space and mark each grid point as *interior*, *surface*, or *empty* as in Step 2 with respect to each of the given proteins.
2 Let X be the number of grid points that are *interior* points for both proteins and X_1 and X_2 be the number of *interior* points of the first protein and the second protein, respectively. If $X \leq 0.05 \times min\{X_1,X_2\}$, then we say that there is no overlap between proteins under the current rigid transformation. We treat the matched structures as candidate sites. Otherwise, we have to give up the rigid transformation.

Step 4: Identifying Candidate Binding Sites with Optimal Transformation. After Step 3, we obtain a set of matched candidate sites between two input proteins. For each candidate binding site, there is a rigid transformation. Thus, we also have a set of rigid transformations between two proteins. We have randomly selected 200 pairs of proteins in PiSite[15] (a database containing a set of actual binding sites) and find that for each pair of proteins, there exists a rigid transformation that can fit all the binding residues between two proteins listed in PiSite. Thus, we know that in most cases for any pair of proteins, there is a unified rigid transformation that fits all pairs of binding sites between them. Our last step is to select a rigid transformation that can "match" the largest number of candidate pairs.

For each obtained rigid transformation *t*, we test if other candidate binding sites can also be "matched" under *t*. A candidate binding site can be *matched* under *t* if at least 1/2 of atom pairs in such a site are within distance *d* (here we set *d* = 2.0 Å). We select the rigid transformation that can match the largest number of candidate binding sites and report the corresponding set of binding sites. We apply special treatments for the following two cases:

Case 1: When any pair of candidate sites obtained by Step 3 cannot be matched under a *unified* transformation, we know that only one of them is the actual binding site. We choose one site with the smallest number of mismatch atom pairs. Here we do not choose the site with the largest number of matched atom pairs since such a subsegment may be long and may contain many spaces.

Case 2: When the number of candidate binding sites is greater than five such that these sites are matched under a *unified* transformation. However, some of them must not be the actual binding sites. Therefore, we must choose some sites more likely to be actual binding sites. In this case, we decrease the value of *d* from 2.0 Å to 1.5 Å and further select the candidate binding sites in which at least 1/2 of atom pairs are within distance 1.5 Å under a *unified* rigid transformation.

Data Set. We use the data set used in the LSA algorithm[6] that contains 24 biologically relevant protein−protein complexes. The 3D structures of these complexes can be found in the PDB database.[10] These protein complexes are divided into two parts: 16 transient chains of complexes and eight mixed types of complexes. The mixed types of complexes consist of four heterodimer chains, two homodimer chains and two transient chains of complexes. Each complex in the data set can be split into its constituent chains. We then predict binding sites between one of the chains and other chains in the complexes.

The binding sites in experimental complex structures are the *actual* binding site on the protein−protein interface. We use exactly the same method as in LSA.[6,16] Here two residues in a pair of proteins are called *interface residues* if any two atoms, one from each residue, interact. By interact, we mean the distance between the two atoms is less than the sum of the van der Waals radius of the two atoms plus 3.0 Å.

RESULTS

Our method is implemented in a program called Pro-BS. Two measures are utilized to assess the performance of Pro-BS. *Specificity* and *sensitivity* are two common measures to assess the quality of the binding sites in the LSA algorithm.[6] *Specificity* indicates the proportion of the predicted residues that are also the interface residues and is defined as the number of predicted residues in the actual interface divided by the number of predicted residues. *Sensitivity* is the proportion of the interface residues that are predicted and is defined as the number of predicted residues in the actual interface divided by the number of actual interface residues.

Number of Residues in the Binding Sites. The local sequence alignment in our method identifies the aligned segments, such that each pair contains almost the same number of atoms. It is reasonable that each site and its binding partner are similar in size. We count the numbers of residues in pairs of actual binding sites. For 24 complexes, the actual protein−protein interfaces can be divided into 75 pairs of sequence segments. Each segment contains the interface residues of the consecutive residue numbers. For each pair of sequence segments, *difference* between the sizes (the numbers of residues) of one sequence segment and its binding partner is at most four. The distribution of these differences on the actual interfaces of 24 complexes is shown in Figure 4. We can see that the numbers of residues in pairs of binding sites are similar.

