

PRUNE and PROBE—two modular web services for protein–protein docking

Pralay Mitra and Debnath Pal*

Bioinformatics Centre and Supercomputer Education Research Centre, Indian Institute of Science, Bangalore 560 012, India

Received February 16, 2011; Revised April 19, 2011; Accepted April 20, 2011

ABSTRACT

The protein–protein docking programs typically perform four major tasks: (i) generation of docking poses, (ii) selecting a subset of poses, (iii) their structural refinement and (iv) scoring, ranking for the final assessment of the true quaternary structure. Although the tasks can be integrated or performed in a serial order, they are by nature modular, allowing an opportunity to substitute one algorithm with another. We have implemented two modular web services, (i) PRUNE: to select a subset of docking poses generated during sampling search (<http://pallab.serc.iisc.ernet.in/prune>) and (ii) PROBE: to refine, score and rank them (<http://pallab.serc.iisc.ernet.in/probe>). The former uses a new interface area based edge-scoring function to eliminate >95% of the poses generated during docking search. In contrast to other multi-parameter-based screening functions, this single parameter based elimination reduces the computational time significantly, in addition to increasing the chances of selecting native-like models in the top rank list. The PROBE server performs ranking of pruned poses, after structure refinement and scoring using a regression model for geometric compatibility, and normalized interaction energy. While web-service similar to PROBE is infrequent, no web-service akin to PRUNE has been described before. Both the servers are publicly accessible and free for use.

INTRODUCTION

Docking provides a mechanistic understanding of protein–protein interaction allowing fundamental insight to the researchers. The field of docking has received increasing attention due to its primary role in studying protein interaction networks. However, in absence of experimental or evolutionary information, protein docking

is difficult due to the low free energy of biological complexes in general, as well as the computational demands of time to execute the docking scheme to arrive at the putative answers. A number of docking methods have been developed over the past three decades (1). Some of them have also been implemented as free web service (2–10). All docking methodologies typically perform four major tasks: (i) generation of docking poses, (ii) selecting a subset of poses, (iii) their structural refinement and (iv) scoring, ranking for the final assessment of the true quaternary structure. These tasks can be integrated or performed in a serial order; however, they are by nature modular allowing an opportunity to substitute one algorithm with another. This allows us to take the best components from one method and use it as input into another to process and provide an output that has a significant improvement in performance, both in the time-taken and the accuracy. This is a different paradigm than the meta servers, which take output from different servers to provide a filtered final output; here we combine components from various methods to improve the final outcome.

One of the tasks that make protein–protein docking unwieldy for common use is the generation of a large number of docking poses. Although attempts have been made to use directed search (11) or pseudo-random methods (12) to reduce the search space, the structural refinement of poses and the integrated or edge-scoring strategy can be further improved. For example, multi-parametric scoring functions require a considerable computing time and reduction of parametric space without losing efficacy can be of significant advantage. We have recently shown that a single parameter edge-scoring function can select a subset of poses from a large pool of unrefined docking poses arising out of exhaustive sampling search (13). Only *interface area* (IA) is sufficient to screen the poses and can be used as simple rule-based edge-scoring function using the proposition that native like poses must have largest to near the largest interface areas. This edge-scoring function can be integrated into any docking search scheme to eliminate >95% of the poses generated during the docking search. Our single

*To whom correspondence should be addressed. Tel: +918022932901; Fax: +918023600551; Email: dpal@serc.iisc.ernet.in

parameter-based pruning technique reduces the computational time significantly, yet increases the chances of selecting native-like models in the top rank list. This unique method, named PRUNE, has been benchmarked (13) against PatchDock (6) and FireDock (14) program, two state-of-the-art methods, widely used for protein–protein docking. In this article, we provide additional results by benchmarking against docking search inputs from FTDock (15), ZDock (8) and Gramm-X (5) as well.

