

Structural bioinformatics

iScore: a novel graph kernel-based function for scoring protein–protein docking models

Cunliang Geng¹, Yong Jung^{2,3,4}, Nicolas Renaud⁵,
Vasant Honavar^{2,3,4,6,7,8,9}, Alexandre M. J. J. Bonvin^{1,*}  and
Li C. Xue^{1,*} 

¹Bijvoet Center for Biomolecular Research, Faculty of Science – Chemistry, Utrecht University, Utrecht 3584 CH, The Netherlands, ²Bioinformatics & Genomics Graduate Program, Pennsylvania State University, University Park, PA 16802, USA, ³Artificial Intelligence Research Laboratory, Pennsylvania State University, University Park, PA 16823, USA, ⁴Huck Institutes of the Life Sciences, Pennsylvania State University, University Park, PA 16802, USA, ⁵Netherlands eScience Center, Amsterdam 1098 XG, The Netherlands, ⁶Center for Big Data Analytics and Discovery Informatics, Pennsylvania State University, University Park, PA 16823, USA, ⁷Institute for Cyberscience, ⁸Clinical and Translational Sciences Institute and ⁹College of Information Sciences & Technology, Pennsylvania State University, University Park, PA 16802, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on December 18, 2018; revised on May 8, 2019; editorial decision on June 9, 2019; accepted on June 11, 2019

Abstract

Motivation: Protein complexes play critical roles in many aspects of biological functions. Three-dimensional (3D) structures of protein complexes are critical for gaining insights into structural bases of interactions and their roles in the biomolecular pathways that orchestrate key cellular processes. Because of the expense and effort associated with experimental determinations of 3D protein complex structures, computational docking has evolved as a valuable tool to predict 3D structures of biomolecular complexes. Despite recent progress, reliably distinguishing near-native docking conformations from a large number of candidate conformations, the so-called scoring problem, remains a major challenge.

Results: Here we present iScore, a novel approach to scoring docked conformations that combines HADDOCK energy terms with a score obtained using a graph representation of the protein–protein interfaces and a measure of evolutionary conservation. It achieves a scoring performance competitive with, or superior to, that of state-of-the-art scoring functions on two independent datasets: (i) Docking software-specific models and (ii) the CAPRI score set generated by a wide variety of docking approaches (i.e. docking software-non-specific). iScore ranks among the top scoring approaches on the CAPRI score set (13 targets) when compared with the 37 scoring groups in CAPRI. The results demonstrate the utility of combining evolutionary, topological and energetic information for scoring docked conformations. This work represents the first successful demonstration of graph kernels to protein interfaces for effective discrimination of near-native and non-native conformations of protein complexes.

Availability and implementation: The iScore code is freely available from Github: <https://github.com/DeepRank/iScore> (DOI: 10.5281/zenodo.2630567). And the docking models used are available from SBGrid: <https://data.sbgrid.org/dataset/684>.

Contact: a.m.j.j.bonvin@uu.nl or me.lixue@gmail.com

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Protein–protein interactions (PPIs) play a crucial role in most cellular processes and activities such as signal transduction, immune response, enzyme catalysis, etc. Getting insight into the three dimensional (3D) structures of those protein–protein complexes is fundamental to understand their functions and mechanisms (Aloy and Russell, 2006; Kiel et al., 2008). Due to the prohibitive cost and effort involved in experimental determination of the structure of protein complexes (Shoemaker and Panchenko, 2007), computational modelling, and in particular docking, has established itself as a valuable complementary approach to obtaining insights into structural basis of protein interactions, interfaces and complexes (Halperin et al., 2002; Huang, 2014; Melquiond et al., 2012; Rodrigues and Bonvin, 2014; Soni and Madhusudhan, 2017; Stein et al., 2011; Vangone et al., 2017).

Computational docking typically involves two steps (Halperin et al., 2002; Huang, 2014; Rodrigues and Bonvin, 2014; Soni and Madhusudhan, 2017): Sampling, i.e. the search of the interaction space between two molecules to generate as many as possible near-native models; and scoring, i.e. the identification of near-native models out of the pool of sampled conformations. As shown in the community-wide Critical Assessment of PRediction of Interactions (CAPRI) (Lensink and Wodak, 2010; 2013; Lensink et al., 2007, 2017), scoring is still a major challenge in the field. There is thus still plenty of room to improve the scoring functions used in protein–protein docking (Moal et al., 2013; Vangone et al., 2017).

