An Efficient Mechanism for Prediction of Protein-Ligand Interactions Based on Analysis of Protein Tertiary Substructures

Darby Tien-Hau Chang, Chien-Yu Chen, Yen-Jen Oyang*
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.

Hsueh-Fen Juan
Institute of Biotechnology
and Department of Chemical Engineering
National Taipei University of Technology,
Taipei, Taiwan, R.O.C.

Hsuan-Cheng Huang
Institute of Biological Chemistry
Academic Sinica, Taipei, Taiwan, R.O.C.

Abstract

*Analysis of protein-ligand interactions is a fundamental issue in drug design. As the detailed and accurate analysis of protein-ligand interactions involves calculation of binding free energy based on thermodynamics and even quantum mechanics, which is highly expensive in terms of computing time, conformational and structural analysis of proteins and ligands has been widely employed as a screening process in computer-aided drug design. In this paper, an efficient mechanism for identifying possible protein-ligand interactions based on analysis of protein tertiary substructures is proposed. In one experiment reported in this papert, the proposed prediction mechanism has been exploited to obtain some clues about a hypothesis that the biochemists have been speculating. The main distinction in the design of the prediction mechanism is the filtering process incorporated to expedite the analysis. The filtering process extracts the residues located in a cave of the protein tertiary structure for analysis and operates with O(n*log*n) time complexity, where* n *is the number of residues in the protein. In comparison, the α–hull algorithm, which is a widely used algorithm in computer graphics for identifying those instances that are on the contour of a 3-dimensional object, features $O(n^2)$ time complexity. Experimental results show that the filtering process presented in this paper is able to speed up the analysis by a factor ranging from 3.11 to 9.79 times.*

Keywords: *protein structural analysis, protein tertiary structure, kernel density estimation.*

1. Introduction

One of the fundamental issues in drug design is analysis of protein-ligand interactions [9, 11]. The detailed and accurate analysis of protein-ligand interactions involves calculation of binding free energy based on thermodynamics and even quantum mechanism [2, 5]. However, this approach is highly expensive in terms of computing time. As a result, conformational and structural analysis of proteins and ligands has been widely employed as a screening process in computer-aided drug design [6, 14, 15, 16].

In this paper, an efficient mechanism for identifying possible protein-ligand interactions based on analysis of protein tertiary substructures is proposed. Fig. 1 illustrates one application that the proposed prediction mechanism addresses. In this application, the biochemist is given the crystal structure of a protein bound with a specific ligand and wants to conduct a search in the PDB database [4] for the other proteins that could interact with the specific ligand. In one experiment presented in this paper, the proposed prediction mechanism has been exploited to investigate whether some proteins in the caspase family contains a similar binding site as the structure of integrin reported in [18]. The experimental results are in conformity with a hypothesis that the biochemists have been speculating.

Concerning the application illustrated in Fig. 1, it is apparent that only the substructures in a cave of the protein tertiary structure are of interest. Therefore, in order to expedite the analysis process, it is desirable to incorporate a mechanism that can effectively extract the residues in a cave of the protein tertiary structure. In this paper, an efficient filtering process with O(*n*log*n*) time complexity is employed, where *n* is the

* Corresponding author. Email: yjoyang@csie.ntu.edu.tw; cychen@mars.csie.ntu.edu.tw Tel: +886-2-23625336 #431 Fax: +886-23688675. This research is sponsored by National Science Council of R.O.C. under contract NSC 92-2323-B-002-013.

number of residues in the protein. In comparison with the α–hull algorithm [7], which is a widely used algorithm in computer graphics for identifying those instances on the contour of a 3-dimensional object, the filtering process employed in this paper features a lower time complexity, O($n$log$n$) versus O(n^2). Experimental results show that the filtering process presented in this paper is able to speed up the analysis by a factor ranging from 3.11 to 9.79 times.

In the following part of this paper, section 2 elaborates the proposed prediction mechanism. Section 3 reports the experiments conducted to evaluate the effects of the proposed prediction mechanism. Finally, concluding remarks are presented in section 4.