To extend the advantages of the modularly designed method of PRUNE, we have developed another modular method, protein binding evaluation (PROBE), which can work on a subset of docking poses by optimizing their side-chain contacts to score and rank them using a regression model for geometric compatibility [based on two highly correlated geometric parameters (16)], and normalized interaction energy (calculated from correlated non-bonded and solvation energies). This simple-scoring function has been benchmarked on state-of-art predictive docking methods and has been shown to perform either equally well or much superior to their sophisticated counterparts (17). We provide additional comparison of PROBE results on a test data set against top two successful servers in the recent CAPRI evaluations (18). The kind of service provided by PROBE is infrequent: FireDock (3) and FiberDock (10), are the only two sister web services we are aware of that allows upload of a subset of docking poses for ranking, although a number of other softwares exist for re-ranking docking poses in local machine (14,19–22).

Here we describe the web servers that we have developed using the PRUNE and PROBE algorithms. We briefly discuss their methods and describe their modular usage. These are further elaborated in the Help pages and Tutorials depicted at the web site. The execution time usually takes a few hours. The job status is available in the browser and notifications are sent by email, if email address is provided at the job submission page. The servers PRUNE (<http://pallab.serc.iisc.ernet.in/prune>) and PROBE (<http://pallab.serc.iisc.ernet.in/probe>) are publicly available and free for use.

PRUNE: A RULE-BASED METHOD TO SELECT SUBSET OF DOCKING POSES CONTAINING NATIVE-LIKE MODELS

The output of a docking sampling search results in generation of new docking poses that are described by the transformation matrix applied on the initial coordinates of the docking partners. In PRUNE, we only calculate the IA of the unrefined docking poses generated between the reference (static) and the mobile molecule (13). We define IA as the accessible surface area (ASA) of individual subunit buried on complex formation. For IA, ASA of all the atoms in the individual subunits and that of the complex are first calculated as per definition of Lee and Richards (23) using a default probe radius of 1.4Å. An atom is thereafter defined to be an *interface atom* if it loses its ASA by $>0.1\text{Å}^2$ upon complex formation. The summation of the loss of ASA by all the interface atoms divided by 2 is the IA for the dimer. After computing IA of all docking

poses, a histogram of 50 bins based upon IA is drawn. A sixth-order polynomial $[f(x)]$ is fitted on this histogram. After that the saddle points of the polynomial $f(x)$ are determined. Proceeding towards increasing absolute values of x for the polynomial $f(x)$, the first two saddle points (I_1 and I_2) are chosen, where I_1 is the saddle point corresponding to the highest frequency value in the histogram and I_2 is the immediate next saddle point. The bin number at which the straight line joining I_1 and I_2 intersects X-axis is chosen as cut-off point (C_p). If I_2 cannot be computed, then C_p is chosen as double of I_1 . The poses belonging to bin number $\geq C_p$ is selected as the subset of poses suitable for final scoring and ranking. The final subset of docking poses has largest and near-largest interface areas, chosen using the dynamically computed parameter C_p .

The underlying rule as described above and used in the PRUNE server has been extensively tested and benchmarked (13). Briefly, the tests were done on 922 bound, and 77 unbound binary docking targets covering 193–7658Å² interface ranges. The unbound data set was further divided into three categories: rigid body, medium and difficult, as per the definition of Mintseris *et al.* (24). Sampling at 12° rotation for the bound cases returned 91% cases, where at least one near-native docking pose could be retrieved in the pruned subset. A near-native pose was defined using the ‘acceptable’ (10 Å L_{RMSD}) criterion of Mendez *et al.* (25). Sampling at 12° rotation for the unbound data set returned only 68% cases with near-native poses. We, therefore, sampled the data set using 9° rotation which improved the success to 83%. This suggested that the chance of locating a near native pose in the pruned subset increased with lowering of the rotation step size. Comparative tests on rank improvement by FireDock scoring on the subsets of top 1000 poses selected by PatchDock program and our method showed that for 61 unbound rigid body docking cases, our subset gave improved rank in 2-fold more cases than PatchDock input; for the 10 medium category cases the comparative improvement was in ~3-fold cases and in six difficult category cases there were no winners. The ability of the FireDock program to return a correct result within top 10 rank is increased by at least 2-folds if the subset of poses obtained by our method was used, in contrast to use of all the docking poses available from the sampling search.