Scoring functions can be classified into three types: (i) physical energy term-based, (ii) statistical potential-based and (iii) machine learning-based. Physical energy-based scoring functions are usually a weighted linear combination of multiple energetic terms. These are widely used in many docking programs such as HADDOCK (Dominguez et al., 2003; Vangone et al., 2016), SwarmDock (Torchala et al., 2013), pyDock (Cheng et al., 2007; Grosdidier et al., 2007; Jiménez-García et al., 2013), ZDock (Pierce et al., 2014; Pierce and Weng, 2007) and ATTRACT (Zacharias, 2003). Taking HADDOCK as an example, its scoring function consists of intermolecular electrostatic and van der Waals energy terms combined with an empirical desolvation potential (Fernández-Recio et al., 2004) as well as a buried surface area (BSA)-based term depending on the stage of the protocol (Vangone et al., 2016). Statistical potential-based scoring functions such as 3D-Dock (Moont et al., 1999), DFIRE (Zhou and Zhou, 2002) and SIPPER (Pons et al., 2011b), typically convert distance-dependent pairwise atom–atom or residue–residue contacts distributions into potentials through Boltzmann inversion. Unlike classical scoring functions that consist of linear combinations of energy terms, or simple geometric and physicochemical features (Bourquard et al., 2011; Fink et al., 2011; Moal et al., 2017), a machine learning approach can discover complex nonlinear combinations of features of protein–protein interfaces to train a classifier to label a docking model as near-native model or not. Simple machine learning algorithms work with fixed dimensional feature vectors. Because interfaces of different docking models can vary widely in size and shape, and in the arrangement of their interfacial residues, most machine learning based scoring functions typically use global features of the entire interface, for example, the total interaction energy and the BSA. However, such an

approach fails to effectively utilize details of the spatial arrangement of interfacial residues/atoms.

Graphs, in which the nodes encode the amino acid residues or atoms and the intermolecular contacts between them are encoded by the edges, offer a natural and information-rich representation of protein–protein interfaces. Unlike the global interface feature vectors described above, a graph has a residue- or atom-level resolution and naturally encodes the topological information of interacting residues/atoms (Bunke and Riesen, 2011; Vento, 2015). Furthermore, the size of a graph is not fixed and can vary depending on the size of the interface.

Such graph-based descriptions have been used previously in several scoring functions (Chang et al., 2008; Khashan et al., 2012; Pons et al., 2011a). Graph (or network) topology-based metrics have mostly been used. Chang et al. (2008) exploited node degrees (measuring the number of direct contacts of a node) and clustering coefficients (measuring how likely a node and its neighbours tend to form a clique) to score docking models. Similarly, Pons et al. (2011a) used closeness (measuring how far a node from the rest of the nodes in a network) and betweenness (measuring how important a node as a connector in a network) in scoring with the intuition that residues with high centralities in a network tend to be key functional residues. Unlike the network topology-based approaches, the SPIDER (Khashan et al., 2012) scoring function uses a graph to represent the interface at residue level with nodes labeled by their amino acid identity. It ranks the docking models by counting the frequency of native motifs in the interface graph. However, all the preceding fail to fully exploit the rich features of protein interfaces.

Against this background, we represent the interface with a labeled graph, where the nodes encode the interface residues, edges encode residue–residue contacts, and the nodes are annotated with evolutionary conservation profiles. We treat the scoring problem as a binary classification problem. By calculating the similarity between an interface graph from a docking model with the positive (native) and negative (non-native) interface graphs in the training set, we predict the likelihood of the query interface graph belonging to the positive class or the negative class (Fig. 1). We make use of a novel *graph kernel* to compute the pair-wise similarity between the graph representations of protein–protein interfaces. We call the resulting graph kernel-based scoring function GraphRank.

GraphRank exploits random walk graph kernel (RWGK) (Vishwanathan et al., 2010) for computing the similarity of labeled graphs, which has previously been used for protein function prediction (Borgwardt et al., 2005) to calculate the similarity between two interface graphs. By simultaneously conducting random walks on two graphs, RWGK measures the similarity of two graphs by aggregating the similarity of the set of random walks on the two graphs. Unlike previous graph-based scoring functions, RWGK allows GraphRank to fully exploit various node labels and edge labels and to explicitly specify the starting and ending probability of the random walks. GraphRank has two major advantages over classical machine learning based scoring functions. First, GraphRank uses a more detailed representation of protein interfaces than that provided by the fixed dimensional feature vectors used by classical machine learning approaches. GraphRank exploits residue level attributes and network topology. Second, GraphRank uses the full profile of

