Nucleic Acids Research, 2006, Vol. 34, Web Server issue **W303–W309**
doi:10.1093/nar/gkl344

Protemot: prediction of protein binding sites with automatically extracted geometrical templates

Darby Tien-Hao Chang[1,*], Yi-Zhong Weng[1], Jung-Hsin Lin[3], Ming-Jing Hwang[4] and Yen-Jen Oyang[1,2,*]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 106, Taiwan, Republic of China, [2]Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan, Republic of China, [3]School of Pharmacy, National Taiwan University, Taipei 106, Taiwan, Republic of China and [4]Institute of Biomedical Sciences, Academia Sinica, Taipei 115, Taiwan, Republic of China

Received February 14, 2006; Revised March 7, 2006; Accepted April 18, 2006

ABSTRACT

Geometrical analysis of protein tertiary substructures has been an effective approach employed to predict protein binding sites. This article presents the Protemot web server that carries out prediction of protein binding sites based on the structural templates automatically extracted from the crystal structures of protein–ligand complexes in the PDB (Protein Data Bank). The automatic extraction mechanism is essential for creating and maintaining a comprehensive template library that timely accommodates to the new release of PDB as the number of entries continues to grow rapidly. The design of Protemot is also distinctive by the mechanism employed to expedite the analysis process that matches the tertiary substructures on the contour of the query protein with the templates in the library. This expediting mechanism is essential for providing reasonable response time to the user as the number of entries in the template library continues to grow rapidly due to rapid growth of the number of entries in PDB. This article also reports the experiments conducted to evaluate the prediction power delivered by the Protemot web server. Experimental results show that Protemot can deliver a superior prediction power than a web server based on a manually curated template library with insufficient quantity of entries. Availability: http://protemot.csie.ntu.edu.tw/step1.cgi http://bioinfo.mc.ntu.edu.tw/protemot/step1.cgi.

INTRODUCTION

Function prediction of new proteins is a critical issue in systems biology research (1). One of the most widely adopted approaches is based on sequence similarity of homologous proteins (2). However, in many cases, proteins with similar functions are not homologues. Therefore, search for a tertiary substructure that geometrically matches the 3D pattern of the binding site of a well-studied protein provides a complementary approach for prediction of protein functions (3–6). In this regard, as the structural genomics project worldwide continues to work hard to determine the tertiary structures of many new proteins, automatic extraction of structural templates becomes an essential mechanism for effectively accommodating to the new release of protein structure databases such as PDB (Protein Data Bank) (7) that contains a large number of crystal structures of protein–ligand complexes.

This article describes the design of the Protemot web server, which is equipped with an automatic mechanism to extract structural templates of protein binding sites from the latest release of PDB. With the version of PDB released on November 14, 2005, the automatic extraction mechanism identified a total of 2362 distinctive templates of protein binding sites. The design of Protemot is also distinctive by the mechanism employed to expedite the analysis process that matches the tertiary substructures on the contour of the query protein with the templates in the library. This expediting mechanism is essential for providing reasonable response time to the user as the number of entries in the template library continues to grow rapidly due to rapid growth of the number of entries in the PDB.

This article also reports the experiment conducted to evaluate the prediction power delivered by the Protemot web server. The evaluation has been conducted with a

*To whom correspondence should be addressed. Tel: +886-2-3366-4888 (ext. 431); Fax: +886-2-2368-8675; Email: darby@csie.ntu.edu.tw
*Correspondence may also be addressed to Yen-Jen Oyang. Email: yjoyang@csie.ntu.edu.tw

comparison against the prediction power delivered by the web server based on the well-known Catalytic Site Atlas (CSA) (3), whose maintenance requires manual intervention. Though manually curated template libraries are generally well annotated, the application of such libraries may cause some unexpected problems. For example, as demonstrated by the experimental results reported in this article, a manually curated template library may contain insufficient quantity of entries for carrying out proper comparison and making accurate prediction. Therefore, development of automatic and accurate mechanisms for maintaining the template libraries can greatly facilitate binding sites annotation of proteins with unknown functions.