Surface Atoms in the Binding Sites. In Step 2 of our method, we filter out a segment which has no more than 2/3 of atoms as surface atoms. The choice of 2/3 is from the observation of actual data. For the 24 complexes, 72.5% of atoms of the actual protein−protein interfaces are surface atoms. The actual interfaces of 24 complexes contain 75 sequence segments, and 60 of

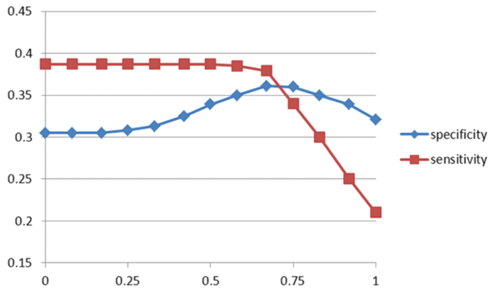


Figure 6. The specificities and sensitivities of 24 complexes for Pro-BS with different threshold values in Step 2.

Table 1. Values of Specificity and Sensitivity with and without Surface Detection on 24 Complexes

	Pro-BS without surface		Pro-BS with surface	
	Spec[a]	Sens[b]	Spec	Sens
transient complexes	26.3	32.9	32.6	31.8
mixed complexes	37.9	51.5	43.0	50.2

[a] Spec (%) is the average specificity on the data set. [b] Sens (%) is the average sensitivity on the data set.

Table 2. Numbers of Predicted Residues of 24 Complexes for Pro-BS at the Four Different Values of *d*

	Pro-BS				
	d = 1.5 Å	*d* = 2.0 Å	*d* = 2.5 Å	*d* = 3.0 Å	LSA
mean	33.7	51.5	76.2	89.1	50.2
variance	645.8	179.1	895.4	1797.1	627.9

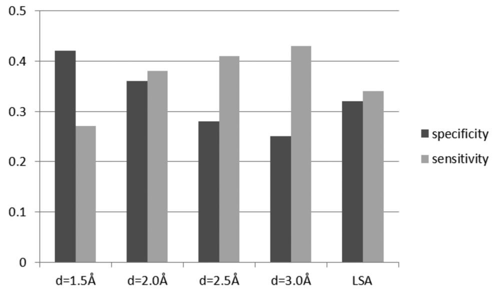


Figure 7. The specificities and sensitivities of 24 complexes for Pro-BS at the four different values of *d*.

Table 3. Comparison of the Results by Pro-BS and LSA for 16 Transient Chains of Complexes and Eight Mixed Types of Complexes

		Pro-BS		LSA	
PDB code	interface[a]	Spec[b]	Sens[c]	Spec	Sens
		Transient			
1apmE	41	13.0	14.6	8.0	9.7
1efuA	83	40.0	33.7	38.0	22.8
1efuB	89	33.7	32.6	36.0	20.2
1g3nA	65	27.8	30.8	52.0	40.0
1g3nB	28	35.7	17.9	12.0	21.4
1g3nC	38	17.3	23.7	26.0	34.2
1gotA	36	21.1	11.1	10.0	13.8
1gotB	123	39.1	27.6	64.0	26.0
1k9oE	38	21.6	28.9	20.0	26.3
1k9oI	21	15.6	23.8	4.0	9.5
1rrpA	72	44.6	40.3	40.0	27.7
1rrpB	68	38.2	30.9	32.0	23.5
1ughE	35	23.3	28.6	12.0	17.1
1ughI	34	47.1	70.6	42.0	61.7
1ytfA	18	56.3	50.0	12.0	33.3
1ytfD	74	47.8	43.2	68.0	45.9
Overall		32.6	31.8	29.8	27.1
		Mixed Types			
1allA	43	46.7	48.8	42.0	48.8
1azeA	27	39.1	66.7	46.0	85.1
1bncA	42	39.1	59.5	26.0	30.9
1daaA	63	33.8	34.9	46.0	36.5
1lucA	65	44.7	52.3	20.0	20.0
1hcgA	33	35.7	45.5	26.0	30.3
1lw6I	18	52.9	50.0	32.0	88.8
1tcoB	66	51.8	43.9	54.0	40.9
Overall		43.0	50.2	36.5	47.7

[a] The number of residues on the actual interface in complex. [b] Spec (%) is the average specificity of the corresponding method on the data set. [c] Sens (%) is the average sensitivity of the corresponding method on the data set.

them contain more than 2/3 of atoms as surface atoms. The percentage of the surface atoms on the actual interfaces of 24 complexes are shown in Figure 5. We can see that 2/3 is a reasonable cutoff.