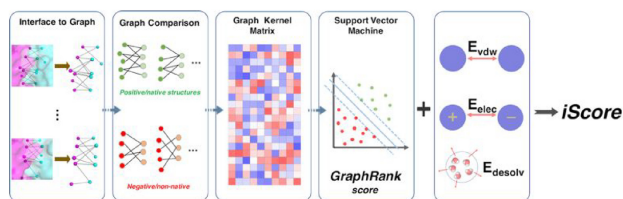


Fig. 1. Schematic workflow of our graph kernel-based scoring method. Docking models for a protein–protein complex are first represented as graphs by treating the interface residues as graph nodes and the intermolecular contacts they form as graph edges. Interface features are added to the graph as node or edge labels (only PSSM profiles as node labels in this case). Then, each of the interface graphs of the docking models is compared to the interface graphs of both the positive (native) structure and negative (non-native) models. This graph comparison generates a similarity matrix for the docking models with the number of rows and columns corresponding to the number of docking models and the total number of positive and negative graphs, respectively. Next, the support vector machine takes the graph kernel matrix as input and predicts decision values that are used as the GraphRank score. The final scoring function iScore is a linear combination of the GraphRank score and HADDOCK energetic terms (van der Waals, electrostatic and desolvation energies). The weights of this linear combination are optimized using the genetic algorithm (GA) over the BM4 HADDOCK dataset

interface conservation as node labels, i.e. each node is represented as a 20 by 1 vector of conservation profile extracted from the Position Specific Scoring Matrix (PSSM). Residue conservation information plays an important role in protein–protein recognitions (Andreani and Guerois, 2014; de Oliveira and Deane, 2017; Hopf et al., 2014) and hence different types of conservation information have been used in several existing scoring functions (Andreani et al., 2013; Tress et al., 2005; Xue et al., 2014). The PSSM is a multiple-sequence-alignment (MSA) based conservation matrix. Its value is a log likelihood ratio between the observed probability of one type of amino acid appearing in a specific position in the MSA and the expected probability of that amino acid type appearing in a random sequence. Each position in a protein can be represented as a 20 by 1 PSSM profile, which captures the conservation characteristic of each amino acid type at a specific position.

For GraphRank we designed a specific random walk graph kernel to compare interface graphs. A graph similarity matrix was calculated from a balanced dataset of native and non-native structures from the protein–protein docking benchmark version 4.0 (BM4) (Hwang et al., 2010), and was used to train a support vector machine (SVM) classifier. GraphRank, the resulting scoring function, uses only the residue conservation information as node labels and as the basis of starting and ending probabilities of random walks. We further combined the GraphRank score with intermolecular energies, resulting our final scoring function, iScore. We benchmarked the iScore and GraphRank scoring functions on two independent sets of docking models for two different purposes: (i) 4 sets of *docking software-specific* models and their respective scoring functions and (ii) the CAPRI score set, a set of *docking software-nonspecific* models, in which models from different docking programs are mixed together. We also compare our performance with that of IRAPPA (Moal et al., 2017), one of the latest state-of-the-art machine learning based scoring functions. The results of our experiments on both benchmark sets show that iScore achieves scoring performance that is competitive with or superior to that of the state-of-the-art scoring functions. These results represent the first successful demonstration of the use of graph kernel applied to protein interfaces for effective discrimination of near-native and non-native conformations of protein complexes.

2 Materials and methods

2.1 Constructing interface graph and random walk graph kernel

2.1.1 Representing protein–protein interfaces as labeled bipartite graphs

A residue is defined as an interface residue if any of its atoms is within 6 Å of any atom of another residue in the partner protein. This is a commonly used interface definition (Xue et al., 2015), and, for example, a similar cutoff (5.5 Å) has been shown to work well for contacts-based binding affinity prediction (Vangone and Bonvin, 2015). We represent the interface of a native complex or a docking model as a bipartite graph (Fig. 1), in which each node is an interface residue, and each edge consists of two nodes that are within 6 Å distance from each other (based on any atom–atom distance within 6 Å between those residues). We further label the graph node with residue conservation profiles from Position Specific Scoring Matrix (PSSM). Each node is thus represented by a 20×1 vector of PSSM profile. Our current implementation uses a single type of nodes, namely residues, labeled with their PSSM profiles, and a single type of edges, namely, those that encode inter-residue contacts. However, our framework admits multiple types of nodes and edge labels.