METHODS

In this section, we will first describe how to automatically extract structural templates of protein binding sites from the PDB. Second, we will elaborate the structural alignment procedure incorporated in Protemot for predicting the binding sites of the query protein.

Each of those templates automatically extracted from the PDB consists of a number of contact residues. A residue in the crystal structure of a protein–ligand complex of PDB is said to be one of the contact residues, if it contains one or more heavy atoms that are <4.5 Å away from the heavy atoms of the ligand. In the template extraction process, three filters were employed to guarantee quality of the templates extracted. First, PDBsum (8) is queried to identify all ligand names within each PDB file. Subsequently, 'pseudo-ligands', e.g. counter ions, metal ions or molecules used for setting up proper crystallization conditions, will be filtered out according to the list detailed in Supplementary Table S.3. The second filter will recognize COMPND, SEQRES, MODEL, ATOM, TER, HETATM, ENDMDL and END cards to extract ligands in PDB files. This filter will discard those ligands that appear in a PDB file many times with identical name but without a chain ID. It is anticipated that these parsing limitations will exclude some real ligands at this stage. Finally, templates with less than three amino acids are filtered out by the third filter. It is noted that in our templates the metal ions are reserved. Most of the metal ions have significant biological functions and therefore could be used to extend the mechanism of template matching.

In the final stage of the template extraction process, the CD-HIT clustering algorithm (9) is invoked to remove redundant templates. A template is said to be redundant if there exists another protein structure in the PDB that meets the following criteria:

(i) the sequence alignment (10) of the protein from which the template is extracted and the second protein has >60% of identity;
(ii) with the sequence alignment, the second protein contains exactly the same template at the same locations as the protein from which the template is extracted;
(iii) the sequence of the second protein is longer than the protein from which the template is extracted.

With the automatic extraction mechanism described above, 2362 distinctive templates have been extracted from the version of PDB released on November 14, 2005.

Figure 1 depicts the workflow of the analysis procedure incorporated in Protemot to match the tertiary substructures on the contour of the query protein with the templates in the library. The design of the analysis procedure in Protemot has been derived from the design that we employed in the Proteminer web server (6,11). The major difference is the incorporation of the refinement process and the screening process marked with an asterisk in Figure 1. The incorporation of the refinement process has been aimed at expediting the entire analysis procedure, which is essential for providing reasonable response time to the user as the number of entries in the template library continues to grow rapidly due to the rapid growth of the number of entries in the PDB. As will be elaborated in the later part of this article, with the refinement process incorporated, we can adopt a strict criterion in the filtering process depicted in Figure 1 and therefore speed up the entire analysis procedure by a factor >10 without trading off accuracy.

The structural alignment procedure shown in Figure 1 works by first invoking a novel filtering process to extract those residues in the proximity of a cavity of the query protein. The filtering process is based on the kernel density estimation algorithm that we have proposed recently (6,11) and its detailed implementation can be found in the Supplementary Data. The filtering process effectively reduces the number of coordinate systems that the geometric hashing algorithm (12), which is the core of the entire analysis procedure, needs to examine; i.e. with the filtering process, the geometric hashing algorithm narrows down its search domain