We have further conducted experiments to validate our observation. Different threshold values, from 0 to 1 with step size 1/12, are tested in Step 2. The final specificity and sensitivity at each threshold value are shown in Figure 6. It is clear that 2/3 is a reasonable choice.

To illustrate the effect of surface detection, we calculate the values of specificity and sensitivity with and without surface detection on 24 complexes. The details are displayed in Table 1. For the 16 transient chains of complexes, the specificity values for Pro-BS without surface detection and the version with surface detection are 26.3% and 32.6%, respectively. The sensitivity values for the two versions are 32.9% and 31.8%, respectively. For the eight mixed types of complexes, the specificity values for Pro-BS without surface detection and the version with surface detection are 37.9% and 43.0%, respectively. The sensitivity values for the two versions are 51.5% and 50.2%, respectively. With surface detection, the specificity values can be improved by at least 5.1%.

Binding Sites Prediction at Different *d* Values. For our method, the balance between specificity and sensitivity can be optimized by carefully choosing the value of *d*. We have tried four different values (1.5 Å, 2.0 Å, 2.5 Å, and 3.0 Å) for *d* to run Pro-BS on 24 complexes.[6] The average predicted sizes for these four *d* values are 33.7 residues, 51.5 residues, 76.2 residues, and 89.1 residues, respectively, while the average actual size is 50.8 residues. The numbers of predicted residues for Pro-BS at these four *d* values are shown in Table 2. Notice that the variances of predicted sizes for Pro-BS at four different *d* are 645.8, 179.1,

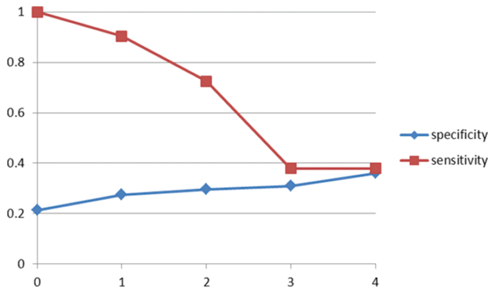


Figure 8. The values of specificity and sensitivity for each step of Pro-BS.

895.4 and 1797.1, while the variance of predicted sizes for LSA is 627.9. It is obvious that when d = 2.0 Å, the variance is the minimum.

We compute specificities and sensitivities for each complex under the four different d values. The specificities for these four values of d are 41.8%, 36.1%, 28.0%, and 25.3%, respectively. The sensitivities for these four values of d are 27.2%, 37.9%, 41.1%, and 43.5%, respectively. The results are shown in Figure 7. It is clear that by setting d = 2.0 Å, both specificity and sensitivity are somewhat better than that of the LSA algorithm.[6]

Comparison to the LSA Algorithm on 24 Complexes. When comparing to the LSA algorithm, Pro-BS predicts the binding sites on 24 complexes by setting d = 2.0 Å. For 16 transient chains of complexes, the values of specificity and sensitivity for Pro-BS are 32.6% and 31.8%, respectively, while the specificity and sensitivity of the LSA algorithm 6 are 29.8% and 27.1%, respectively. For eight mixed types of complexes, the values of specificity and sensitivity for Pro-BS are 43.0% and 50.2%, respectively. The specificity and sensitivity values for the LSA algorithm are 36.5% and 47.7%, respectively. The results are shown in Table 3. Therefore, the overall performance of Pro-BS is better than that of the LSA algorithm. It is interesting to see that Pro-BS does not always outperform the LSA algorithm. In some cases, the LSA algorithm can get better results. In general, the results for the transient chains of complexes are worse than that of the mixed types of complexes.

