J Mol Model (2013) 19:3901–3910
DOI 10.1007/s00894-013-1916-7

ORIGINAL PAPER

An information-theoretic classification of amino acids for the assessment of interfaces in protein–protein docking

Christophe Jardin · Arno G. Stefani · Martin Eberhardt · Johannes B. Huber · Heinrich Sticht

Received: 4 May 2013 / Accepted: 9 June 2013 / Published online: 5 July 2013

Abstract Docking represents a versatile and powerful method to predict the geometry of protein–protein complexes. However, despite significant methodical advances, the identification of good docking solutions among a large number of false solutions still remains a difficult task. We have previously demonstrated that the formalism of mutual information (MI) from information theory can be adapted to protein docking, and we have now extended this approach to enhance its robustness and applicability. A large dataset consisting of 22,934 docking decoys derived from 203 different protein–protein complexes was used for an MI-based optimization of reduced amino acid alphabets representing the protein–protein interfaces. This optimization relied on a clustering analysis that allows one to estimate the mutual information of whole amino acid alphabets by considering all structural features simultaneously, rather than by treating them individually. This clustering approach is fast and can be applied in a similar fashion to the generation of reduced alphabets for other biological problems like fold recognition, sequence data mining, or secondary structure prediction. The reduced alphabets derived from the present work were converted into a scoring function for the evaluation of docking solutions, which is available for public use via the web service score-MI: http://score-MI.biochem.uni-erlangen.de

Keywords Protein interaction · Structure analysis · Reduced amino acid alphabet · Protein interface · Mutual information

C. Jardin · M. Eberhardt · H. Sticht (✉)
Bioinformatik, Institut für Biochemie,
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Fahrstrasse 17, 91054 Erlangen, Germany
e-mail: H.Sticht@biochem.uni-erlangen.de

A. G. Stefani · J. B. Huber
Lehrstuhl für Informationsübertragung,
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Cauerstrasse 7, 91058 Erlangen, Germany

Introduction

Protein–protein interactions play a central role in various aspects of the structural and functional organization of the cell. Their elucidation is crucial to understanding processes such as metabolic control, signal transduction, and gene regulation [1–6]. Large-scale studies using yeast two-hybrid assays or mass spectrometry provide an increasing list of protein–protein interactions [7–11]. However, experimental structural determination of all of them is impractical, and only a small fraction of the potential complexes will be amenable to direct experimental analysis. In this context, docking simulations help to predict in silico the structures of protein complexes [12, 13]. Protein docking simulations generate a large number of putative complex structures. The identification of correct solutions from this vast array of incorrect structures, however, remains a difficult task, and to date no general solution to this problem is available.

In a previous study [14], we used the concept of mutual information (MI) to identify those structural features that are particularly informative to distinguish between good and bad docking solutions. In particular, we focused on the different types of amino acid contacts present in the interfaces. In order to simplify the different types of contacts, and to increase the interpretability of the results, a reduced amino acid alphabet was generated, in which the 20 amino acids were grouped into four classes according to their biophysical properties. This strategy results in a total of ten different types of contacts in protein interfaces, for which the MI can be calculated. We also derived a strategy to convert the respective MI values into a scoring function for the identification of good docking solutions. This previous work demonstrated the general applicability of the approach, but was done for only a relatively small dataset and also lacked a systematic optimization of the reduced alphabets applied.

In the present work, we have improved the amino acid grouping of reduced alphabets by an iterative approach that uses clustering analysis and the MI as an objective function. In addition, a significantly larger dataset was generated as a basis for the clustering procedure. This new strategy allows more reliable identification of good docking solutions and is now available for public use via the web service score-MI: http://score-MI.biochem.uni-erlangen.de

Methods

Dataset of docked complexes

The dataset is based on 261 protein–protein complexes provided by Vakser et al. as unbound docking benchmarks 1.0 and 3.0 (http://dockground.bioinformatics.ku.edu/). For the generation of the docking decoys, we used the FTDock docking algorithm (version2.0) [15] as implemented in the 3D-Dock Suite. To generate a realistic docking scenario, the experimental structures of the two isolated (unbound) subunits were used whenever available (127 of the 261 docking cases). For the remaining 134 cases, docking started from the unbound conformation of one of the interaction partners and used the bound conformation for the second one.

For each of these 261 complexes 10,000 docking solutions were generated. To remain consistent with the selection procedure in our previous work [14], which was based on the Dockground 1.0 dataset [16–18], these solutions were classified as near-physiological (‘close’) or non-physiological (‘false’) according to the following criteria: close docking solutions are characterized by a ligand RMSD of less than 5 Å (for backbone Cα-atoms) to the correct complex geometry. The 203 complexes for which at least one close docking solution was generated by FTDock were included in subsequent analysis. For each of these 203 complexes, up to 20 close as well as the 100 top-scoring false docking solutions were included. The resulting dataset had 22,934 docking decoys, of which 2,634 were close and 20,300 were false docking solutions.

This dataset provided the basis for the generation of a residue-based interaction map of the interfaces. According to the concept of a residue-based potential [19], only one contact was counted per pair of interface residues. The distance of such a residue–residue contact was defined as the closest atomic distance between the amino acids involved. Interface residues are defined as having at least one atom in less than 7 Å distance from the docking partner [14]. This resulted in more than $3.4 \cdot 10^6$ interface contacts in the present dataset.

Calculation of the mutual information

Mutual information (MI) in information theory is a measure of the coupling between two random variables X and Y telling us what we can learn about X when we observe Y (and vice versa) [20]. The mutual information $I(X;\,Y)$ depends only on the probability distributions Pr(X) and Pr(Y) as well as their joint probability function Pr(XY), with M_x and M_y denoting the alphabet sizes of X and Y:

$$I(X;Y) = \sum_{i=1}^{Mx} \sum_{k=1}^{My} \Pr(X = x_i, Y = y_k) \log_2 \left(\frac{\Pr(X = x_i, Y = y_k)}{\Pr(X = x_i)\Pr(Y = y_k)} \right) \tag{1}$$

In the application of this concept to docking analysis, the binary random variable X expresses whether a docking solution is close or false, $X \in \{c, f\}$. The random variable Y can, for example, specify the number of contacts of a certain type in the interfaces of the close and false docking solutions. For this type of application, the information content of different structural features is usually assessed separately [14]. In the present work, the MI-formalism was extended to address a second question related to docking: The comparison of the performance of various reduced alphabets. Such alphabets group similar amino acids together in one class, thus increasing the statistical robustness of the predictions [21].