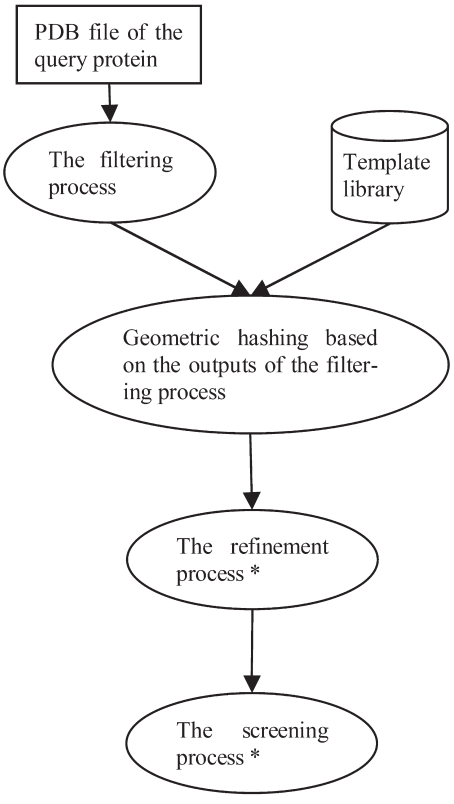

Figure 1. Workflow of the analysis procedure incorporated in Protemot.

of the coordinate systems associated with the query protein to those defined by the residues that pass the filtering process.

With the output of the filtering process, the geometric hashing algorithm (12) is then invoked to match the tertiary substructures on the contour of the query protein with the templates in the library. In our implementation, the structural alignment is conducted at the residue level with each residue represented by its alpha carbon in the vector space. In addition, the common practice for carrying out protein structural alignment with the geometric hashing algorithm is employed (11,13–15). 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. Accordingly, the time complexity of the geometric hashing algorithm for aligning the query protein with a template is $O(n_1 n_2(n_1+n_2))$, where n_1 is the number of residues in the template and n_2 is the number of the residues in the query protein that pass the filtering process.

As mentioned earlier, one of the major improvement in the design of Protemot over the design of Proteminer (6) is the inclusion of the refinement process marked with an asterisk in Figure 1. The refinement process, which is based on the algorithm proposed in Ref. (16), carries out an optimization operation to fine-tune the alignment frames output by the geometric hashing algorithm. Figure 2 shows the pseudo-code of the refinement process. In the loop of the refinement process, the optimization algorithm presented in Ref. (16) is invoked to efficiently solve the general problem defined in the following.

Given a number of paired vectors in the 3D vector space $(\boldsymbol{v_1}, \boldsymbol{v_1}')$, $(\boldsymbol{v_2}, \boldsymbol{v_2}'), \ldots, (\boldsymbol{v_h}, \boldsymbol{v_h}')$, find an optimal translation matrix M_t and an optimal rotation matrix M_r, so that $\frac{1}{h}\sum_{i=1}^{h} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\|^2$ is minimized, where $\hat{\boldsymbol{v}}_i$ is the vector obtained by applying M_t and M_r to vector $\boldsymbol{v_i}'$.

In our implementation, the refinement process is applied to only the alternative coarse-grain alignment frames that are most highly ranked in the output of the geometric hashing algorithm. Three measures are employed to rank the alternative alignment frame. The first measure is the number of residues in the template that are successfully aligned with the residues in the query protein. In this paper, one residue in the query protein is said to be successfully aligned with one residue in the template, if the distance between this pair residues in the alignment frame of the coordinate system is <3 Å. Furthermore, this pair of residues must have similar physicochemical properties. The criterion employed to check the similarity of physicochemical properties is that this pair of residues must correspond to an entry in the PAM 250 matrix (17) that is ≥2. If two possible alignment frames yield the same score with the first measure, then the second measure is employed to rank these two alignment frames. The second measure computes the lumped sum of the PAM 250 scores corresponding to the pairs of aligned residues. If two possible alignment frames yield the same score with both the first and the second measures, then the third measure is employed. The third measure computes the root mean square deviation (r.m.s.d.) of the pairs of aligned residues. In our implementation, the refinement process is applied to only the 100 highest ranked possible alignment frames output by the geometric hashing algorithm.