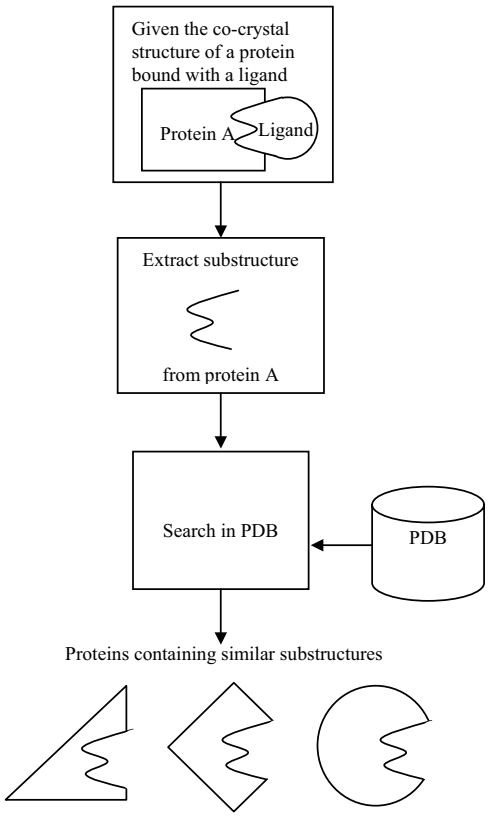


Fig. 1. One application that the proposed prediction mechanism addresses.

2. The prediction mechanism

The prediction mechanism that we have developed for the problem illustrated in Fig. 1 carries out analysis in two steps. In the first step, a filtering process based on an efficient kernel density estimation algorithm is invoked to identify the crucial tertiary substructures on the contour of the protein that the analysis should focus on. In the second step, the geometric hashing algorithm in computer graphics [8, 10] is invoked to compare the crucial substructures of the target protein and the binding/active site of the reference protein. In this paper, we refer to the protein that contains the binding/active site of interest the reference protein and the proteins in PDB with which the alignment is to be conducted the target proteins.

The efficient kernel density estimation algorithm that forms the basis of the filtering process treats a given compact set of instances $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n\}$ in the vector space as n samples randomly taken from a probability distribution with an unknown form and employs the learning algorithm that we have recently proposed [12, 13] to construct an approximate probability density function of the following form:

$$\hat{f}(\boldsymbol{v}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\beta}{\lambda\cdot\sigma_i}\right)^m \exp\left(-\frac{\| \boldsymbol{v}-\boldsymbol{s_i} \|^2}{2\sigma_i^2}\right), \qquad (1)$$

where

(i) $\boldsymbol{v}$ is a vector in an m-dimensional vector space,

(ii) β is the parameter that controls the smoothness of the approximation function,

(iii) $\sigma_i = \beta\delta_i = \beta\dfrac{\overline{R}(\boldsymbol{s_i})\sqrt{\pi}}{\sqrt[m]{(k+1)\Gamma(\frac{m}{2}+1)}}$ and

$$\overline{R}(\boldsymbol{s_i}) = \frac{m+1}{m}\left(\frac{1}{k}\sum_{h=1}^{k}\| \hat{\boldsymbol{s}}_{\boldsymbol{h}} - \boldsymbol{s_i} \|\right),$$

where $\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \ldots, \hat{\boldsymbol{s}}_k$ are the k nearest neighbors of $\boldsymbol{s_i}$ and k is a parameter to be set by the user.

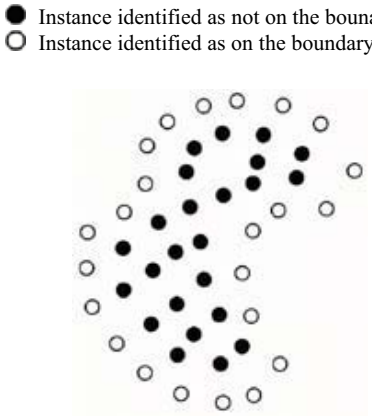
(iii) $\lambda = \sum_{h=-\infty}^{\infty}\exp\left(-\dfrac{h^2}{2\beta^2}\right)$.