PRUNE SERVER: INPUT, OUTPUT AND USER INTERFACE

Input and output

The PRUNE server requires four inputs: (i) receptor coordinate file (Protein Data Bank, PDB format), (ii) ligand coordinate file (PDB format), (iii) a file containing the transformation matrices for generating docking poses and (iv) information on the source of the file format of the transformation matrices. Currently, four different file formats for transformation matrices from FTDock (15), ZDock (8), PatchDock (6) and Gramm-X (5) are supported by the server. All the inputs are mandatory as they are used to recreate the three-dimensional

representation of the docking poses and compute their IA. A typical example executed in a 2.3 GHz single CPU workstation for pruning 10 000 poses takes 5 min for generation of the protein-complex coordinates and another 76 min for pruning based on interface area, when the two molecules are of chain length 294 and 76. IA is computed using the NACCESS program (26), which takes 1–2 sec per protein molecule depending on the size of the complex.

The result page outputs transformation matrices of only those poses that lie beyond the cut-off point (C_p) of the IA distribution, in the same format as the input. The user may download the information on the pruned poses and use it for ranking by any software of his/her choice. Alternately, a single button click option is also provided at the PRUNE result page to score and rank using our scoring method by forwarding the pruned poses to the PROBE server. Example size of the pruned subsets and the percentage of near native poses obtained from PRUNE server is shown in Table 1. These can be compared against the total number of near native poses available from the docking search. As can be seen, for all the targets in the test set, a small subset of the total number of poses is selected which contain at least one near native pose.

PROBE FOR SCORING AND RANKING DOCKING POSES

The subset of docking poses available after the use of edge- or integrated-scoring function during a docking

search is suitable for use with the PROBE algorithm (17). The subset of docking poses received by PROBE are structurally refined using the Restricted Side-Chain Optimization (RISCO) algorithm which includes a step of Monte Carlo based rigid-body optimization (14). However, since a geometry optimization does not guarantee that all steric clashes be eliminated, we use a pseudo-scaling procedure to eliminate spurious estimation of energy/ASA values. It may be noted that both van der Waals and Coulomb interaction makes significant contribution at closer than canonical contact distances and the pseudo-scaling procedure eliminates these potential errors. The same is true, when ASA is used for estimation of solvation energy. The final scoring in PROBE is based on four physiochemical parameters: interface packing (IP) (16), surface complementarity (SC) (16), pseudo-scaled non-bonding energy (NE) and solvation energy (SE). The SC and IP value of the docking pose is used to calculate a distance surface complementarity/interface packing (SP) from an expected SC and IP value representing a geometric model, derived from the linear regression curve fitted on SC and IP values from a non-redundant data set of interfaces (mathematically, $SP = |SC - IP \times 0.6547 - 0.1495|$). The poses are grouped that are alike within $\leq 1.0 \text{ \AA}$ root-mean-square deviation (RMSD) and the difference of SP within < 0.04 . Representative poses for each group are chosen and group-based SP, NE and SE values calculated. The NE and SE values are mapped on to a normalized grid and their values are used together along with SP in a scoring function to compute the scores.

Table 1. Performance of PRUNE showing the percentage of poses selected and the near natives^a isolated when a list of docking poses derived from four different docking methods are input