Clustering protocol

To assess the performance of different reduced alphabets, calculation of the overall information content of all descriptors is required, which was done by cluster analysis here. Cluster analysis is a means to group objects in such a way that the objects in one cluster are more similar to each other than to those in other clusters [22, 23]. This approach allows one to estimate the MI of a whole alphabet by assessing all underlying structural features simultaneously. Cluster analyses were done using the ELKI framework [24]—a knowledge discovery in databases (KDD, “data mining”) software framework. For our purpose, we used the wide-spread k-means clustering. Given a set of n observations (x_1, x_2, …, x_n), k-means clustering aims to partition the n observations into k clusters ($k \le n$) S={S_1, S_2, …, S_k} so as to minimize the total distance of the objects from their respective cluster center:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2, \quad \text{where } \mu_i \text{ is the mean of points in } S_{i.} \tag{2}$$

For k-means clustering, we used Lloyd’s algorithm [25], also known as Voronoi iteration, modified with k-means++ initialization [26]. Implicit weighting of the descriptors was prevented by in-descriptor normalization prior to clustering.

The MI of an alphabet in each clustering repeat was evaluated by assessing the MI of all k clusters simultaneously. In this clustering analysis, Y in Eq. 1 refers to the k clusters, and all structural descriptors are treated for all docking solutions simultaneously in one single clustering. In the present work, the structural descriptors evaluated are the number and type of pairwise amino acid contacts in the protein interfaces.

For clustering, a maximum of 250 iterations was allowed and the number of clusters was set to 20. The clustering was repeated 20 times to achieve statistical soundness. This number of repeats was high enough in pre-tests to keep the standard deviation below 15 % of the average $\overline{\mathrm{MI}}$ value (data not shown).

The maximum achievable MI ($\mathrm{MI}_{\mathrm{max}}$) depends on the frequencies of close and false solutions, which makes comparison between different datasets difficult. To obtain a normalization of the values, all MI values reported in this paper are given as percentage of $\mathrm{MI}_{\mathrm{max}}$, which is 0.51 in the present dataset. The corresponding measure was termed $\mathrm{MI}_{\mathrm{norm}}$.

Performance of the reduced amino acid alphabets

From each reduced amino acid alphabet resulting from the clustering procedure, a scoring function was derived using a previously established formalism [14] and tested in a five-fold cross-validation for its ability to discriminate close from false docking solutions. For cross-validation, the present dataset of 203 complexes was divided into five sets of almost equal sizes: three sets of 41 complexes and two sets of 40 complexes, with all the respective close and false decoys. Four of the five sets were combined into the training set and the performance was tested on the remaining set. This procedure was repeated five times, each time treating another of the five sets as the test set.

The performance of the scoring function was assessed by counting the close docking solutions among the three, five, or ten top-scoring solutions. To obtain a more realistic estimation of performance, their number was corrected by the number of close solutions that are expected to be found by chance on the same top x ranks using the following equation:

$$\text{enrichment in the top x positions} = \frac{\text{close solutions present in the top x positions}}{\text{close solutions expected by chance in the top x positions}} \qquad (3)$$

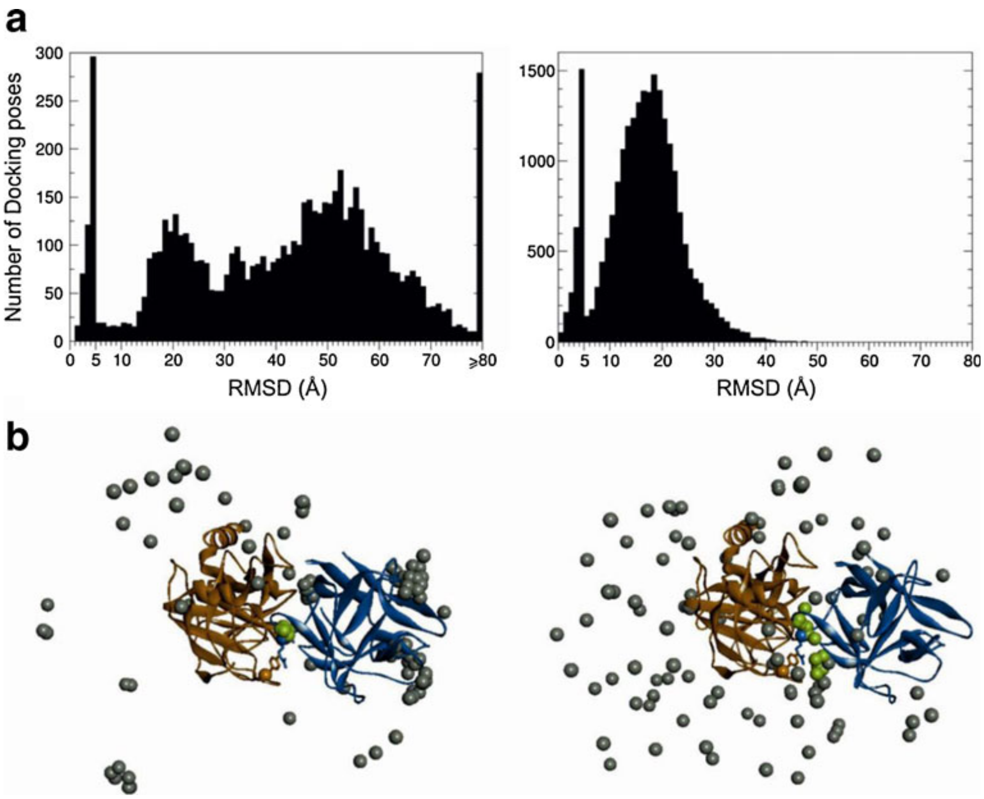


Fig. 1 Distribution of close and false docking solutions in different datasets. **a** Root mean square deviation (RMSD) distribution in the old dataset (*left*) used in [14] and in the new dataset (*right*). The RMSD values denote the deviation from the native complex structure. In both datasets a threshold of 5 Å was used to distinguish between close and false docking solutions. **b** Spatial distribution of the close and false solutions of the soybean trypsin inhibitor docked to the porcine pancreatic trypsin subunit (PPT/STI complex, PDB: 1avw) in the old (*left*) and new (*right*) dataset. The native structure is shown in ribbons with the interacting residues Tyr151 of PPT and Arg65 of STI in sticks, and their Cα atoms in balls. The additional balls indicate the position of the Cα atom of Arg65 for the native (*deep blue*), close (*green*) and false (*grey*) STI docking solutions