The PSSM was calculated through PSI-BLAST (Altschul, 1997) of BLAST 2.7.1+. The parameters of the BLAST substitution matrix, word size, gap open cost and gap extend cost were automatically set based on the length of protein sequence using the recommended values in the BLAST user guide (<https://www.ncbi.nlm.nih.gov/books/NBK279684/>) (see Supplementary Table S1). Other parameters were: Number of iterations set to 3 and the e-value threshold to 0.0001. The BLAST database used was the nr database (the non-redundant BLAST curated protein sequence database), version of February 4, 2018.

2.1.2 Random walk graph kernel for interface graphs

We define a random walk graph kernel (RWGK) to measure the similarity of two interface graphs. Given two labeled graphs, a RWGK first applies simultaneous random walks on the two graphs with the same walk length (the number of edges) and then calculates the similarity between those two random walks. The RWGK score is then the weighted sum of the walk similarity varying the walk length from 0 to infinity (Ghosh et al., 2018).

Gärtner et al. 2003) proposed an elegant approach for calculating all random walks within two graphs using direct product graphs. A graph G consists of a set of n nodes $V = \{v_1, v_2, \dots, v_n\}$ and a set of m edge $E = \{e_1, e_2, \dots, e_m\}$ where the edge e_i is defined by two nodes. Given two graphs $G = \{V, E\}$ and $G' = \{V', E'\}$, the direct product graph $G_{\times}$ is a graph defined as follows:

$$G_{\times} = G \times G' = \{V_{\times}, E_{\times}\}, \quad (1)$$

$$V_{\times} = \left\{ (v_i, v'_j) \mid v_i \in V, v'_j \in V' \right\}, \quad (2)$$

$$E_{\times} = \left\{ \left((v_i, v'_j), (v_k, v'_l) \right) \mid (v_i, v_k) \in E, (v'_j, v'_l) \in E' \right\}, \quad (3)$$

where $V_{\times}$ is the node set and $E_{\times}$ is the edge set. In other words, $G_{\times}$ is a graph over pairs of nodes from G and G' , and two nodes in $G_{\times}$ are neighbors if and only if the corresponding nodes in G and G' are both neighbors (Vishwanathan et al., 2010).

The simultaneous random walks on graphs G and G' are equivalent to a random walk on the direct product graph $G_{\times}$. In other words, each walk on the direct product graph $G_{\times}$ corresponds to

two walks on the two individual graphs, allowing the calculation of a similarity score between them. When the walk length is 1, these similarity scores are the elements of the weight matrix $W_{\times}$ of $G_{\times}$. $W_{\times}^l$ consists of similarity scores of walk length of l . The similarity between graphs G and G' is thus the weighted sum of these walk similarities.

Formally, the random walk graph kernel is originally defined by Vishwanathan et al. (2010) as:

$$k(G, G') = \sum_{l=0}^{\infty} \mu(l) q_{\times}^T W_{\times}^l p_{\times}, \quad (4)$$

where l is the length of random walk on $G_{\times}$, $\mu(l)$ is a factor that allows one to (de-)emphasize walks with different lengths, $W_{\times}$ is the weight matrix of $G_{\times}$, and $q_{\times}$ and $p_{\times}$ are the starting and stopping probabilities of random walks on $G_{\times}$, respectively. In our study, we limit the maximum walk length to 3, and $\mu(l)$ is set to 1 for $l = 0$ to 3.

And $W_{\times}$, $q_{\times}$ and $p_{\times}$ are designed as follows.

$$W_{\times}^l((v_i, v'_i), (v_j, v'_j)) = \begin{cases} \begin{cases} k_{node}(v_i, v'_i) * k_{node}(v_j, v'_j) * k_{edge}(e_l, e'_l), & i = j, l = 0 \\ 0, & i \neq j \end{cases} \\ \begin{cases} k_{node}(v_i, v'_i) * k_{node}(v_j, v'_j) * k_{edge}(e_l, e'_l), \\ \text{if } ((v_i, v'_i), (v_j, v'_j)) \in E_{\times} \\ 0, \text{ otherwise} \end{cases} \end{cases}, \quad l = 1, \quad (5)$$

where $k_{edge}(e_l, e'_l)$ is the kernel to measure the similarity between two edges, $e_l = (v_i, v_j)$ and $e'_l = (v'_i, v'_j)$. Since we do not use specific edge labels here, $k_{edge}(e_l, e'_l)$ is simply set to 1. $k_{node}(v_i, v'_i)$ is the kernel to measure similarity between nodes defined as follows:

$$k_{node}(v_i, v'_i) = \exp\left(-\frac{\|\bar{v}_i - \bar{v}'_i\|^2}{2\sigma^2}\right), \quad (6)$$

where $\bar{v}_i$ and $\bar{v}'_i$ are node labels for nodes v_i and v'_i , respectively. As described above, we used PSSM residue conservation profiles as node label. σ was set to 10 by simply checking the distribution of some $\|\bar{v}_i - \bar{v}'_i\|$ values.