Target	Docking		Total number of near native poses generated				Percentage cases after pruning								
	Partner 1 PDB_ Chain	Partner 2 PDB_ Chain	FTDock ^b	ZDock ^c	PatchDock ^d	Gramm-X ^e	FTDock ^b		ZDock ^c		PatchDock ^d		Gramm-X ^e		
							Poses	Near native	Poses	Near native	Poses	Near native	Poses	Near native	
1J2J_AB ^f	605	1O3Y_A	1OXZ_A	33	850	22	18	3.42	0	0.61	0.4	0.94	0	0.66	0
1EAW_AB ^g	745	1EAX_A	9PTI_A	41	527	25	70	2.20	14.6	2.25	2.1	1.17	0	14.14	17.1
Capri 8	853			2	33	17	0	2.55	0	0.46	9.1	0.36	11.8	1.89	X
1Z0K_AB ^f	893	2BME_A	1YZM_A	68	1105	24	12	5.60	10.3	0.27	0.1	0.79	0	5.64	8.3
1CLV_AI ^h	1042	1JAE_A	1QFD_A	60	3020	40	41	3.66	55.0	0.84	7.7	4.47	80.0	3.26	100
1GPW_AB ^f	1049	1THF_D	1K9V_F	24	209	33	26	2.20	45.8	3.52	77.5	4.88	90.9	3.21	38.5
Capri 6	1086			19	188	4	0	1.00	26.3	3.41	55.9	4.30	100	3.09	X
1UGH_EI ⁱ	1096	2SSP_E	1UDI_I	30	666	32	11	2.24	40.0	2.22	36.0	2.33	31.3	0.43	0
Capri 26	1166			22	64	18	3	3.30	27.3	0.56	1.6	0.56	0	0.50	0
2G77_AB ^h	1262	1FKM_A	1Z06_A	20	400	26	8	2.72	30.0	2.78	13.3	2.07	30.8	2.08	25.0
1DFJ_EI ^g	1291	9RSA_B	2BNH_A	4	178	0	1	2.93	67.0	6.47	100	2.37	X	2.53	100
1KXP_AD ^g	1671	1IJJ_B	1KW2_B	3	151	0	0	2.31	100	4.33	86.1	1.07	X	3.09	X

^aNear natives are defined as per the 'acceptable' criteria of Mendez *et al.* (25).

^b12° rotation: 27 720 total poses.

^c6° rotation: 54 000 total poses.

^dUsing default parameters at PatchDock web server: <http://bioinfo3d.cs.tau.ac.il/PatchDock/>.

^e10° rotation: 10 000 total poses.

^fRef. (29).

^gRef. (24).

^hRef. (28).

ⁱRef. (27).

X means that there were no near native pose in the total set of poses generated, so no near natives could be pruned.

The scores are sorted to compute ranks of representative poses. Data showing performance of PRUNE+PROBE on 12 unbound targets chosen arbitrarily—three from different CAPRI targets (<http://www.ebi.ac.uk/msd-srv/capri/>), one from Bernauer *et al.* (27), two from the recently published Benchmark 4.0 (28) and the rest from Benchmark 2.0 (24) and 3.0 (29) are given in Table 2. The *P*-values of the poses calculated using hypergeometric distribution indicate a high confidence of the top ranked predictions (30). Comparison shows that the overall ranking performance of ClusPro (31) is better, but PROBE is able to give better results in cases where ClusPro underperforms or fails. Interestingly, the PROBE performance competes strongly with FiberDock, despite use of much simpler scoring function and methodologies.

PROBE SERVER: INPUT, OUTPUT AND USER INTERFACE

Input and output

The primary task of PROBE server is to take a subset of docking poses and rank them after refining and scoring. Therefore, the inputs are designed to cater to two situations: (i) where the user has subset of poses generated by some docking search technique and he/she wants to rank them or, (ii) the user has two unbound protein molecules to dock them. For all uploaded coordinate files, a number of preprocessing steps are performed, such as, checking the minimum peptide length of 25, removal of solvent molecules—like water, resolving the atoms with

Table 2. Comparative performance of PROBE on select unbound targets^a

Target PDB	Rank ^b			PROBE <i>P</i> -value
	ClusPro ^c	FiberDock ^d	PROBE ^e	
1J2J	14	No result	12	0.179
IEAW	2	41	1	0.010
Capri 8	No result	No result	19	0.011
1Z0K	12	28	5	0.098
1CLV	1	8	2	0.109
1GPW	2	1	3	0.011
Capri 6	1	45	1	0.003
1UGH	2	29	7	0.083
Capri 26	1	22	63	0.072
2G77	3	19	23	0.157
1DFJ	2	No result	7	0.023
1KXP	1	30	1	0.003

^aThe PROBE scoring and ranking was done on the pruned subsets as depicted in Table 1.

^bRank of top 'acceptable' solution as per Mendez *et al.* (25).

^cTo dock the proteins, the coordinate files of the subunits were uploaded to the ClusPro web server: <http://cluspro.bu.edu/>.