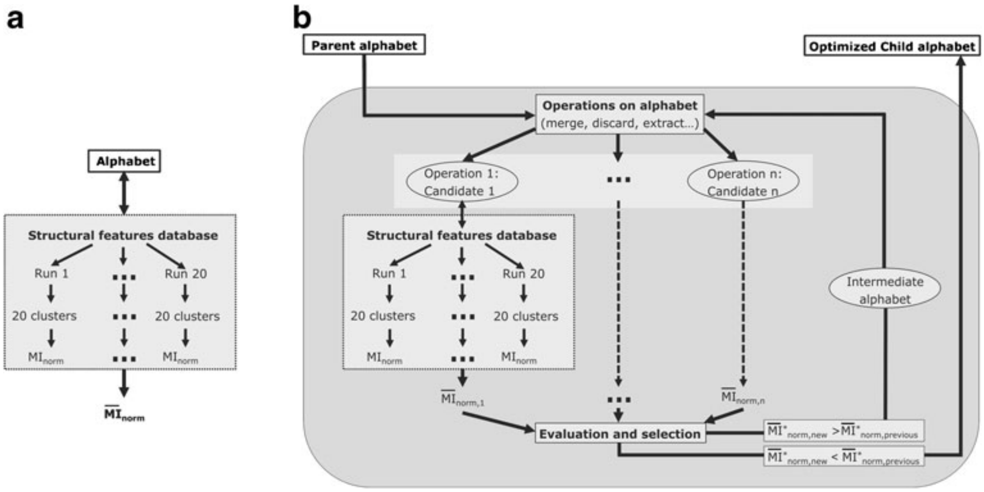


Fig. 2a,b Procedure for the calculation of $\overline{\mathrm{MI}}_{\mathrm{norm}}$ and the optimization of amino acid alphabets. **a** Schematic presentation of the $\overline{\mathrm{MI}}_{\mathrm{norm}}$ calculation strategy. (**b**) Depiction of the iterative alphabet optimization procedure. Additional steps of the iterative procedure include the generation of a large number of novel candidate alphabets by different operations (merging, discarding, extracting,…). The intermediate alphabet with the highest information content was subjected to further rounds of optimization, until no further optimization was possible and the final intermediate was termed child alphabet (see text for details)

Table 1 Results of the iterative alphabet optimization procedure. Results are shown for three iteration runs starting either from the parent alphabets P_O or P_L. For each step of iteration the respective intermediate alphabet 'I' as well as the final child alphabet 'C' is given. The type of action performed on the alphabet is listed and the affected character or class is highlighted in *bold*. For reasons of clarity, only those operations that led to the maximal increase of the $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ in the respective iteration step are shown

	Alphabet	Operation	$\overline{\mathrm{MI}}_{\mathrm{norm}}$	$\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$
P_O	ACFGILMPV~DE~HKR~NQSTWY	-	5.11 (±0.25)	4.86
$I1_O$	ACFGILMPV~DE~HKR~NQSTW~**Y**	isolate Y	7.04 (±0.28)	6.76
$I2_O$	ACFGILPV~DE~HKR~**M**Y~NQSTW	move M	8.19 (±0.21)	7.98
$I3_O$	ACFGILPV~DE~HKR~M**W**Y~NQST	move W	8.71 (±0.28)	8.43
$I4_O$	ACFGILV~DE~HK**P**R~MWY~NQST	move P	9.34 (±0.29)	9.05
$I5_O$	CFGILV~DE~HKPR~MWY~**A**NQST	move A	9.87 (±0.39)	9.48
$I6_O$	CFGIL~DE~HKPR**V**~MWY~ANQST	move V	10.26 (±0.24)	10.02
$I7_O$	CFGIL~DE**K**~HPRV~MWY~ANQST	move K	10.55 (±0.28)	10.27
C_O	CFGIL~DEK~HP**Q**RV~MWY~ANST	move Q	10.72 (±0.20)	10.52
P_L	APST~CILMV~DENQ~FWY~G~HKR	-	7.93 (±0.22)	7.71
$I1_L$	APST~**CGILMV**~DENQ~FWY~HKR	merge cl.2+5	9.06 (±0.23)	8.83
$I2_L$	APST~CGILV~DENQ~F**M**WY~HKR	move M	9.76 (±0.24)	9.52
$I3_L$	APST~CGIL~DENQ~FMWY~HKR**V**	move V	10.48 (±0.19)	10.29
$I4_L$	APST~CG**H**IL~DENQ~FMWY~KRV	move H	10.97 (±0.27)	10.70
C_L	**AKPRSTV**~CGHIL~DENQ~FMWY	merge cl.1+5	11.02 (±0.26)	10.76
P_L	APST~CILMV~DENQ~FWY~G~HKR	-	7.93 (±0.22)	7.71
$I1_L$*	APST~CILMV~DENQ~FWY~HKR	discard G	8.76 (±0.35)	8.41
$I2_L$*	APST~CILMV~DENQ~FWY~KR	discard H	9.28 (±0.32)	8.96
$I3_L$*	APST~C**G**ILMV~DENQ~FWY~KR	insert G	9.55 (±0.35)	9.20
$I4_L$*	APST~CGILM□DENQ~FWY~KR	discard V	9.69 (±0.29)	9.40
$I5_L$*	APST**V**~CGILM~DENQ~FWY~KR	insert V	10.10 (±0.38)	9.72
C_L*	APSTV~CG**H**ILM~DENQ~FWY~KR	insert H	10.18 (±0.41)	9.77

To validate the performance of the approach, results were compared to those from ZRANK [27] and dDFIRE [28], which represents the latest improved version of DFIRE [29]. ZRANK uses a combination of different energetic terms for scoring (van der Waals, electrostatic, and desolvation energies), whereas DFIRE uses an all-atom knowledge-based potential. Since ZRANK requires polar hydrogens for execution, these atoms were added to the PDB files with HBPLUS [30]. Enrichment values were calculated according to Eq. 3 above.