One interesting observation is that, regardless of which $\beta = \dfrac{\sigma_i}{\delta_i}$ ratio is employed, we have $\dfrac{\lambda}{\beta} \cong \sqrt{2\pi}$. If this observation can be proved to be generally correct, then we can further simplify equation (1) and obtain

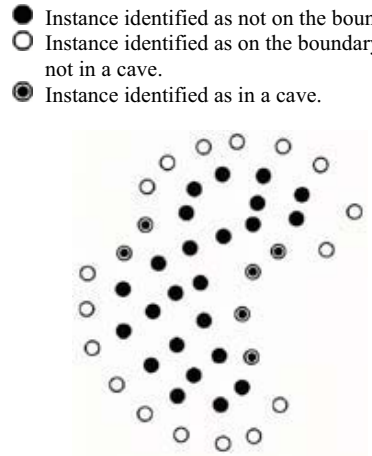
$$\hat{f}(\boldsymbol{v}) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{\sqrt{2\pi}\cdot\sigma_i}\right)^m \exp\left(-\frac{\| \boldsymbol{v}-\boldsymbol{s_i} \|^2}{2\sigma_i^2}\right). \qquad (2)$$

As the approximate probability density function presented in equations (1) and (2) is a continuous and smooth function in the vector space, we can expect that the function values at the instances located on the boundary of the set of the instances are generally smaller than the function values at the rest of instances. Accordingly, we can set a threshold of the function values to distinguish those instances that are located on the boundary from those that are not. Fig. 2 depicts a

2-dimensional example to illustrate the effect of the filtering process. In this example, a 2-dimensional object is composed of a number of primitive instances represented by dots in the figure. Fig. 2(a) shows the instances that are identified as on the boundary of the object by the filtering process.



(a) The effect after the instances on the boundary of the object have been identified.



(b) The effect after the instances in the caves of the object have been identified.

Fig. 2. An example that illustrates the effects of the filtering process.

With the instances on the boundary of the object been successfully identified, the next task of the filtering process is to further classify each of these instances depending on whether it is located in a cave of the object or not. This task can be carried out by applying equation (1) or (2) again but with a larger β value. Applying equation (1) or (2) with a larger β value implies that the approximate probability density function obtained is smoother. As a result, the function values at those instances that are located in a cave will be generally larger than the function values at those instances that are on the boundary of the object but not in a cave. Accordingly, a threshold can be set to classify these instances. Fig. 2(b) shows the final result obtained in this example and Fig. 3 shows the pseudo-code of the filtering process.

With the filtering process carried out for both the reference protein and the target protein, the next task that the proposed prediction mechanism carries out is conducting structural alignment on the crucial substructures identified. In the proposed prediction mechanism, analysis is carried out at the residue level with each residue represented by its alpha carbon in the vector space. In other words, a protein substructure is defined by the coordinates of the alpha carbons included in the substructure. In our implementation, we have adopted the common practice for carrying out protein structural alignment with the geometric hashing algorithm [6, 14, 15, 16, 17]. With this practice, the coordinate systems examined by the geometric hashing algorithm are limited to those defined by the two backbone bonds connected to the alpha carbon of each residue. With the filtering process elaborated above, in our implementation, the geometric hashing algorithm further narrows down its search space to only the coordinate systems defined by the residues located in a cave.

Algorithm kernel density estimation based filtering
Input: A set $S = \{s_1, s_2, \ldots, s_n\}$ of instances in the vector space and parameters β_1, β_2, k, r_1, and r_2.
Output: $\hat{S}$, a subset of S.

Set $\hat{S} \leftarrow S$.
For each $s_i \in S$ do the following:
  Compute $\hat{f}(s_i)$ according to equation (1) with $\beta = \beta_1$.
Set $w \leftarrow \max_{1 \le i \le n} \{\hat{f}(s_i)\}$.
For each $s_i \in S$ do the following:
  If $\hat{f}(s_i) \ge r_1 \cdot w$, then $\hat{S} \leftarrow \hat{S} - \{s_i\}$.
For each $s_i \in S$ do the following:
  Compute $\hat{f}(s_i)$ according to equation (1) with $\beta = \beta_2$.
Set $w \leftarrow \max_{1 \le i \le n} \{\hat{f}(s_i)\}$.
For each $s_i \in \hat{S}$ do the following:
  If $\hat{f}(s_i) \le r_2 \cdot w$, then $\hat{S} \leftarrow \hat{S} - \{s_i\}$.
Return ($\hat{S}$).

Fig. 3. The pseudo-code of the kernel density estimation based filtering process.