Running time: We use a Pentium(R) 4 (CPU of 2.40 GHz) to run Pro-BS. For each of the 24 complexes, it takes less than 50 s for Pro-BS.

Analysis of Pro-BS. We notice that Pro-BS performs poorly on instances. To better understand Pro-BS, we investigate the performance of each step by the specificity and sensitivity. The specificities and sensitivities are shown in Figure 8. In Step 1, Pro-BS reports the set of segments from two proteins. Among 24 complexes, 15 have all actual interface residues which are contained in the reported segments.

However, there are many false positives residues in the reported segments, and we need further filtration. The sensitivity and specificity for Step 1 are 90.3% and 27.5%, respectively. In Step 2, Pro-BS filters out the subsegments with no more than 2/3 of its atoms as surface atoms. The specificity (29.6%) is improved by 2.1% and the sensitivity (72.5%) is reduced by 17.8%. The reduction of the sensitivity is huge. However, if we do not perform this step, Step 4 will reduce the sensitivity further. Pro-BS in Step 3 yields the candidate sites of similar 3D structures by rigid transformations. The specificity (31.0%) is improved by 1.4%, yet the sensitivity (37.9%) is reduced by 34.6%. Many residues on the actual interface are not captured by Step 3 of Pro-BS. Step 3 reduces the sensitivity future; however, without Step 3, Step 4 cannot be executed. In Step 4, Pro-BS selects the candidate binding sites under a unified transformation. Among 24 complexes, 14 of them can be improved in terms of the specificity. The specificity and sensitivity for Step 4 are 36.1% and 37.9%, respectively. Some incorrect residues can be pruned by Step 4, while correctly predicted residues being retained.

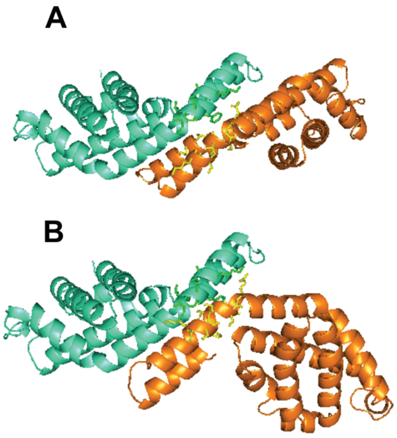


Figure 9. Conformation discovered by Pro-BS for 1all(A:B). (A) is the experimental complex and (B) is the figuration by Pro-BS. The C_α rmsd between two complexes is 5.3 Å.

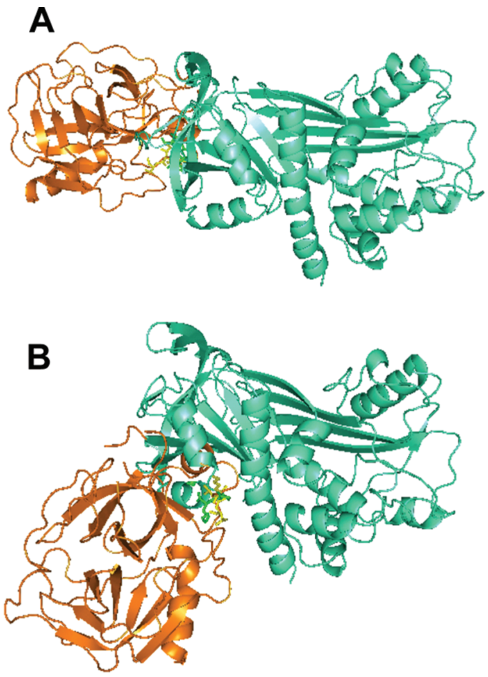


Figure 10. Conformation discovered by Pro-BS for 1k9o(E:I). (A) is the experimental complex and (B) is the figuration by Pro-BS.

Further, we have investigated several instances. One of the good instances is 1all(A:B). Figure 9 displays the orientation discovered by our program. The C_α rmsd (root mean squared deviation) between the experimental complex and the predicted complex is 5.3 Å.