Web interface

To allow easy public access to the MI-based scoring function, we implemented the web server score-MI: http://score-MI.biochem.uni-erlangen.de. The server and the front end were designed using Perl, PHP, and HTML. As a minimum input for scoring, the user has to provide a file containing several docking solutions with valid chain identifiers. Scoring of an average complex (1,000 amino acids) takes approximately 1 h for 100 docking poses. In addition to the tabular presentation of the individual MI terms, a Jmol applet [31] was implemented for visual inspection of the docking solutions. A menu allows to select each solution individually for display with the interface residues of both partners highlighted in different colors. Finally, a file can be downloaded by the user that contains the docked complexes ordered by their rank.

Results and discussion

Dataset of docking solutions

Investigation of the structural features of protein interfaces requires a sufficiently large dataset to draw statistically valid conclusions. In our previous work [14], we used the precompiled Dockground 1.0 dataset, which contains 505 close and 6,100 false docking solutions for 61 different protein–protein complexes [16–18]. The new dataset, which was generated with FTDock, is more than three times larger and contains 2,634 close and 20,300 false docking solutions for a total of 203 different protein–protein complexes.

The two datasets, however, differ not only in the number but also in the distribution of the docking solutions (Fig. 1). As can be seen from Fig. 1a, the new dataset contains a much higher portion of false docking solutions with RMSDs between 5 and 10 Å, for which discrimination from the close solutions (RMSD<5 Å) should be particularly difficult. In addition, the RMSD values indicate that close and false docking solutions are now distributed more evenly in docking space. This is exemplified in Fig. 1b for the complex between the porcine pancreatic trypsin and its inhibitor (PPT/STI complex, PDB entry: 1avw). In the new dataset (right panel) the docking poses are distributed homogeneously around the native structure, whereas the old dataset (left panel) exhibits several distinct clusters that contain the majority of the docking solutions. Thus, the new dataset is not only more than three times larger, but also exhibits a more realistic distribution of close and false docking solutions, and should therefore provide a suitable basis for further method development.

Evaluation and optimization of reduced amino acid alphabets

In our previous work, we have defined a 4-class reduced alphabet to calculate the MI of different types of interface contacts [14]. Due to interactions and redundancies in the dataset, however, the MI-values of different types of contacts are not strictly additive and therefore do not allow any conclusions to be drawn about the overall MI of the alphabet. Thus, other reduced alphabets that exhibit a higher MI might exist and might therefore be more suitable for the identification of good docking solutions. Therefore, we developed a formalism that allows us to calculate the MI of an entire alphabet from a clustering analysis (Eqs. 1, 2). This clustering analysis is fast and can therefore also be applied to iterative alphabet optimization using the MI as an objective function. As starting points for this procedure, we used the 4-class alphabet ACFGILMPV-DE-HKR-NQSTWY from

Table 2 Performance of different reduced alphabets in cross-validation. The enrichment (En) of close solutions on the top ranks was calculated for the parent (P_O and P_L) and child (C_L, C_L*, and C_O) alphabets. The performance of a full 20-class alphabet and of the two common scoring tools ZRANK and dDFIRE are given for comparison

Alphabet		En (top 3)	En (top 5)	En (top 10)
P_O	ACFGILMPV-DE-HKR-NQSTWY	1.69	1.78	1.69
P_L	APST-CILMV-DENQ-FWY-G-HKR	2.53	2.34	2.27
C_L	AKPRSTV-CGHIL-DENQ-FMWY	2.69	2.68	2.51
C_L*	APSTV-CGHILM-DENQ-FWY-KR	2.86	2.57	2.48
C_O	ANST-CFGIL-DEK-HPQRV-MWY	2.64	2.55	2.56
	Each amino acid as individual class	2.68	2.82	2.74
ZRANK		2.79	2.67	2.31
dDFIRE		2.67	2.62	2.43

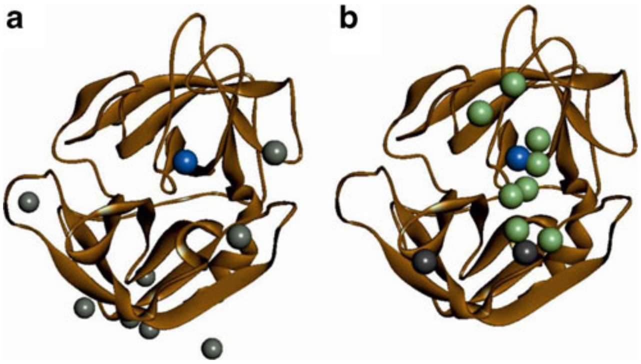


Fig. 3a,b Distribution of the ten top-scoring docking solutions. Results are compared for the application of **a** the P_O and **b** the optimized C_L* alphabets to the elastase-inhibitor complex (PDB code 1PPF). *Brown ribbon* Elastase, *balls* center of mass of inhibitor for each docking solution. *Blue* Native solution, *green* close solution, *grey* false solutions. See Table 3 for a detailed list of the underlying mutual information (MI) scores

Fig. 4a,b Screenshots of the score-MI web service. **a** Job submission interface and **b** exemplified result page. Results include a detailed tabular presentation of the individual MI terms for the top ranked solutions and a Jmol applet for visualization of the identified interfaces are provided. The user can download the ranked docking solutions and a tabular version of the corresponding MI scores ▶

Othersen et al. [14] and a 6-class alphabet APST-CILMV-DENQ-FWY-G-HKR resulting from a previous optimization procedure reported in Launay et al. [32]. These two ‘parent’ alphabets are henceforth termed P_O and P_L, respectively.

The MI of these alphabets was optimized using the strategy outlined in Fig. 2. Starting from the parent alphabets in the first round of iteration, the following operations were applied to generate a large number of novel candidate alphabets: merging two classes, discarding an amino acid or class, moving an amino acid to a different class, extracting and reintroducing an amino acid as an additional class, assigning previously discarded amino acids to an existing class, or introducing them in the alphabet as a new class. These operations resulted in approximately 120 novel candidate alphabets per iteration for which the $\overline{\mathrm{MI}}_{\mathrm{norm}}$ was subsequently calculated using the clustering procedure described in methods.

A comparison of the candidate alphabets from each round of iteration was based on the $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ value, which was calculated from $\overline{\mathrm{MI}}_{\mathrm{norm}}$ by subtracting the standard deviation to obtain a lower bound for $\overline{\mathrm{MI}}_{\mathrm{norm}}$. The intermediate alphabet with highest $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ value was subjected to further rounds of optimization if the respective $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ value was higher than the $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ of its precursor alphabet. The procedure was repeated until no further increase of the $\overline{\mathrm{MI}}^{*}_{\mathrm{norm}}$ was achieved and the final intermediate was termed child alphabet.