^dThe docking partners were uploaded in PatchDock web server for docking. The maximum permissible top 100 poses output by the PatchDock web server was scored and ranked using FiberDock web server (<http://bioinfo3d.cs.tau.ac.il/FiberDock/>).

^eThe poses were generated locally by the ZDock program using 6° rotational sampling and pruned using the PRUNE server. The pruned subset of poses were scored and ranked using the PROBE server.

multiple occupancy, or conversion of atoms under amino acid type from HETATM MSE to ATOM MET (if the user wish to do so) as per PDB file format. If the first option is selected then additionally the file with transformation matrix information is needed to recreate the docking poses. The format of the matrix file also needs to be mentioned using drop down box (currently supported formats are: FTDock, ZDock, PatchDock and Gramm-X). For the second option, the server uses FTDock module (15) to generate the docking poses. The default rotation sampling is set to 12° (default of FTDock module) for generating the poses, but the user can avail other choices, such as 9°, 15° and 18°. The time taken for docking-pose generation increases as the rotation step size is decreased. We, therefore, strongly recommend that the user generate their own docking poses and upload the same for scoring and ranking by PRUNE+PROBE. In any case, users wishing to avail denser sampling (<9°) need to generate the docking poses locally in their own system. By default, the server returns the top 10 predictions, but the user may change it.

The output of the PROBE server is a rank-sorted list of complexes with details on individual parameters, and the final PROBE score values. Hyperlinks are provided to download the coordinates of receptor and the ligand files. A summary of results in tabulated form is also provided (Figure 1). The coordinates of these files are post-processed versions of the original PDB files uploaded by the user. Each predicted complex can also be visualized using the Jmol software (<http://jmol.sourceforge.net/>) using java applet. For this the Java software has to be installed in the user's system. The input/output contents are similar to the FireDock and FiberDock server, although derived from different scoring functions.

IMPLEMENTATION

The server is designed using C, Perl, Python, FORTRAN, HTML and PHP. The front end of the server is designed in PHP (version 5.2.9) and HTML. Perl has a very strong string-handling features, so it has been used for initial preprocessing of the protein coordinate files. The core of the method has been implemented in C language. The low-level features of C language helps to optimize and parallelize the code wherever needed. The parallel code in C language uses Message Passing Interface (MPI)—an application programming interface which allows many computers to interact with each other and thus distribute the load of the job. The computation takes place in a server which is a Linux cluster with AMD64 bit Opteron system.

The pruning of docking poses designed for PROBE and PRUNE server is implemented in Python (version 2.5.2). The Python code first fits the sixth-order polynomial on IA histogram generated by C language and then computes the saddle points to draw a straight line. The cut-off point (C_p) on X-axis is then returned to C program for further computation. The IA calculations are done using FORTRAN code.