As far as the time complexity of the prediction mechanism is concerned, in equations (1) and (2) we need to identify the k nearest neighbors for each of the n instances. If the kd-tree structure [3] is incorporated, then the average time complexity for constructing a kd-tree with n instances is $O(n \log n)$, if k is considered as a constant. One practical implementation employed in this paper for evaluating equation (1) is to include only the nearest k' instances of vector $\boldsymbol{v}$, since the influence of the Gaussian function decreases exponentially. With this practice, the time complexity for evaluating the approximate function value at one instance is therefore $O(k' \log n)$ and the overall average time complexity of the filtering process is O($n$log$n$), if both k and k' are considered as constants. Concerning the structural alignment process, as the geometric hashing algorithm narrows down its search space to only the coordinate systems defined by the residues located in a caves of the reference protein and the target protein, the time complexity for comparing the crucial substructures of these two proteins is $O(n_1' n_2' (n_1' + q))$, where n_1' and n_2' are the numbers of residues identified as in the caves of the two proteins, respectively, and q is the number of residues in the binding sites of the reference protein.

3. Experimental results

This section reports two experiments conducted to evaluate the effects of the proposed prediction mechanism. The main objective of the first experiment is to test the accuracy of the proposed prediction mechanism. The second experiment demonstrates how biochemists can exploit the proposed prediction mechanism to facilitate their research works. Table 1 shows how the parameters are set for the filtering process in the experiments and the criteria for successful alignment of two alpha carbons in the geometric hashing algorithm. In the experiments, the likelihood of residue substitution is also taken into account. If the entry in the PAM 250 matrix [1, 11] corresponding to a pair of residues aligned by the geometric hashing algorithm is smaller than 2, then this pair of residues is excluded from the list of successfully aligned.

In the first experiment, three datasets, each of which contains a reference structure and a number of target proteins, are used to test whether the proposed prediction mechanism is able to identify the region on the contour of the target protein that contains a similar substructure as the reference protein. Table 2(a) shows the characteristics of these three reference protein structures. The first two reference structures are two alcohol dehydrogenase enzymes in PDB, PDB ID = 1hdz and 1b15, and the third reference structure, PDB ID = 115g, contains an integrin αVβ3 bound with a peptide ligand as reported in [18]. For each of the two enzyme proteins, 5 proteins from the same family in PDB are employed as the target proteins. For integrin, the alternative structures of integrin with different bindings, PDB ID = 1jv2 and 1m1x, are employed as the target proteins. Table 2(b) reports the results of the first experiment. The experimental results show that, with a high degree of accuracy, the proposed prediction mechanism is able to identify the residues in the binding/active sites of the target protein. The only miss occurs when protein *1hj6* is aligned with reference protein *1b15*. However, as Table 2(b) shows, the miss is not due to the filtering process invoked to expedite the analysis. Without the filtering process, the geometric hashing algorithm still can only successfully align 6 out of the 8 residues in the active site of protein *1hj6* with the residues in the active site of the reference protein.

In the second experiment, the proposed prediction mechanism is invoked to figure out whether some proteins in the Caspase family may contain a similar binding site as integrin. Table 3 shows the results outputted by the proposed prediction mechanism. It is observed that caspase-7, PDB ID = 1f1j, caspase-8, PDB ID = 1f9e, and caspase-9, PDB ID = 1jxq, have the largest numbers of residues successfully aligned with the residues in the binding site of integrin. This result is in conformity with a hypothesis that the biochemists have been speculating. However, the outputs of the proposed prediction mechanism can only be regarded as interesting clues and, as shown in Table 3(b), it is typical that multiple possible alignments are found. Therefore, the biochemists must conduct more in depth analyses, such as protein docking or protein affinity analysis, to further verify the hypothesis.

Table 1. Parameter settings in the experiments.

Parameter	Value
β_1 in pseudo code	0.45
β_2 in pseudo code	0.9
k in equations (1) and (2)	30
k': the number of nearest Gaussian functions involved in evaluating equation (1) or (2)	30
r_1 in pseudo code	0.45
r_2 in pseudo code	0.63

(a) Parameter settings for the filtering process.