We bias the random walks to start and end with conserved residues by giving those higher starting and ending probabilities. For this, we define the starting and ending probabilities $q_{\times}((v_i, v'_i))$ and $p_{\times}((v_i, v'_i))$ from the normalized conservation score as follows:

$$q_{\times}((v_i, v'_i)) = \begin{cases} 0, & \text{if } IC_{v_i} < 0.5 \text{ and } IC_{v'_i} < 0.5 \\ \frac{IC_{v_i} * IC_{v'_i}}{\sum_{j=1}^n \sum_{k=1}^{n'} IC_{v_j} * IC_{v'_k}}, & \text{otherwise} \end{cases}, \quad (7)$$

$$p_{\times}((v_i, v'_i)) = q_{\times}((v_i, v'_i)) \quad (8)$$

where IC_{v_i} and $IC_{v'_i}$ are the PSSM information content (IC) for the nodes v_i and v'_i , respectively, and n and n' are the numbers of nodes in graph G and G' , respectively. IC is always ≥ 0 . The higher the IC, the more conserved a residue is.

2.1.3 Support vector machine (SVM) algorithm

SVM maps arbitrary data objects (vectors, sequences, graphs, etc.) into a kernel-induced feature space where it searches for a hyper-plane that maximizes (or approximately maximizes) the separation

between classes (Vapnik, 2013). We used the SVM implementation from the LIBSVM (Chang and Lin, 2011) package to train a scoring function taking the $N \times N$ graph kernel matrix from the training dataset as input (N is the number of the training graphs). Given a test data (an interface graph of a docking model in our case), we calculate the kernel vector that consists of the similarities of this query graph with all the training graphs. The trained SVM-based scoring model uses the resulting vector of similarities of the query graph with all of the training graphs as well as the labels of the training graphs to predict the likelihood of the query graph corresponds to a near-native conformation.

2.2 Evaluation metrics to compare scoring functions

Each scoring function has its own default protocol for selecting top models. To avoid subjectivity in the selection of top models in our comparisons, we used the success rate at cluster level to evaluate the scoring functions on the BM5 dataset. We defined a cluster as a hit if at least one of the top four models in that cluster is of acceptable or better quality. The success rate on top N clusters was defined as the number of cases (complexes) with at least one hit out of the N clusters divided by the total number of complexes considered.

The quality of the docking models was evaluated using standard CAPRI criteria based on the interface or ligand Root Mean Squared Deviations (i-RMSDs and l-RMSDs, respectively) and fraction of native contacts (Fnat) [for details refer to Figure 1 of Lensink et al. (2007)]. They were classified as incorrect (i-RMSD $> 4 \text{ \AA}$ or Fnat < 0.1), acceptable ($2 \text{ \AA} < \text{i-RMSD} \leq 4 \text{ \AA}$ and Fnat ≥ 0.1), medium ($1 \text{ \AA} < \text{i-RMSD} \leq 2 \text{ \AA}$ and Fnat ≥ 0.3) or high (i-RMSD $\leq 1 \text{ \AA}$ and Fnat ≥ 0.5) quality (Lensink et al., 2007).

2.3 Training on docking benchmark 4 docking models

2.3.1 Training dataset for GraphRank

The dataset for training was based on protein-protein complexes from the protein-protein docking benchmark version 4.0 (BM4), considering only dimers and excluding antibody complexes, resulting in a set of 117 non-redundant protein-protein complexes. Docking models for those complexes had been generated previously by running HADDOCK in its *ab initio* mode using center of mass restraints (Karaca et al., 2013). The crystal structures of these 117 complexes (the 'native' structures) form our positive training set. The average number of nodes and edges in the corresponding graphs for this native set are 68 ± 25 and 119 ± 55 , respectively. To create a balanced training set, we randomly selected 117 non-native (wrong) models from the pool of HADDOCK models with i-RMSD $\geq 10 \text{ \AA}$ and number of graph nodes ≥ 5 as our negative training set. The average number of nodes and edges in the non-native set are 48 ± 14 and 70 ± 23 , respectively. In total, we thus have 234 (=117*2) structures as our training set.