At the output of the refinement process, three criteria are further imposed to screen the alignment frames. Those alignment frames that cannot pass all three criteria are deleted. The first criterion is that >50% of the residues in the query protein are successfully aligned with the residues in the template. The second criterion is that the r.m.s.d. of the pairs of aligned residues must not exceed 1.5 Å. The third criterion concerns which direction the opening of the binding site points to. In our implementation, we first define g_0 as follows:

$$g_0 = \left\langle \frac{x_{\max} - x_{\min}}{2}, \frac{y_{\max} - y_{\min}}{2}, \frac{z_{\max} - z_{\min}}{2} \right\rangle,$$

where $x_{\max}$ and $x_{\min}$, respectively, denote the maximum and minimum values of the x-coordinates of the residues in the query protein that are successfully aligned with the residues in the template. Similar definition applies to $y_{\max}$, $y_{\min}$, $z_{\max}$ and $z_{\min}$. Then, let g_1 denote the geometric center of all the alpha carbons that are within 10 Å from g_0. In our implementation, we defined vector $g_1 - g_0$ as the direction which the opening of the binding site points to. According to this definition, both the template and the substructure of the query protein that is aligned with the template are associated with a vector. The third criterion requires that the cosine value of the angle between these two vectors must be >0.87.

1. Given an initial alignment A which contains h aligned alpha carbon pairs $(\boldsymbol{v_1}, \boldsymbol{v_1}')$, $(\boldsymbol{v_2}, \boldsymbol{v_2}')$, …, $(\boldsymbol{v_h}, \boldsymbol{v_h}')$.
2. Compute the motion matrices M_t and M_r according to A.
3. Apply M_t and M_r on all alpha carbons of the query protein.
4. Re-align the template and the query protein under the new coordinate system and obtain a new alignment.
5. Go back to step 2 until the alignment converges.

Figure 2. Pseudo-code of the refinement process.

PRACTICAL ISSUES

This section addresses a few issues concerning the practical use of Protemot. The first issue concerns how fast the template library can be updated upon the new release of PDB. In the first phase of the automatic extraction mechanism, the entire PDB is scanned once to identify all the templates, including non-redundant as well as redundant templates, from the crystal structures of protein–ligand complexes in the PDB. Accordingly, the time complexity of this phase of operation in terms of the number of entries in the PDB is $O(n)$. Then, the CD-HIT clustering algorithm is invoked to remove redundant templates and the time complexity of the clustering algorithm is $O(qn)$ (9), where q is the number of non-redundant templates identified by the clustering algorithm. Therefore, the overall time complexity of the automatic extraction mechanism is $O(qn)$. In practice, upon the release of a new version of PDB, we can re-generate a new template library in 15 min on a Pentium-4 based personal computer with a 2.6 GHz processor and 1 GB main memory.

Since generation of the template library from scratch does not take long, in our current implementation, we simply re-generate the entire template library upon the new release of PDB. However, in case it is desirable that the template library is incrementally updated, one can easily implement this feature by modifying the criteria employed to detect redundant templates. This incremental approach is of interest to the users who want to exploit the automatic extraction mechanism to enrich a template library that has been built based on a manually crafted approach such as CSA. In this regard, it is of interest to learn whether the automatic extraction mechanism can generate templates that are highly similar to the templates in a manually curated template library. As shown in Table 1, for the majority of the templates in the CSA library, we can find templates in the Protemot library that have a high degree of structural similarity.

The next issue addressed in this section is the speedup achieved with the refinement process marked by an asterisk in Figure 1. The refinement process carries out an optimization operation to fine-tune the alignment frame output by the geometric hashing algorithm. Experimental results show that, with the refinement process, we can adopt a stricter criterion in the filtering process to reduce the number of residues on the contour of the query protein that can pass. The exact parameter settings employed in Protemot to realize the stricter criterion are provided in the Supplementary Data. As mentioned earlier, the time complexity of the geometric hashing algorithm that matches the substructure of the query protein with a template is $O(n_1 n_2 (n_1+n_2))$, where n_1 is the number of residues in the template and n_2 is the number of the residues in the query protein that pass the filtering process. Therefore, by reducing the number of residues on the contour of the query protein that can pass the filtering process, we can expedite the structural alignment process substantially. Table 2 reports the effects achieved in five cases. In each case, a query protein was submitted to the Protemot web server and the execution times observed with/without the refinement process were recorded. Basically, speedups of >10 times are generally observed.