$\lvert v' \rvert - \lvert v'' \rvert \le 7$Å
$\lvert \theta_x' - \theta_x'' \rvert \le 0.2$ radian
$\lvert \theta_y' - \theta_y'' \rvert \le 0.2$ radian
$\lvert v' - v'' \rvert \le 6$Å

v' and v'' are the vectors from the origin to the two alpha carbons, respectively. θ_x' and θ_x'' are the angles between axis x and the two vectors v' and v'', respectively. θ_y' and θ_y'' are the angles between axis y and the two vectors v' and v'', respectively.

(b) Criteria for successful alignment of two alpha carbons in the geometric hashing algorithm.

Table 2. Data of the first experiment.

PDB ID	# of residues	# of residues in the binding/active site	# of residues remaining with the filtering process applied
1hdz	748	14	307
1b15	508	8	203
1l5g	1470	18	833

(a) Characteristics of the reference proteins.

Reference protein		1hdz				
Target protein		3hud	1htb	1hdy	1deh	1hdx
# of residues in the active site		14	14	14	14	14
Geometric hashing without filtering	Execution time of geometric hashing in seconds	130.24	114.68	129.57	117.13	137.43
	# of residues in the active site that are successfully aligned	14	14	14	14	14
	RMSD of aligned pairs	0.79	0.37	0.47	0.36	0.49
Geometric hashing with filtering applied	Execution time of filtering in seconds	0.18	0.2	0.18	0.19	0.18
	Execution time of geometric hashing in seconds	27.78	25.38	25.45	24.76	26.2
	# of residues in the active site that are successfully aligned	14	14	14	14	14
	RMSD of aligned pairs	0.96	0.37	0.54	0.36	0.66
Speedup due to the filtering process		4.66	4.48	5.06	4.69	5.21

Reference protein		1b15				
Target protein		1ide	1hj6	1idc	1idd	1idf
# of residues in the active site		8	8	8	8	8
Geometric hashing without filtering	Execution time of geometric hashing in seconds	16.75	15.89	16.7	15.7	16.7
	# of residues in the active site that are successfully aligned	8	6	8	8	8
	RMSD of aligned pairs	0.52	0.42	0.55	0.49	0.43
Geometric hashing with filtering applied	Execution time of filtering in seconds	0.08	0.07	0.08	0.08	0.08
	Execution time of geometric hashing in seconds	2.55	2.13	2.05	2.18	2.13
	# of residues in the active site that are successfully aligned	8	6	8	8	8
	RMSD of aligned pairs	0.56	0.54	0.62	0.5	0.57
Speedup due to the filtering process		6.37	7.22	7.84	6.95	7.56

Reference protein		1l5g	
Target protein		1jv2	1m1x
# of residues in the binding site		18	18
Geometric hashing without filtering	Execution time of geometric hashing in seconds	1051.5	1036.22
	# of residues in the binding site that are successfully aligned	18	18
	RMSD of aligned pairs	1.21	1.22
Geometric hashing with filtering applied	Execution time of filtering in seconds	0.44	0.45
	Execution time of geometric hashing in seconds	338.12	318.66
	# of residues in the binding site that are successfully aligned	18	18
	RMSD of aligned pairs	1.23	1.22
Speedup due to the filtering process		3.11	3.25

(b) Experimental results, where RMSD stands for the root-mean-square difference.

Concerning the experimental results shown in Table 2 and Table 3, there are a few issues that deserve further discussion. The first issue concerns the speedups reported in these two tables, which range from 3.11 to 9.79 times. In fact, these numbers just represent a subjective tradeoff between the accuracy of the analysis results and the magnitude of speedup obtained. As the expediting mechanism works by reducing the number of residues to be included in the analysis process, in principle, we could set the parameters listed in Table 1 with a conservative view and an insignificant magnitude of speedup would result. On the other hand, if we set the parameters listed in Table 1 with a more aggressive view, then we could obtain a higher level of speedup but might lose some degree of analysis accuracy as a consequence.

Table 3. Results of the second experiment.