2.3.2 Training dataset for iScore

For the training of iScore we selected BM4 complexes for which HADDOCK, running in *ab initio* mode using center of mass restraints, generated at least one good model in the final water refinement stage. This resulted in 63 cases for which at least one docking model with acceptable or better quality was present in the final set of 400 water-refined models. This dataset is denoted in the following as the BM4 HADDOCK dataset.

2.3.3 Training the graph kernel based scoring function

(GraphRank)

We applied the commonly used SVM classifier C-SVC from LIBSVM (Chang and Lin, 2011) to train our scoring function. We precomputed the random walk graph kernel matrix (234×234) for the training data and used it as input of the SVM classifier. The SVM outputs the predicted decision values for a test case (the decision values from libsvm is defined as $d \times |\rightarrow w|$, where d is the distance from a point to the hyperplane and $\rightarrow w$ is the weight vector of SVM that defines the classification hyperplane). To be consistent with energy terms which we later incorporated into iScore (the lower the energy, the better a model), we used the negative decision value from the SVM as the final score of GraphRank. The resulting optimized SVM classifier is denoted as the 'GraphRank' scoring function.

2.3.4 Integrating GraphRank score with energetic terms (iScore)

We combined the GraphRank score with three energetic terms from HADDOCK to train a simple linear scoring function named iScore.

The HADDOCK energetic terms used are:

- Evdw, the intermolecular van der Waals energy described by a 12-6 Lennard-Jones potential;
- Eelec, the intermolecular electrostatic energy described by a Coulomb potential;
- Edesolv, an empirical desolvation energy term.

The van der Waals and electrostatic energies are calculated using a 8.5 Å distance cut-off using the OPLS united atom force field (Jorgensen and Tirado-Rives, 1988).

The GraphRank score and HADDOCK terms were normalized with the following equation:

$$\text{normalised } X = \frac{X - \text{median}(X)}{IQR(X)}, \quad (9)$$

where the X is a set of values for a specific term, $\text{median}(X)$ is the median value of this term, IQR is the interquartile range, which is the difference between the 75th and 25th percentiles.

We optimized the weights of the various iScore terms (the normalized GraphRank score and energetic features) on the BM4 HADDOCK dataset (63 cases and 400 models/case), using a genetic algorithm (GA). We used the normalized discounted cumulative gain (nDCG) (Wang et al., 2013) to evaluate the model ranking from each combination of the GraphRank score and energetic terms. This is a common measure of ranking quality for evaluating web search engine algorithms (Croft et al., 2010). Specifically, nDCG is defined as follows:

$$nDCG = \frac{DCG}{iDCG}, \quad (10)$$

$$DCG = \sum_{i=1}^n \frac{2^{w_i} - 1}{i}, \quad (11)$$

$$iDCG = \sum_{j=1}^m \frac{2^{w_j} - 1}{j}, \quad (12)$$

where DCG is the discounted cumulative gain calculated over the total number of models (here n in Eq. 11 is 400). $iDCG$ is the ideal DCG (meaning all the hits are ranked at the top 1, 2, ..., m , where

m is the total number of hits), and $nDCG$ is the normalized DCG. i is the ranking position of a model, w_i is the weight of a model ranked at position i . Here, we set $w_i = 1$ if i is a near-native model, and $w_i = 0$ otherwise. Thus, the contribution of a model to DCG becomes 0 or $\frac{1}{i}$, where i is the ranking of the model.

The fitness function for the GA optimization (maximization) was defined as squared $nDCG$ values averaged over the $N=63$ cases:

$$GA \text{ fitness} = \frac{\sum_i^N nDCG_i^2}{N}, \quad (13)$$

The parameters of the GA optimization were: Population size = 800, maximum generations = 100, crossover rate = 0.8 and stopping tolerance = 0.001. The GA converged quickly, stopping at the 51th generation. The GA optimization was repeated 30 times and the median values were used as final weights.

2.4 Validation and comparison with state-of-the-art scoring functions

2.4.1 Validation on models from different docking programs

We validated iScore's performance on docking models from four different docking programs: HADDOCK (Dominguez et al., 2003; van Zundert et al., 2016), SwarmDock (Torchala et al., 2013), pyDock (Cheng et al., 2007; Grosdidier et al., 2007; Jiménez-García et al., 2013) and ZDock (Pierce et al., 2014; Pierce and Weng, 2007). These models were used to evaluate our scoring functions and compare them with the original scoring functions in these respective docking programs. The protein-protein complexes used for testing consist of the new entries from the protein-protein docking benchmark version 5.0 (BM5) (Vreven et al., 2015), on which none of the docking software listed above has been previously trained. These new cases are not present in and hence are non-redundant with BM4, which is our training set. Antibody complexes were excluded. The HADDOCK docking models for the BM5 new cases were generated using predicted interface residues from CPORT (de Vries and Bonvin, 2011) as reported in the BM5 paper (Vreven et al., 2015). The docking models for ZDock, pyDock and SwarmDock were taken from the work of Moal et al. (2017). In total, we could use 9, 18, 14 and 10 complexes for HADDOCK, SwarmDock, pyDock and ZDock, respectively, with the number of models per case varying from 125 to 500, for which at least one near-native model was present in the set of generated models.