Table 1. A statistics of structural similarity between the templates in the Protemot library and those in the CSA library

	Number of templates in the CSA library for which a match in the following categories is found in the Protemot library
Highly probable	73
Probable	17
Possible	15
Unlikely	42
Total number of templates in the CSA library	147

Table 2. Speed up achieved with the refinement process

PDB ID of the query protein	Execution time (in seconds) without the refinement process	Execution time (in seconds) with the refinement process	Speed up
1BCK	4520	441	10.25
1A46	9312	733	12.70
1AAW	14000	1064	13.16
1TRN	14788	1138	12.99
2HGS	18240	1330	13.71

INPUT AND OUPUT

Figure 3 shows the user interface of the Protemot web server. The user only needs to either upload the tertiary structure of the query protein in the PDB format or enter a PDB ID. There is another optional field, in which the user can specify the portion of the protein structure that is of particular interest. If the user does not fill this field, then Protemot will search the entire contour of the protein tertiary structure for possible match.

The structures of the query protein and the identified template will be superimposed and stored in a PDB file. As shown in Figure 4, this PDB output can be rendered with

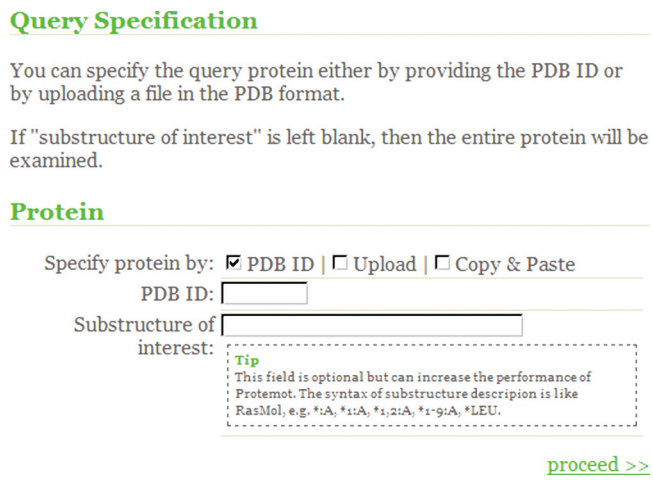

Figure 3. The query page of the Protemot web server.

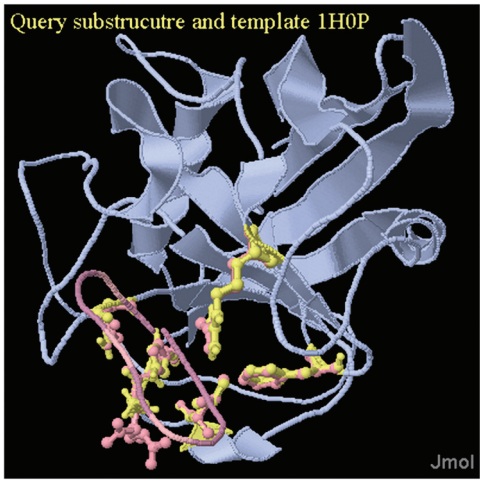

Figure 4. An example output of the Protemot web server, in which yellow balls are the residues in the query protein (1BCK) that matches the template; pink balls are the residues of the template.

Jmol (available at http://www.jmol.org/), which is embedded in this server. Users can also download the PDB file for their preferred visualization tools.

EVALUATION

This section reports the experiment conducted to evaluate the prediction power of Protemot. It has been observed in the experiment that, owing to the enriched collection of templates, Protemot can deliver a superior prediction power than a web server based on a manually curated template library, which normally contains much fewer entries.