The results of this optimization procedure are shown in Table 1. The first block shows the iterative optimization for the 4-class P_O parent alphabet. For this alphabet, the first step of the iteration procedure ($I1_O$) was the isolation of tyrosine as a fifth class, which increased the $\overline{\mathrm{MI}}_{\mathrm{norm}}$ from 5.11 to 7.04. In the subsequent iteration steps the amino acids M, W, P, A, V, K, and Q were moved to different classes resulting in a

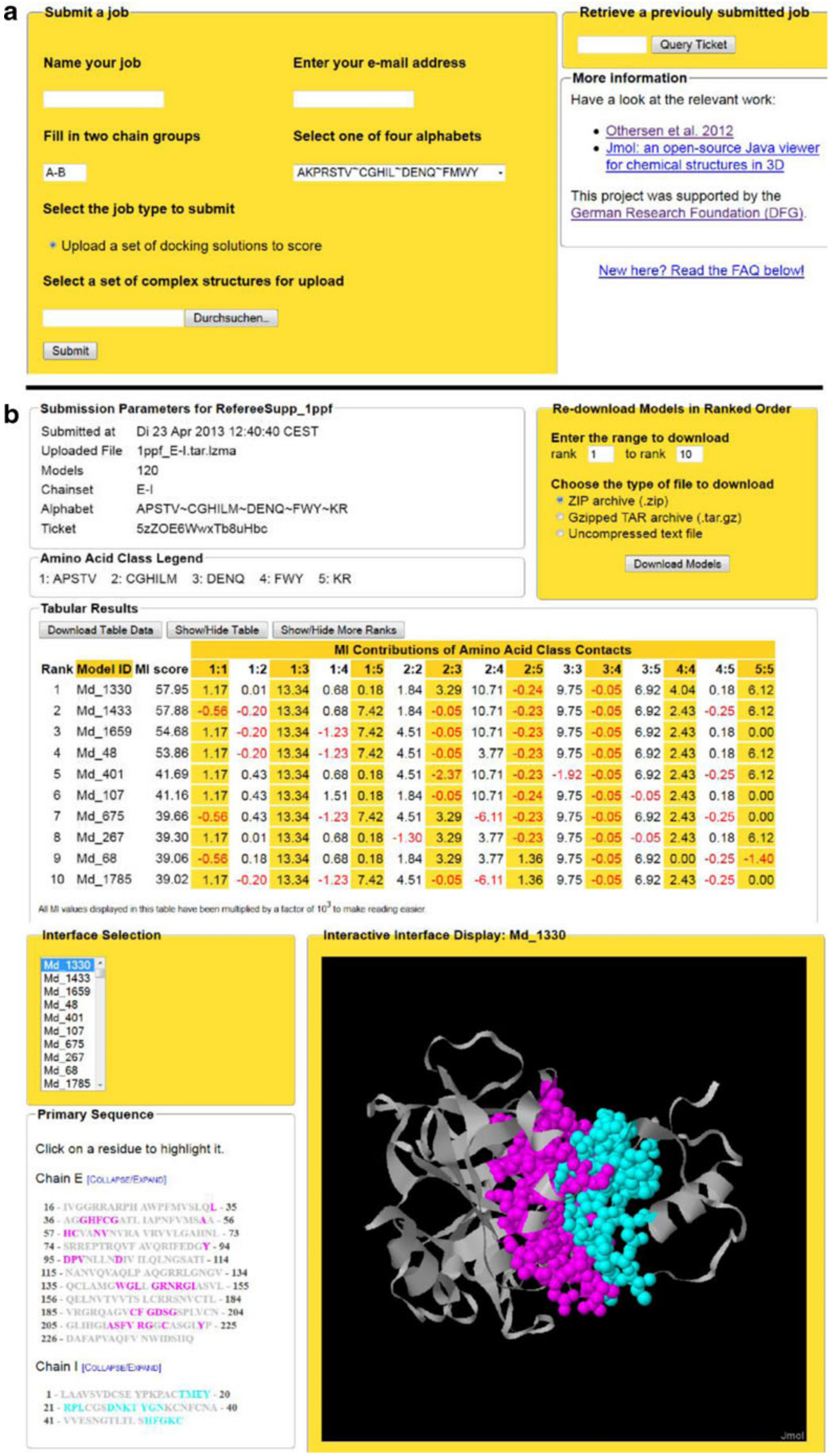
Table 3 Scoring of docking solutions with different alphabets. Results are compared for the application of the 4-class alphabet P_O (*top half*) and the five-class alphabet C_L* (lower half) to the elastase-inhibitor complex (PDB code 1PPF). In the table, classes are numbered consecutively, and pairs of numbers indicate contacts between the respective amino acid classes, e.g., “1:2” for P_O indicates contacts between HKR and DE. Ranks marked by a *prime* indicate close docking solutions. The final column lists the ligand RMSD compared to the correct docking solution

P_O Amino Acid Class Legend
1: HKR 2: DE 3: NQSTWY 4: ACFGILMPV

			MI Contributions of Amino Acid Class Contacts										
Rank	**Model ID**	**MI score**	**1:1**	**1:2**	**1:3**	**1:4**	**2:2**	**2:3**	**2:4**	**3:3**	**3:4**	**4:4**	**RMSD**
1	42	30.867	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	0.03	1.137	10.68
2	36	30.867	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	0.03	1.137	28.04
3	86	30.867	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	0.03	1.137	33.82
4	66	30.842	6.178	1.489	1.706	2.984	5.583	4.896	6.85	-0.011	0.03	1.137	10.74
5	43	30.842	6.178	1.489	1.706	2.984	5.583	4.896	6.85	-0.011	0.03	1.137	48.17
6	45	30.829	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	-0.008	1.137	19.25
7	62	30.064	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	0.03	0.334	22.58
8	26	30.064	6.178	1.489	1.706	2.984	5.583	4.896	6.85	0.014	0.03	0.334	18.13
9	28	30.039	6.178	1.489	1.706	2.984	5.583	4.896	6.85	-0.011	0.03	0.334	18.73
10	39	30.039	6.178	1.489	1.706	2.984	5.583	4.896	6.85	-0.011	0.03	0.334	10.37