Calculating HADDOCK energetic terms. We used HADDOCK to calculate the intermolecular energies for the docking models from other docking programs. For this, the missing atoms of the models were built according to the OPLS force field topology with standard HADDOCK scripts using CNS (Brünger et al., 1998). A short energy minimization (EM) was then performed with the following settings: 50 steps of conjugate gradient EM, van der Waals interactions truncated below the distance of 0.5 Å, and dielectric constant set to 1.

Removing docking models containing clashes. Docking models originating from rigid-body docking programs, such as ZDock and pyDock, often contain clashes that a short EM cannot resolve. We removed those clashing models from the test dataset following the CAPRI assessment procedure: A clash is defined by a pair of heavy atoms between protein partners with a distance below 3 Å. We discarded all models with more than 0.1 clashes per Å² of buried surface.

Clustering. The remaining docking models for each case were clustered with the fraction of common contacts (FCC) method (Rodrigues et al., 2012) using a 0.6 cut-off and requiring a minimum number of 4 members per cluster.

Comparison with IRaPPA on models from different docking programs. We compared our performance with that of IRaPPA on models of the new BM5 complexes from SwarmDock, pyDock and ZDock (Moal et al., 2017). The authors of IRaPPA kindly provided us their selection of top 10 models (one model from each of the top 10 clusters). This allowed us to compare our results with IRaPPA on a per model level. iScore's default protocol of selecting top 10 models is to select top 2 models from the top 5 clusters for each target when applicable. If less than 5 clusters are present, iScore evenly selects top models from each cluster. In cases where the models are too diverse to be clustered (e.g. only 1 cluster with 4 models and the large majority of models not clustering), iScore selects all models from all available clusters, and then chooses the remaining models from not clustered models.

2.4.2 Validation on the CAPRI score set

The CAPRI score set consists of a set of models collected from CAPRI participants and used in the scoring experiment of CAPRI (Lensink and Wodak, 2014). During the CAPRI scoring competitions, each scoring group is asked to select top 10 models. We tested our scoring functions on this dataset and compared its performance with various scoring functions used in the CAPRI challenge. Docking models with clashes were removed as described above. Both dimers and multimers were considered here. We used 13 cases from the CAPRI score set with number of models ranging between 497 and 1987. Following the CAPRI assessment protocol, we considered only 10 models for assessment. iScore's default model selection protocol was used, i.e. simply selecting the top 2 models of the top 5 clusters for each target.

3 Results

3.1 Training and optimization

We first trained a graph kernel-based scoring function called GraphRank using an SVM classifier. GraphRank ranks docking models based on their similarity to the native/non-native set of structures used in the training. The similarity is measured concerning interface topology and conservation profiles. The smaller the GraphRank score is, the more similar the docking model is to native complexes.

We then trained iScore by integrating the GraphRank score with three intermolecular energy terms from HADDOCK (see Section 2). iScore consists of a linear combination of those four features whose weights were optimized on the BM4 HADDOCK docking models. To avoid extreme values of energies, we independently normalized the various terms for models from each case with their median and interquartile range values. The iScore function with its optimized weights is:

$$\begin{aligned} iScore = & 0.941 * nGraphRank_{score} + \\ & 0.041 * nE_{vdw} + \\ & 0.217 * nE_{elec} + \\ & 0.032 * nE_{desolv} \end{aligned} \quad (14)$$

where $nGraphRank_{score}$, nE_{vdw} , nE_{elec} and nE_{desolv} are the normalized GraphRank score, E_{vdw}, E_{elec} and E_{desolv} energies, respectively. The GraphRank score has the highest weight (0.941),