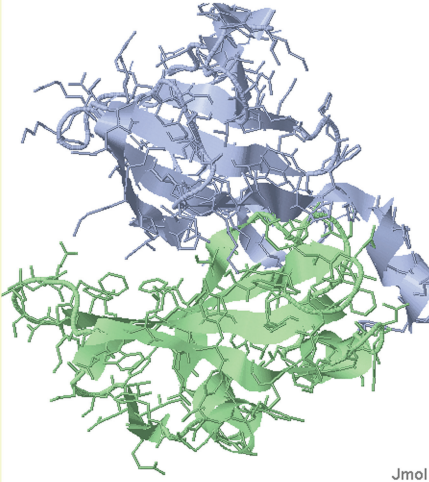
A

Result of PROBE							
	<i>Receptor Molecule</i>		<i>Ligand Molecule</i>		<i>PROBEresult</i>		
Rank	Interface Area	Interface Packing	Surface Complementarity	Nonbonded Energy	Solvation Energy	PROBE Score	View
Rank 1	2070	0.14	0.23	-1.795	13515.02	0.10	<input type="button" value="Jmol"/>
Rank 2	903	0.20	0.28	-0.861	5857.58	6.41	<input type="button" value="Jmol"/>
Rank 3	689	0.21	0.28	-0.674	4615.52	7.86	<input type="button" value="Jmol"/>
Rank 4	1150	0.18	0.28	-0.953	3730.87	7.90	<input type="button" value="Jmol"/>
Rank 5	972	0.22	0.29	-1.029	1386.34	8.09	<input type="button" value="Jmol"/>
Rank 6	925	0.18	0.23	-0.849	2979.59	8.19	<input type="button" value="Jmol"/>
Rank 7	919	0.20	0.32	-0.803	4581.25	8.26	<input type="button" value="Jmol"/>
Rank 8	701	0.26	0.38	-0.799	3961.97	8.34	<input type="button" value="Jmol"/>
Rank 9	868	0.20	0.34	-0.881	2676.89	8.38	<input type="button" value="Jmol"/>
Rank 10	808	0.19	0.34	-0.674	4382.62	8.40	<input type="button" value="Jmol"/>

[Download all complexes](#)

B

Rank 1



Jmol

Figure 1. The PROBE result page. (A) The table contains five physicochemical parameters (column second–sixth) as computed by the PROBE server at the protein–protein interface. These parameters are used to calculate the PROBE score to determine the ranks. First column which indicates the rank of the predicted complex is hyperlinked to allow download of coordinates and the button at last column is for Jmol-visualization of the complex. A compressed file containing all the predicted complexes can be downloaded as well. A summary table can be retrieved from hyperlink ‘PROBEresult’. (B) The visualization of the complex is available at the browser. The title indicates the rank of the complex.

CONCLUSION

Two modular web services, PRUNE and PROBE are described for protein–protein docking. The web services will allow the users to try out a variety of methods to improve their predictions. The advantage of such a web service is that it can also be continually upgraded to include new methods. PRUNE is a unique web-service that uses a single parameter, interface-area-based edge function that has been shown to time-efficiently select subsets of docking poses for improved scoring and ranking of poses. PROBE on the other hand, is similar to the previously described FireDock and FiberDock server, using its own method for scoring and ranking for a subset of docking poses. It can also work as a standalone docking server using the FTDock program’s docking-pose generation module. The method has been shown to efficiently score docking poses using its simple scoring function.

ACKNOWLEDGEMENTS

The authors thank Gurunath M. Katagi and Sumanta Mukherjee for their help. P.M. thanks All India Council for Technical Education, New Delhi for the National Doctoral Fellowship.

FUNDING

Funding for open access charge: Department of Biotechnology, New Delhi, Government of India.

Conflict of interest statement. None declared.

REFERENCES

- Ritchie,D.W. (2008) Recent progress and future directions in protein-protein docking. *Curr. Protein Pept. Sci.*, **9**, 1–15.
- Macindoe,G., Mavridis,L., Venkatraman,V., Devignes,M.D. and Ritchie,D.W. (2010) HexServer: an FFT-based protein docking server powered by graphics processors. *Nucleic Acids Res.*, **38**, W445–W449.
- Mashiach,E., Schneidman-Duhovny,D., Andrusier,N., Nussinov,R. and Wolfson,H.J. (2008) FireDock: a web server for fast interaction refinement in molecular docking. *Nucleic Acids Res.*, **36**, W229–W232.
- Lyskov,S. and Gray,J.J. (2008) The RosettaDock server for local protein-protein docking. *Nucleic Acids Res.*, **36**, W233–W238.
- Tovchigrechko,A. and Vakser,I.A. (2006) GRAMM-X public web server for protein-protein docking. *Nucleic Acids Res.*, **34**, W310–W314.
- Schneidman-Duhovny,D., Inbar,Y., Nussinov,R. and Wolfson,H.J. (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Res.*, **33**, W363–W367.
- Comeau,S.R., Gatchell,D.W., Vajda,S. and Camacho,C.J. (2004) ClusPro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics*, **20**, 45–50.
- Chen,R., Li,L. and Weng,Z. (2003) ZDOCK: an initial-stage protein-docking algorithm. *Proteins*, **52**, 80–87.
- Neuvirth,H., Raz,R. and Schreiber,G. (2004) ProMate: a structure based prediction program to identify the location of protein-protein binding sites. *J. Mol. Biol.*, **338**, 181–199.
- Mashiach,E., Nussinov,R. and Wolfson,H.J. (2010) FiberDock: a web server for flexible induced-fit backbone refinement in molecular docking. *Nucleic Acids Res.*, **38**, W457–W461.
- Fischer,D., Lin,S.L., Wolfson,H.L. and Nussinov,R. (1995) A geometry-based suite of molecular docking processes. *J. Mol. Biol.*, **248**, 459–477.
- Goodsell,D.S. and Olson,A.J. (1990) Automated docking of substrates to proteins by simulated annealing. *Proteins*, **8**, 195–202.