The evaluation has been conducted with a comparison against the prediction power delivered by the web server based on the well-known CSA maintained with manual intervention (3). The CSA-based web server is located at http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/CSS/makeEbiHtml.cgi?file=form.html. Because the CSA-based web server was designed for prediction of the active sites of enzymes, the experiment has been designed accordingly.

In the experiment, the template library of Protemot contains a total of non-redundant 1051 entries automatically extracted from a total of 12 783 enzyme crystal structures in PDB. The template library covers 78% of those third-level, sub-subclass level, EC (Enzyme Classification) numbers present in PDB and 55% of those fourth-level, the most detailed level, EC codes present in PDB. On the other hand, the web version of CSA contains only 147 templates that were extracted with manual intervention. The testing dataset used in the experiment contains a total of 1000 non-redundant, randomly selected enzyme structures distributed over 587 fourth-level EC codes. Care has been taken to guarantee that the testing dataset does not contain any of those enzyme structures from which the templates were extracted.

In the experiment, if the query enzyme contains no substructures that can pass the matching criteria elaborated in the previous section when compared with all the templates in Protemot, then no prediction is given. On the other hand, if the query enzyme contains a substructure that can pass the matching criteria when compared with one particular template, the query enzyme is predicted to be associated with the same EC number as the enzyme from which the template was extracted. In case the query enzyme contains a substructure that can pass the matching criteria when compared with two or more templates, then the multiple predictions will be ranked based on the scoring function employed to rank the alternative alignment frames output by the geometric hashing algorithm.

Tables 3 and 4 show how Protemot performs in comparison with the CSA-based web server. In these two tables, a prediction is said to be correct only when the predicted EC number that is highest ranked matches the answer, which is a relatively strict criterion. Meanwhile, there are cases in which the web server failed to make a prediction due to lack of a similar template in the library. In the following discussion, the actions made by the web server in those cases are not considered as correct ones, since each of the 1000 testing enzymes is associated with an EC number. Table 3 reports the experimental results when the fourth-level EC codes are used as the answers. On the other hand, Table 4 reports the experimental results when the third-level EC codes are used as the answers. There are two columns in these two tables that report the experimental results with the CSA-based web server. The statistics listed in the column under CSA (highly probable + probable + possible) was obtained by treating the predictions that the CSA-based web server classifies as 'unlikely' as 'no match'. In other words, with this criterion, we refuse to trust a prediction made by the CSA-based web server, if the prediction is classified as 'unlikely'. On the other hand, the statistics listed in the column under CSA (highly probable + probable) was obtained by treating the predictions that the CSA-based web server classifies as 'unlikely' or 'possible' as 'no match'.

Table 3. Comparison of how Protemot and the CSA-based web server perform based on the fourth-level EC codes

	CSA (highly probable + probable + possible)	CSA (highly probable + probable)	Protemot	Overlap between Protemot and CSA (highly probable + probable)
Number of testing enzymes	1000	1000	1000	
The template library contains a template that is extracted from a protein–ligand complex structure with the same fourth-level EC code as the query enzyme and the web server makes a correct prediction.	81	75	408	44
The template library contains a template that is extracted from a protein–ligand complex structure with the same fourth-level EC code as the query enzyme but the web server makes an incorrect prediction.	61	8	310	0
The template library contains a template that is extracted from a protein–ligand complex structure with the same fourth-level EC code as the query enzyme but the web server makes no prediction.	4	63	14	1
The template library does not contain a template that is extracted from a protein–ligand complex structure with the same fourth-level EC code as the query enzyme and the web server makes no prediction.	65	777	14	13
The template library does not contain a template that is extracted from a protein–ligand complex structure with the same fourth-level EC code as the query enzyme but the web server makes a prediction, which is certainly incorrect.	789	77	254	28

Table 4. Comparison of how Protemot and the CSA-based web server perform based on the third-level EC codes