For a few instances, both methods do not perform well. Figure 10 shows such a protein complex, namely 1k9o(E:I), which is not correctly predicted. We can see that the two subunits

Table 4. Values of Specificity and Sensitivity with and without Step 4 on 24 Complexes

	Pro-BS without Step 4		Pro-BS with Step 4	
	Spec[a]	Sens[b]	Spec	Sens
transient complexes	28.3	31.8	32.6	31.8
mixed complexes	36.2	50.2	43.0	50.2

[a] Spec (%) is the average specificity on the data set. [b] Sens (%) is the average sensitivity on the data set.

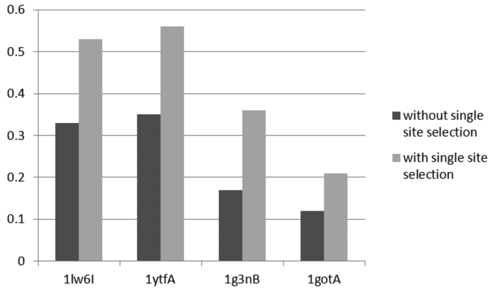


Figure 11. The specificity values with and without *single site selection*.

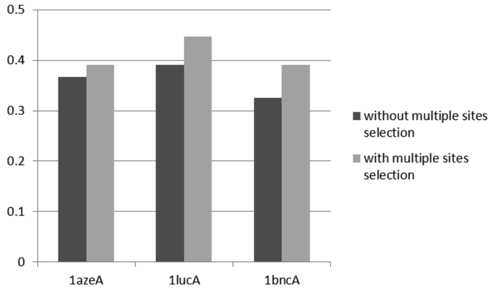


Figure 12. The specificity values with and without *multiple sites selection*.

are 'interacted' at wrong residues. The C_α rmsd between two complexes is 16.7 Å.

Candidate Sites with Optimal Transformation. In Step 4 of our algorithm, we select the candidate binding sites which are matched under a unified transformation. To demonstrate the effect of step 4, we calculate the values of specificity and sensitivity with and without Step 4 on 24 complexes. The details are shown in Table 4. For 16 transient chains of complexes, the specificity value for Pro-BS without Step 4 is 28.3% and that for Pro-BS with Step 4 is 32.6%. The sensitivity values for the two versions are both 31.8%. For eight mixed types of complexes, the specificity value for Pro-BS without Step 4 is 36.2% and that for Pro-BS with Step 4 is 43.0%. The sensitivity values for the two versions are both 50.2%. Therefore, we can see that Step 4 can improve the value of specificity, while the sensitivity value remains essentially the same. We also illustrate the effect of the techniques in Case 1 and Case 2 of Step 4.

Case 1: Given the set of candidate sites, if there are not two or more sites sharing similar transformations, Pro-BS chooses the site which has the minimum number of mismatch atom pairs as the final candidate. This step is referred to as *single site selection*. Among 24 complexes, four of them (1lw6I, 1ytfA, 1g3nB, 1gotA) are of this case. The specificity values with and without *single site selection* are given in Figure 11.

Case 2: When the number of candidate binding sites share a unified transformation is greater than five, we know that some of them may not be actual binding sites. Thus, Pro-BS decreases the value of d from 2.0 Å to 1.5 Å and further selects the candidate binding sites such that at least 1/2 of atom pairs are within distance 1.5 Å. This step is referred to as *multiple sites selection*. Among 24 complexes, three of them (1azeA, 1lucA, 1bncA) are of this case. The specificity values with and without *multiple sites selection* are given in Figure 12.

■ CONCLUSION AND DISCUSSION

We develop a method for predicting protein−protein binding sites in complexes. Our method identifies the binding sites by finding two complementary 3D substructures, where each substructure from one protein. The search space to identify such two substructures is huge. We adopt a four-step strategy to solve this issue. Our experiments show that Pro-BS achieves the overall specificity and sensitivity of 36.1% and 37.9%, respectively.

Our experiments indicate that for all complexes in data set, the number of residues in the two binding sites differ at most by four. This motivates us to design Step 1. Most of the binding sites on the interfaces have more than 2/3 of atoms labeled surface, and followed by which, we design Step 2. In addition, the results show that optimal value of d is 2.0 Å for the 24 complexes. Comparison with the LSA algorithm, Pro-BS can improve the specificity and sensitivity by at least 2.8% and 2.5%, respectively.