PDB ID of the Target Protein		1cl5	1cww	1cy5	1f1j	1f9e	1gqf
# of residues		97	102	93	469	1476	530
# of residues remaining with filtering applied		26	31	38	184	542	103
Geometric hashing without filtering	Execution time of geometric hashing	7.83	8.23	7.52	103.32	931.17	149.93
	# of residues in a cave that are successfully aligned	9	10	9	16	15	14
	RMSD of aligned pairs	3.91	3.46	3.37	4.24	3.99	4.75
Geometric hashing with filtering applied	Execution time of filtering	0.01	0.01	0.01	0.09	0.43	0.1
	Execution time of geometric hashing	1.3	1.52	1.76	24	217.1	15.22
	# of residues in a cave that are successfully aligned	9	8	9	13	15	13
	RMSD of aligned pairs	3.91	4.08	3.37	4.01	4.25	5.06
Speedup due to the filtering process		5.98	5.38	4.25	4.29	4.28	9.79

PDB ID of the Target Protein		1jxq	1k86	1k88	1nme	2ygs
# of residues		940	464	461	238	92
# of residues remaining with filtering applied		329	110	112	66	33
Geometric hashing without filtering	Execution time of geometric hashing	452.89	112.72	109.42	33.73	7.83
	# of residues in a cave that are successfully aligned	14	14	13	12	9
	RMSD of aligned pairs	4.24	4.06	4.04	4.14	3.51
Geometric hashing with filtering applied	Execution time of filtering	0.22	0.09	0.09	0.03	0.02
	Execution time of geometric hashing	83.45	18.45	15.43	4.75	1.54
	# of residues in a cave that are successfully aligned	12	12	13	11	9
	RMSD of aligned pairs	4.16	4.15	4.04	3.87	3.51
Speedup due to the filtering process		5.41	6.08	7.05	7.06	5.02

(a) Output of the proposed prediction mechanism.

Protein integrin αVβ3 (reference protein)			Protein caspase-8 (PDB ID = 1f9e)			PAM250 Score
Chain	Residue Index	Residue Type	Chain	Residue Index	Residue Type	
A	178	TYR	D	320	TYR	10
A	218	ASP	A	297	GLU	3
B	119	ASP	B	388	GLN	2
B	121	SER	B	339	SER	2
B	122	TYR	B	340	TYR	10
B	123	SER	B	378	SER	2
B	126	ASP	B	351	GLN	2
B	158	ASP	A	291	GLN	2
B	215	ASN	B	381	ASP	2
B	216	ARG	B	384	LYS	3
B	217	ASP	D	323	ASP	4
B	219	PRO	D	322	PRO	6
B	220	GLU	D	324	GLU	4
B	251	ASP	B	374	ASN	2

Protein integrin αVβ3 (reference protein)			Protein caspase-8 (PDB ID = 1f9e)			PAM250 Score
Chain	Residue Index	Residue Type	Chain	Residue Index	Residue Type	
A	150	ASP	K	289	ASN	2
A	178	TYR	K	290	TYR	10
A	218	ASP	L	385	GLN	2
B	119	ASP	K	170	ASN	2
B	121	SER	K	236	SER	2
B	122	TYR	K	244	TYR	10
B	126	ASP	K	178	ASP	4
B	127	ASP	K	180	ASN	2
B	215	ASN	K	239	ASP	2
B	216	ARG	K	240	LYS	3
B	217	ASP	K	286	GLN	2
B	218	ALA	K	284	ALA	2
B	219	PRO	L	387	PRO	6
B	220	GLU	K	283	GLN	2
B	251	ASP	V	4604	ASP	4

(b) Two possible mappings of the residues in the crucial substructures of caspase-8 to the residues in the binding site of integrin $\alpha \mathrm{V} \beta 3$.

Another issue that deserves further discussion is whether these parameters should be set differently, if different sets of proteins are to be analyzed. According to our experiences, when another reference protein is given, the only parameter values in Table 1(a) that need to be adjusted are r_1 and r_2. That is, except the values of r_1 and r_2, the user basically can adopt the parameter values listed in Table 1 (a), when a different reference protein is given. The main reason why only r_1 and r_2 need to be taken into account is that the parameter setting just represents a subjective tradeoff between the accuracy of the analysis and the magnitude of speedup obtained. By setting r_1 to 1 and r_2 to 0, we basically keep all the residues. On the other hand, by setting either r_1 to 0 or r_2 to 1, we can eliminate all the residues. Since we can realize our view of tradeoff without any limitation through adjusting the values of r_1 and r_2, there is no need to manipulate other parameters. Basically, when given a reference protein, the user needs to try a number of possible combinations of r_1 and r_2, and select one that can accurately extract the residues in the binding site of the reference protein.