	CSA (highly probable + probable + possible)	CSA (highly probable + probable)	Protemot	Overlap between Protemot and CSA (highly probable + probable)
Number of testing enzymes	1000	1000	1000	
The template library contains a template that is extracted from a protein–ligand complex structure with the same third-level EC code as the query enzyme and the web server makes a correct prediction.	143	118	514	80
The template library contains a template that is extracted from a protein–ligand complex structure with the same third-level EC code as the query enzyme but the web server makes an incorrect prediction.	531	28	447	11
The template library contains a template that is extracted from a protein–ligand complex structure with the same third-level EC code as the query enzyme but the web server makes no prediction.	47	575	26	14
The template library does not contain a template that is extracted from a protein–ligand complex structure with the same third-level EC code as the query enzyme and the web server makes no prediction.	22	265	2	2
The template library does not contain a template that is extracted from a protein–ligand complex structure with the same third-level EC code as the query enzyme but the web server makes a prediction, which is certainly incorrect.	257	14	11	1

In the following discussion, we will employ two indexes to measure the overall prediction power of the web server. The first index simply measures the number of correct predictions made by the web server. As shown in Tables 3 and 4, the CSA-based web server can only correctly predict the binding sites of <15% of the total of 1000 testing enzymes, regardless of which criterion is adopted to examine the results. On the other hand, Protemot is able to correctly predict the binding sites of over 40% or 50% of the testing enzymes, depending on which level of the EC codes is used as the answers. Protemot has been able to make more correct predictions because its template library contains a lot more entries extracted from the enzyme structures in PDB than the web version of CSA, 1051 versus 147, owing to the automatic extraction mechanism.

The second index employed to measure the prediction power of the web server reflects the confidence level that the user can have on the predictions made by the web server. This index bears a similar notion of the positive predictive value (18) or precision (19) and is defined as follows:

$$\text{confidence} = \frac{\text{correct\_predictions}}{\text{correct\_predictions} + \text{incorrect\_predictions}}.$$

As shown in Table 3, for the total of 1000 testing enzymes, Protemot made 408 correct predictions, 310 + 254 = 564 incorrect predictions. Therefore, the confidence level delivered by Protemot was 408/(408 + 564) = 41.98%. On the other hand, in the column under CSA (highly probable + probable), there are 75 correct predictions, 8 + 77 = 85 incorrect predictions. Accordingly, the confidence level delivered by the CSA-based web server was 46.88%. By the same definition, the confidence level corresponding to the column under CSA (highly probable + probable + possible) was only 8.70%. If we employ the harmonic mean of the two indexes to measure the overall prediction power of the web server, which is a normal procedure adopted in information retrieval research (20), the overall index value for Protemot will be 41.38%. On the other hand, the overall index values corresponding to CSA (highly probable + probable) and CSA (highly probable + probable + possible) will be 12.93 and 8.39%, respectively. If we used the third-level EC codes as answers, then according to the numbers in Table 4, the overall index value for Protemot will be 52.13%. On the other hand, the overall index values corresponding to CSA (highly probable + probable + possible) and CSA (highly probable + probable + possible) will be 20.34 and 14.81%, respectively.

The experimental results reveal that the quantity of templates is crucial for enhancing the prediction power of the structure-based binding sites predictor such as Protemot. It is likely that binding sites predictor based on a manually curated library can achieve superior prediction power to Protemot, if the template library contains sufficient quantity of templates. However, as the number of entries in PDB continues to grow exponentially, automatic mechanism is essential for timely accommodating to new PDB releases and facilitating the functional annotation of proteins with unknown functions.

CONCLUSION

In this paper, a web server designed for prediction of protein-binding sites with an automatically extracted template library is presented. As shown in the experimental results, owing to the automatic extraction mechanism, the template library of Protemot contains substantially more entries than the CSA. The direct implication is that the quantity of the templates is crucial for making the web server more powerful in predicting the binding sites of proteins with unknown functions.