C_L* Amino Acid Class Legend
1: APSTV 2: CGHILM 3: DENQ 4: FWY 5: KR

			MI Contributions of Amino Acid Class Contacts															
Rank	**Model ID**	**MI score**	**1:1**	**1:2**	**1:3**	**1:4**	**1:5**	**2:2**	**2:3**	**2:4**	**2:5**	**3:3**	**3:4**	**3:5**	**4:4**	**4:5**	**5:5**	**RMSD**
1’	1330	58.653	1.186	0.019	13.33	1.025	0.378	2.271	2.829	12.018	-0.405	9.038	-0.088	5.609	3.656	0.299	7.488	3.55
2’	1039	55.518	1.186	-0.288	13.33	1.025	0.378	2.271	-0.091	12.018	-0.405	9.038	0.004	5.609	3.656	0.299	7.488	1.40
3’	1433	51.533	-0.554	-0.288	13.33	-1.49	7.944	-1.275	-0.091	12.018	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	3.83
4’	48	51.260	1.186	-0.288	13.33	-1.49	7.944	2.271	2.829	3.539	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	3.49
5’	149	50.536	1.186	0.019	13.33	1.025	7.944	-1.275	2.829	3.539	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	4.02
6’	1438	48.934	1.186	0.019	13.33	-1.49	0.378	-1.275	2.829	12.018	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	3.16
7’	1659	48.647	1.186	0.019	13.33	-1.49	7.944	2.271	-0.091	3.539	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	2.98
8	78	47.616	1.186	0.019	13.33	1.025	7.944	-1.275	-0.091	3.539	-0.405	9.038	-0.088	5.609	0.811	-0.514	7.488	8.79
9	35	46.857	1.186	0.019	13.33	1.349	0.378	2.271	-0.091	3.539	-0.405	9.038	0.004	5.609	3.656	-0.514	7.488	5.33
10’	107	46.340	1.186	-0.288	13.33	1.025	0.378	2.271	-0.091	12.018	-0.405	9.038	0.004	-0.724	0.811	0.299	7.488	3.39



Submit a job

Name your job

Enter your e-mail address

Fill in two chain groups: A-B

Select one of four alphabets: AKPRSTV~CGHIL~DENQ~FMWY

Select the job type to submit

● Upload a set of docking solutions to score

Select a set of complex structures for upload

Durchsuchen...

Submit

Retrieve a previouly submitted job

Query Ticket

More information

Have a look at the relevant work:

- Othersen et al. 2012
- Jmol: an open-source Java viewer for chemical structures in 3D

This project was supported by the German Research Foundation (DFG).

New here? Read the FAQ below!

Submission Parameters for RefereeSupp_1ppf

Submitted at	Di 23 Apr 2013 12:40:40 CEST
Uploaded File	1ppf_E-I.tar.lzma
Models	120
Chainset	E-I
Alphabet	APSTV~CGHILM~DENQ~FWY~KR
Ticket	5zZOE6WwxTb8uHbc

Re-download Models in Ranked Order

Enter the range to download
rank 1 to rank 10

Choose the type of file to download
● ZIP archive (.zip)
○ Gzipped TAR archive (.tar.gz)
○ Uncompressed text file

Download Models

Amino Acid Class Legend

1: APSTV 2: CGHILM 3: DENQ 4: FWY 5: KR

Tabular Results

Download Table Data | Show/Hide Table | Show/Hide More Ranks

			MI Contributions of Amino Acid Class Contacts														
Rank	Model ID	MI score	1:1	1:2	1:3	1:4	1:5	2:2	2:3	2:4	2:5	3:3	3:4	3:5	4:4	4:5	5:5
1	Md_1330	57.95	1.17	0.01	13.34	0.68	0.18	1.84	3.29	10.71	-0.24	9.75	-0.05	6.92	4.04	0.18	6.12
2	Md_1433	57.88	-0.56	-0.20	13.34	0.68	7.42	1.84	-0.05	10.71	-0.23	9.75	-0.05	6.92	2.43	-0.25	6.12
3	Md_1659	54.68	1.17	-0.20	13.34	-1.23	7.42	4.51	-0.05	10.71	-0.23	9.75	-0.05	6.92	2.43	0.18	0.00
4	Md_48	53.86	1.17	-0.20	13.34	-1.23	7.42	4.51	-0.05	3.77	-0.23	9.75	-0.05	6.92	2.43	0.18	6.12
5	Md_401	41.69	1.17	0.43	13.34	0.68	0.18	4.51	-2.37	10.71	-0.23	-1.92	-0.05	6.92	2.43	-0.25	6.12
6	Md_107	41.16	1.17	0.43	13.34	1.51	0.18	1.84	-0.05	10.71	-0.24	9.75	-0.05	-0.05	2.43	0.18	0.00
7	Md_675	39.66	-0.56	0.43	13.34	-1.23	7.42	4.51	3.29	-6.11	-0.23	9.75	-0.05	6.92	2.43	-0.25	0.00
8	Md_267	39.30	1.17	0.01	13.34	0.68	0.18	-1.30	3.29	3.77	-0.23	9.75	-0.05	-0.05	2.43	0.18	6.12
9	Md_68	39.06	-0.56	0.18	13.34	0.68	0.18	1.84	3.29	3.77	1.36	9.75	-0.05	6.92	0.00	-0.25	-1.40
10	Md_1785	39.02	1.17	-0.20	13.34	-1.23	7.42	4.51	-0.05	-6.11	1.36	9.75	-0.05	6.92	2.43	-0.25	0.00

All MI values displayed in this table have been multiplied by a factor of 10^3 to make reading easier

Interface Selection

Md_1330
Md_1433
Md_1659
Md_48
Md_401
Md_107
Md_675
Md_267
Md_68
Md_1785

Interactive Interface Display: Md_1330

Primary Sequence

Click on a residue to highlight it.