4. Conclusion and future work

In this paper, an efficient mechanism for prediction of possible protein-ligand interaction based on analysis of protein tertiary substructures is proposed. In one experiment presented in this paper, the proposed prediction mechanism has been exploited to investigate whether some proteins in the caspase family contains a similar binding site as the structure of integrin reported in [18] and the experimental results are in conformity with a hypothesis that the biochemists have been speculating. However, the predictions made by the proposed mechanism can only be regarded as interesting clues for more in depth investigations. The experimental results also show that the filtering process presented in this paper is able to speed up the analysis process by a factor ranging from 3.11 to 9.79 times.

As the experiences learned from this research work have been encouraging, continuous investigation of the following issues is of interest. The first issue concerns whether we can identify significant structure activity relationships (SAR) among the binding sites of proteins. With the SAR, the drug design process will be greatly facilitated. The second issue that deserves further investigation concerns prediction of protein functions based on crucial tertiary substructures.

References:

1. Altschul S.F. (1991) Amino acid substitution matrices from an information heoretic perspective. *J. Mol. Biol.* **219**, 555-565.
2. Atkins, P.W. and Depaula, J. (2001), Physical Chemistry, 7th ed., W H Freeman & Co.
3. Bentley, J. L. (1975) Multidimensional binary search trees used for associative searching," Communication of the ACM, **18**, 509-517.
4. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000) The Protein Data Bank. *Nucleic Acids Research*, **28**, 235-242.
5. Bourne, P.E. and Weissig H. ed. (2003) Structural Bioinformatics, New Jersy: Wiley-Liss, Inc.
6. Boutonnet, NS, Rooman, MJ, Ochagavia, ME, Richelle, J, Wodak, SJ. (1995) Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. Protein Eng. **8**, 647-62.
7. Edelsbrunner, H., and Mucke, E.P. (1994) Three-dimensional alpha shapes. *ACM Trans. Graphics*, **13**, 43-72.
8. Haim, J. W. (1997) Geometric hashing: an overview. *IEEE Comput. Sc. and Eng.* **4**, 10-21.
9. Krane, D. E.and Raymer, M. L. (2002) *Fundamental Concepts of Bioinformatics*, Benjamin Cummings.
10. Lamdan, Y. and Wolfson, H. (1988) Geometric Hashing: A General and Efficient Model-Based Recognition Scheme, *Proc. Int'l Conf. Computer Vision*, 238-249.
11. Lesk, A.M. (2002) *Introduction to bioinformatics*, New York : Oxford University Press.
12. Oyang, Y.-J., Chang, D. T.-H., Chen, C.-Y., and Hwang, S.-C., Expediting Protein Structural Analysis with an Efficient Kernel Density Estimation Algorithm , In Proceedings of IEEE 5th International Symposium on Multimedia Software Engineering, Taichung, Taiwan, 2003.
13. Oyang, Y.-J., Hwang, S-C, Ou, Y.-Y., Chen, C.-Y., and Chen, Z.-W. (2002) A Novel Learning Algorithm for Data Classification with Radial Basis Function Networks, In *Proceedings of 9th International Conference on Neural Information Processing* (ICONIP-2002), Singapore, 2002.
14. Orengo, C. and Taylor, W. (1996) SSAP: Sequential Structure Alignment Program for Protein Structure Comparison. Methods in Enzymology, **266**, 617-635.
15. Pennec, X. and Ayache, N. (1994) An O(n2) Algorithm for 3D Substructure Matching of Proteins, In A. Califano, I. Rigoutsos, and H.J. Wolson, editors, Shape and Pattern Matching in Computational Biology - Proc. First Int. Workshop, Seattle, pp. 25-40, Plenum Publishing.
16. Pennec, X. and Ayache, N. (1998) A geometric algorithm to find small but highly similar 3D substructures in proteins, Bioinformatics, **14**, 516-522.
17. Tu, J.-T. (2003) Protein Active Site Prediction By Matching 3D Structural Data, Master thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2003.
18. Xiong, J.P., Stehle, T., Zhang, R., Joachimiak, A., Frech, M., Goodman, S.L., and Arnaout, M.A. (2002) Crystal structure of the extracellular segment of integrin alpha Vbeta3 in complex with an Arg-Gly-Asp ligand. Science. **296**, 151-5