13. Mitra,P. (2010) Algorithmic approaches for protein-protein docking and quaternary structure inference. *PhD Thesis*, Indian Institute of Science, Bangalore.
14. Andrusier,N., Nussinov,R. and Wolfson,H.J. (2007) FireDock: fast interaction refinement in molecular docking. *Proteins*, **69**, 139–159.
15. Gabb,H.A., Jackson,R.M. and Sternberg,M.J. (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. *J. Mol. Biol.*, **272**, 106–120.
16. Mitra,P. and Pal,D. (2010) New measures for estimating surface complementarity and packing at protein-protein interfaces. *FEBS Lett.*, **584**, 1163–1168.
17. Mitra,P. and Pal,D. (2011) Using correlated parameters for improved ranking of protein-protein docking decoys. *J. Comput. Chem.*, **32**, 787–796.
18. Lensink,M.F. and Wodak,S.J. (2010) Docking and scoring protein interactions: CAPRI 2009. *Proteins*, **78**, 3073–3084.
19. Liang,S., Meroueh,S.O., Wang,G., Qiu,C. and Zhou,Y. (2009) Consensus scoring for enriching near-native structures from protein-protein docking decoys. *Proteins*, **75**, 397–403.
20. Pierce,B. and Weng,Z. (2007) ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins*, **67**, 1078–1086.
21. Liang,S., Wang,G. and Zhou,Y. (2009) Refining near-native protein-protein docking decoys by local resampling and energy minimization. *Proteins*, **76**, 309–316.
22. Pierce,B. and Weng,Z. (2008) A combination of rescoring and refinement significantly improves protein docking performance. *Proteins*, **72**, 270–279.
23. Lee,B. and Richards,F.M. (1971) The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, **55**, 379–400.
24. Mintseris,J., Wiehe,K., Pierce,B., Anderson,R., Chen,R., Janin,J. and Weng,Z. (2005) Protein-protein docking benchmark 2.0: an update. *Proteins*, **60**, 214–216.
25. Mendez,R., Leplae,R., De Maria,L. and Wodak,S.J. (2003) Assessment of blind predictions of protein-protein interactions: current status of docking methods. *Proteins*, **52**, 51–67.
26. Hubbard,S. and Thornton,J.M. (1993) *NACCESS 2.1.1 ed.* Department of Biochemistry and Molecular Biology, University College, London.
27. Bernauer,J., Aze,J., Janin,J. and Poupon,A. (2007) A new protein-protein docking scoring function based on interface residue properties. *Bioinformatics*, **23**, 555–562.
28. Hwang,H., Vreven,T., Janin,J. and Weng,Z. (2010) Protein-protein docking benchmark version 4.0. *Proteins*, **78**, 3111–3114.
29. Hwang,H., Pierce,B., Mintseris,J., Janin,J. and Weng,Z. (2008) Protein-protein docking benchmark version 3.0. *Proteins*, **73**, 705–709.
30. Feliu,E. and Oliva,B. (2010) How different from random are docking predictions when ranked by scoring functions? *Proteins*, **78**, 3376–3385.
31. Kozakov,D., Hall,D.R., Beglov,D., Brenke,R., Comeau,S.R., Shen,Y., Li,K., Zheng,J., Vakili,P., Paschalidis,I. *et al.* (2010) Achieving reliability and high accuracy in automated protein docking: ClusPro, PIPER, SDU, and stability analysis in CAPRI rounds 13–19. *Proteins*, **78**, 3124–3130.