Chain E [Collapse/Expand]

16 - IVGGRRARPH AWPFMVSLQL - 35
36 - AGGHFCGATL IAPNFVMSAA - 56
57 - HCVANVNVRA VRVVLGAHNL - 73
74 - SRREPTRQVF AVQRIFEDGY - 94
95 - DPVNLLNDIV ILQLNGSATI - 114
115 - NANVQVAQLP AQGRRLGNGV - 134
135 - QCLAMGWGLL GRNRGIASVL - 155
156 - QELNVTVVTS LCRRSNVCTL - 184
185 - VRGRQAGVCF GDSGSPLVCN - 204
205 - GLIHGIASFV RGGCASGLYP - 225
226 - DAFAPVAQFV NWIDSIIQ

Chain I [Collapse/Expand]

1 - LAAVSVDCSE YPKPACTMEY - 20
21 - RPLCGSDNKT YGNKCNFCNA - 40
41 - VVESNGTLTL SHFGKC

final $\overline{\mathrm{MI}}_{\mathrm{norm}}$ of 10.72. For the respective child alphabet C_O no further optimization was possible.

Starting from the 6-class, the P_L alphabet led to a reduction of the classes in the first step of the iteration. In this step, glycine, which was originally considered as a separate class, was merged into the CILMV class. In subsequent steps, the amino acids M, V, and H were moved to different classes. In the final step, two classes were merged resulting in a 4-class alphabet.

Comparison of the $\overline{\mathrm{MI}}_{\mathrm{norm}}$ of C_O and C_L shows that both alphabets exhibit a similar information content. The increase of the $\overline{\mathrm{MI}}_{\mathrm{norm}}$, however, was considerably lower for the $P_L \rightarrow C_L$ optimization procedure, which can be explained readily by the fact that P_L itself represented an optimized alphabet obtained by a different strategy [32]. One rather unexpected observation was the merging of two classes of amino acids in the final step of P_L optimization ($I4_L \rightarrow C_L$) (Table 1). To investigate whether the results can be improved if merging is made more difficult, the calculation procedure was modified in the following way: merging itself was no longer directly possible in the modified protocol. Instead, merging was possible only as a result of two iterative steps, in which an amino acid is first discarded from the alphabet and in a second step again inserted into the alphabet, but into a different class.

Application of the respective protocol to P_L resulted in the removal of G, and H in the first two steps, resulting in a 5-class alphabet comprising 18 amino acids ($I2_L$*). These amino acids are assigned to different classes in subsequent steps of the iteration procedure ($I2_L$* → $I3_L$* and $I5_L$* → C_L*, respectively). The same discarding and re-inserting is also observed for valine resulting in a total of three amino acids, which have been moved to different classes. Using this setup, optimization results in a 5-class alphabet. The $\overline{\mathrm{MI}}_{\mathrm{norm}}$ is lower compared to the 4-class C_L, which might be explained by the lower number of operation allowed for C_L* calculation.

All child alphabets exhibit a quite similar number of four to five classes, although this number was not restrained during the iteration procedure, and theoretically between 1 and 20 classes would have been allowed. The observation that the resulting child alphabets differ with respect to the classification of several amino acids most probably reflects the fact that our search procedure is not exhaustive and becomes trapped in local optima. Despite this limitation, all iteration runs have resulted in child alphabets with a significantly higher $\overline{\mathrm{MI}}_{\mathrm{norm}}$ compared to their parent alphabets, demonstrating that the clustering procedure in conjunction with the optimization protocol results in an increased information content.

Performance of the alphabets in docking predictions

The MI-values listed in Table 1 suggest that the child alphabets should perform better than their parent alphabets in discriminating between good and bad docking solutions. To address this point in a quantitative fashion, the MI values were converted into a scoring function using a previously established protocol [14]. To avoid biasing of the results, the generation of the scoring function and scoring itself were performed in a five-fold cross-validated fashion.

The performance of the scoring function was assessed by counting the close docking solutions among the three, five, or ten top-scoring solutions. To obtain a more realistic estimation of the performance, their number was corrected by the number of close solutions that are expected to be found by chance on the same top ranks (see Methods). The resulting enrichment values are given in Table 2.

Application of the P_O alphabet results in a ~1.75 fold enrichment of close docking solutions on the first ranks, while the P_L alphabet performs significantly better as evidenced by the 2.3 to 2.5 fold enrichment. This better performance of P_L is in line with its higher $\overline{\mathrm{MI}}_{\mathrm{norm}}$ compared to P_O (Table 1) and can be explained by the fact that P_L itself has resulted from a previous alphabet optimization procedure [32]. Table 2 shows that the higher $\overline{\mathrm{MI}}_{\mathrm{norm}}$ of the child compared to their parent alphabets is also reflected in the performance in scoring: all three child alphabets exhibit an enrichment of 2.5- to 2.8-fold and thus perform even better than P_L. There is no significant difference between the performance of the child alphabets: C_L* performs better for the top three ranks, whereas C_O gives slightly better results for the top ten.

Most interestingly, the performance of the child alphabets is almost equivalent to a scoring function in which each amino acid is treated as separate class (Table 2). For the top three positions, one of the reduced alphabets (C_L*) performs even better than the full alphabet. This finding suggests that optimized reduced alphabets are capable of capturing the properties of protein–protein interfaces, with a similar accuracy than the more complex 20-class alphabet. This finding is in line with previous studies demonstrating that optimized reduced alphabets perform similar as full alphabets [21, 32, 33] or can even outperform full alphabets for particular biological problems like protein fold assignment [34].

The improved performance of the optimized alphabets becomes particularly evident for those docking cases for which the P_O alphabet failed to place any close docking solutions on the top ranks. One example for this situation is the complex between the human leukocyte elastase (PMN elastase) protein and the third domain of the turkey ovomucoid inhibitor (Fig. 3a). In contrast, the optimized C_L* alphabet places eight close docking solutions among the top 10 solutions (Fig. 3b). The contribution of the different types of contacts to the overall score can be seen in Table 3. All ten top-scoring solutions selected based on the P_O alphabet exhibit RMSD values of >10 Å compared to the correct solution. In contrast, all ten top-scoring solutions from C_L* exhibit RMSD values of <10 Å, and the three top-scoring solutions even exhibit RMSDs of <4 Å compared to the correct solution. The MI-based scoring approach is also accessible via the web interface score-MI: http://score-MI.biochem.uni-erlangen.de and the key features of the service are shown in Fig. 4.

The performance of the present approach was also compared to that of the two popular scoring functions ZRANK [27] and dDFIRE [28, 29]. According to the results shown in Table 2, these two scoring functions perform quite similarly on the present dataset. The child alphabets from the present work perform similarly to the two established scoring functions for the top three ranks, and even slightly better for the top ten ranks. The overall similar performance is interesting in the light of the fact that the three methods treat the features of amino acids in different ways: either as physical energy terms (ZRANK), as atom-based potential (dDFIRE), or as residue-based potential (present work). Thus, one might speculate that the measured performance is close to the upper limit that can be achieved by an isolated consideration of amino acid properties and that consideration of additional and more sophisticated structural features will be required to further enhance the performance of scoring functions in future.