Although our method shows better overall performance, there are some protein complexes where the LSA algorithm outperforms Pro-BS. For some specific complexes, Pro-BS and LSA both perform low accuracy. It will be beneficial if we could combine the physicochemical properties with our 3D structural similarity algorithm, to come up with a more reliable method. Step 2 and Step 3 of our approach are not effective. We need better approaches to filter out the false positive results while retaining the true positive results. Classical machine learning methods may be adopted for this purpose. In this paper, we consider that proteins bind to proteins. We should consider the instances where binding partners are ligands that are not proteins.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: cswangl@cityu.edu.hk.

■ ACKNOWLEDGMENT

This work is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 122511].

■ REFERENCES

(1) Bradford, J. R.; Westhead, D. R. Improved Prediction of Protein-Protein Binding Sites Using a Support Vector Machined Approach. *Bioinformatics* **2005**, *21*, 1487–1494.

(2) Chen, Y. C.; Lo, Y. S.; Hsu, W. C.; Yang, J. M. 3D-partner: a Web Server to Infer Interacting Partners and Binding Models. *Nucleic Acids Res.* **2007**, *35*, W561–W567.

(3) Lo, A.; Chiu, Y. Y.; Rodland, E. A.; Lyu, P. C.; Sung, T. Y.; Hsu, W. L. Predicting Helix-Helix Interactions from Residue Contacts in Membrane Proteins. *Bioinformatics* **2008**, *25*, 996–1003.

(4) Li, H.; Li, J.; Wong, L. Discovering Motif Pairs at Interaction Sites From Protein Sequences on a Proteome-Wide Scale. *Bioinformatics* **2006**, *22*, 989–996.

(5) Liu, X.; Li, J.; Wang, L. Modeling Protein Interacting Groups by Quasi-Bicliques: Complexity, Algorithm and Application. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2010**, *7*, 354–364.

(6) Carl, N.; Konc, J.; Vehar, B.; Janežič, D. Protein-Protein Binding Site Prediction by Local Structural Alignment. *J. Chem. Inf. Model.* **2010**, *50*, 1906–1913.

(7) Konc, J.; Janežič, D. An Improved Branch and Bound Algorithm for the Maximum Clique Problem. *MATCH-COMMUN MATH CO* **2007**, *58*, 569–590.

(8) Konc, J.; Janežič, D. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics* **2010**, *26*, 1160–1168.

(9) Konc, J.; Janežič, D. ProBiS: a web server for detection of structurally similar protein binding sites. *Nucleic Acids Res.* **2010**, *38*, W436–W440.

(10) Sussman, J. L.; Lin, D.; Jiang, J.; Manning, N. O.; Prilusky, J.; Ritter, O.; Abola, E. E. Protein Data Bank (PDB): Database of Three-Dimensional Structural Information of Biological Macromolecules. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **1998**, *D54*, 1078–1084.

(11) Smith, T. F.; Waterman, M. S. Identification of Common Molecular Subsequences. *J. Mol. Biol.* **1981**, *147*, 195–197.

(12) Huang, B.; Schröder, M. LIGSITE^{CSC}: Predicting Ligand Binding Sites Using the Connolly Surface and Degree of Conservation. *BMC Struct. Biol.* **2006**, *6*, 19–29.

(13) Guo, F.; Wang, L.; Yang, Y.; Lin, G. *The 4th International Conference on Bioinformatics and Biomedical Engineering*, Chengdu, China, 2010.

(14) Li, S. C.; Bu, D.; Xu, J.; Li, M. In *Combinatorial Pattern Matching*; Ferragina, P., Landau, G., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, 2008; Vol. 5029, pp 44−55.

(15) Higurashi, M.; Ishida, T.; Kinoshita, K. PiSite: a Database of Protein Interaction Sites Using Multiple Binding States in the PDB. *Nucleic Acids Res.* **2009**, *37*, D360–D364.

(16) Carl, N.; Konc, J.; Janežič, D. Protein Surface Conservation in Binding Sites. *J. Chem. Inf. Model.* **2008**, *48*, 1279–1286.