Though Protemot has been able to provide a highly desirable service, there still exists a large room for future

improvement with respect to its prediction accuracy. In this regard, our hypothesis is that prediction accuracy can be improved if the templates contain not only the geometric features of the protein binding sites but also the physicochemical features. Accordingly, continuous investigation will be made to perfect the design of Protemot in the future.

SUPPLEMENTARY DATA

Supplementary data are available at NAR Online.

ACKNOWLEDGEMENTS

This research has been supported by the National Science Council of Taiwan for funding under the contracts NSC 94-2627-B-002-007, NSC 9494-2627-B-002-008, and NSC 94-2627-B-001-004. Funding to pay the Open Access publication charges for this article was provided by National Taiwan University.

Conflict of interest statement. None declared.

REFERENCES

1. Brenner,S.E. (2001) A tour of structural genomics. *Nature Rev. Genet.*, **2**, 801–809.
2. Pellegrini,M. (2001) Computational methods for protein function analysis. *Curr. Opin. Chem. Biol.*, **5**, 46–50.
3. Torrance,J.W., Bartlett,G.J., Porter,C.T. and Thornton,J.M. (2005) Using a library of structural templates to recognise catalytic sites and explore their evolution in homologous families. *J. Mol. Biol.*, **347**, 565–581.
4. Ferrè,F., Ausiello,G., Zanzoni,A. and Helmer-Citterich,M. (2005) Functional annotation by identification of local surface similarities: a novel tool for structural genomics. *BMC Bioinformatics*, **6**, 194.
5. Shulman-Peleg,A., Nussinov,R. and Wolfson,H.J. (2005) SiteEngines: recognition and comparison of binding sites and protein–protein interface. *Nucleic Acids Res.*, **33**, W337–W341.
6. Chang,D.T.-H., Chen,C.-Y., Chung,W.-C., Oyang,Y.-J., Juan,H.-F. and Huang,H.-C. (2004) ProteMiner-SSM: a web server for efficient analysis of similar protein tertiary substructures. *Nucleic Acids Res.*, **32**, W76–W82.
7. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
8. Laskowski,R.A., Chistyakov,V.V. and Thornton,J.M. (2005) PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids. *Nucleic Acids Res.*, **33**, D266–D268.
9. Li,W., Jaroszewski,L. and Godzik,A. (2001) Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283.
10. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J.H., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
11. Oyang,Y.J., Hwang,S.C., Ou,Y.Y., Chen,C.Y. and Chen,Z.W. (2005) Data classification with radial basis function networks based on a novel kernel density estimation algorithm. *IEEE Trans. Neural Netw.*, **16**, 225–236.
12. Wolfson,H.J. and Rigoutsos,I. (1997) Geometric hashing: an overview. *Comput. Sci. Eng. IEEE*, **4**, 10–21.
13. Orengo,C.A. and Taylor,W.R. (1996) SSAP: Sequential structure alignment program for protein structure comparison. *Methods Enzymol.*, **266**, 617–635.
14. Pennec,X. and Ayache,N. (1998) A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics*, **14**, 516–522.
15. Boutonnet,N.S., Rooman,M.J., Ochagavia,M.E., Richelle,J. and Wodak,S.J. (1995) Optimal protein-structure alignments by multiple linkage clustering—application to distantly related proteins. *Protein Eng.*, **8**, 647–662.
16. Zhang,Z. (1994) Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, **13**, 119–152.
17. Altschul,S.F. (1991) Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, **219**, 555–565.
18. Speicher,C.E. and Smith,J.W. (1983) *Choosing Effective Laboratory Tests*. W. B. Saunders, Philadelphia, PA.
19. Manning,C.D. and Schütze,H. (1999) *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.
20. Kenney,J.F. and Keeping,E.S. (1962) *Mathematics of Statistics*. Van Nostrand. New York.