Conclusions

We have developed a strategy to optimize the amino acid grouping in reduced amino acid alphabets thereby enhancing their ability to identify good docking solutions. The method relies on a clustering approach using MI as the objective function. Application of the clustering approach does not require generation of a scoring function for each designed alphabet during the iteration process, because the work above demonstrates that there is at least a qualitative correlation between the MI of an alphabet and its performance in scoring. This allows one to use the fast clustering procedure during alphabet optimization and conversion into a scoring function is required only for the resulting child alphabets. Application of this approach allowed us to iteratively increase the MI of reduced alphabets and the performance of the optimized alphabets was demonstrated in a cross-validated scoring approach to be similar to that of the ZRANK and DFIRE scoring schemes.

Acknowledgments The project was funded by Deutsche Forschungsgemeinschaft within the priority program 1395 (SPP 1395) by grants to J.H. and H.S.

References

1. Jones S, Thornton JM (1995) Protein-protein interactions: a review of protein dimer structures. Prog Biophys Mol Biol 63:31–65
2. Lo Conte L, Chothia C, Janin J (1999) The atomic structure of protein–protein recognition sites. J Mol Biol 285:2177–2198
3. Jones S, Thornton JM (1996) Principles of protein–protein interactions. Proc Natl Acad Sci USA 93:13–20
4. Nooren IM, Thornton JM (2003) Diversity of protein–protein interactions. EMBO J 22:3486–3492
5. Pawson T, Nash P (2003) Assembly of cell regulatory systems through protein interaction domains. Science 300:445–452
6. Aloy P, Russell RB (2004) Ten thousand interactions for the molecular biologist. Nat Biotechnol 22:1317–1321
7. Young KH (1998) Yeast two-hybrid: so many interactions, (in) so little time. Biol Reprod 58:302–311
8. Aebersold R, Mann M (2003) Mass spectrometry-based proteomics. Nature 422:198–207
9. Yu H, Braun P, Yildirim MA, Lemmens I, Venkatesan K, Sahalie J, Hirozane-Kishikawa T, Gebreab F, Li N, Simonis N, Hao T, Rual JF, Dricot A, Vazquez A, Murray RR, Simon C, Tardivo L, Tam S, Svrzikapa N, Fan C, de Smet AS, Motyl A, Hudson ME, Park J, Xin X, Cusick ME, Moore T, Boone C, Snyder M, Roth FP, Barabasi AL, Tavernier J, Hill DE, Vidal M (2008) High-quality binary protein interaction map of the yeast interactome network. Science 322:104–110
10. Gavin AC, Superti-Furga G (2003) Protein complexes and proteome organization from yeast to man. Curr Opin Chem Biol 7:21–27
11. Gietz RD, Triggs-Raine B, Robbins A, Graham KC, Woods RA (1997) Identification of proteins that interact with a protein of interest: applications of the yeast two-hybrid system. Mol Cell Biochem 172:67–79
12. Halperin I, Ma B, Wolfson H, Nussinov R (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. Proteins 47:409–443
13. Smith GR, Sternberg MJ (2002) Prediction of protein–protein interactions by docking methods. Curr Opin Struct Biol 12:28–35
14. Othersen OG, Stefani AG, Huber JB, Sticht H (2012) Application of information theory to feature selection in protein docking. J Mol Model 18:1285–1297
15. Gabb HA, Jackson RM, Sternberg MJ (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. J Mol Biol 272:106–120
16. Douguet D, Chen HC, Tovchigrechko A, Vakser IA (2006) Dockground resource for studying protein–protein interfaces. Bioinformatics 22:2612–2618
17. Gao Y, Douguet D, Tovchigrechko A, Vakser IA (2007) Dockground system of databases for protein recognition studies: unbound structures for docking. Proteins 69:845–851
18. Liu S, Gao Y, Vakser IA (2008) Dockground protein-protein docking decoy set. Bioinformatics 24:2634–2635
19. Levitt M, Warshel A (1975) Computer-simulation of protein folding. Nature 253:694–698
20. Cover TM, Thomas JA (2006) Elements of information theory. Wiley-Interscience, Hoboken
21. Bacardit J, Stout M, Hirst JD, Valencia A, Smith RE, Krasnogor N (2009) Automated alphabet reduction for protein datasets. BMC Bioinforma 10:6
22. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice hall advanced reference series. Prentice-Hall
23. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31:264–323
24. Achtert E, Goldhofer S, Kriegel H-P, Schubert E, Zimek A (2012) Evaluation of clusterings—metrics and visual support. 28th International Conference on Data Engineering (ICDE), Washington, pp 1285–1288
25. Lloyd SP (1982) Least-squares quantization in pcm. IEEE Trans Inf Theory 28:129–137

26. Arthur D, Vassilvitskii S (2007) K-means++: The advantages of careful seeding. Paper presented at the proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. New Orleans, Louisiana
27. Pierce B, Weng Z (2007) Zrank: Reranking protein docking predictions with an optimized energy function. Proteins 67:1078–1086
28. Yang Y, Zhou Y (2008) Specific interactions for ab initio folding of protein terminal regions with secondary structures. Proteins 72:793–803
29. Yang Y, Zhou Y (2008) Ab initio folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions. Protein Sci 17:1212–1219
30. Wallace AC, Laskowski RA, Thornton JM (1995) Ligplot: A program to generate schematic diagrams of protein-ligand interactions. Protein Eng 8:127–134
31. Jmol: An open-source java viewer for chemical structures in 3d. http://www.Jmol.Org/
32. Launay G, Mendez R, Wodak S, Simonson T (2007) Recognizing protein-protein interfaces with empirical potentials and reduced amino acid alphabets. BMC Bioinforma 8:270
33. Melo F, Marti-Renom MA (2006) Accuracy of sequence alignment and fold assessment using reduced amino acid alphabets. Proteins 63:986–995
34. Peterson EL, Kondev J, Theriot JA, Phillips R (2009) Reduced amino acid alphabets exhibit an improved sensitivity and selectivity in fold assignment. Bioinformatics 25:1356–1362