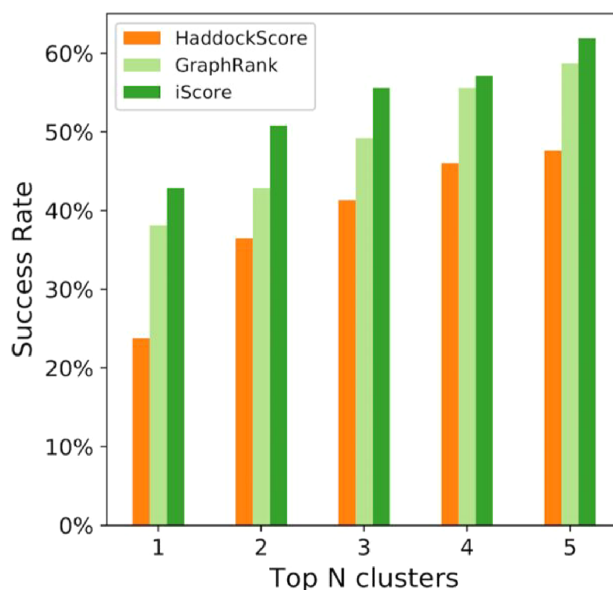


Fig. 2. Success rate of HADDOCK score, GraphRank and iScore on the BM4 HADDOCK training dataset over top N clusters of models

indicating that the GraphRank score using PSSMs alone is the most important component of iScore.

The success rates of HADDOCK score, GraphRank score and iScore on the BM4 HADDOCK dataset (63 cases) are shown in Figure 2. GraphRank scores are obtained by leaving-one-complex-out, i.e. we keep all models from one complex as the testing data and rank them after training GraphRank on the remaining complexes, and we repeat this process for all complexes. Compared with the energy-based HADDOCK score, the graph- and conservation-based GraphRank score has higher success rates. It is also evident that adding energetic features in iScore results in an improved scoring, reaching a success rate of 62% on the top 5 clusters in comparison with 59% for GraphRank.

3.2 Benchmarking on docking software specific docking models and their respective scoring functions

Sampling and scoring are typically not independent components. They are often interrelated since a specific scoring method might depend on the sampling strategy followed and the representation of the system. We benchmarked here the performance of iScore and GraphRank, which are trained on HADDOCK models, on docking software-specific docking models and compared their performance with that of each software respective scoring function.

For this, models from the new protein-protein complexes of Docking Benchmark 5 (Vreven et al., 2015) were generated using four widely used docking programs: HADDOCK (Dominguez et al., 2003; van Zundert et al., 2016), SwarmDock (Torchala et al., 2013), pyDock (Cheng et al., 2007; Grosdidier et al., 2007; Jiménez-García et al., 2013) and ZDock (Pierce et al., 2014; Pierce and Weng, 2007). The numbers of available complexes with near-native docking models for those four widely used docking programs are 9, 18, 14 and 10, respectively, with the numbers of docking models per complex varying from 125 to 500. The scoring performance was assessed with clustering of the docking models using our cluster procedure described in Section 2.

iScore outperforms HADDOCK, ZDOCK and pyDock scoring functions and competes with that of SwarmDock on their respective

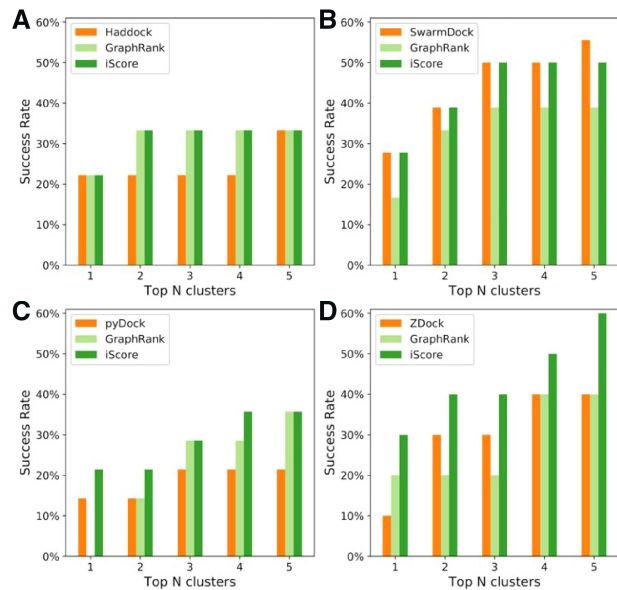


Fig. 3. Success rates measured at cluster level on four sets of docking program-specific models for newly added protein–protein complexes in BM5. GraphRank and iScore are compared with scoring functions from HADDOCK (A), SwarmDock (B), pyDock (C) and ZDock (D) on the docking models of the corresponding docking program, respectively

docking program-specific models (Fig. 3). On the HADDOCK models (Fig. 3A), iScore shows the same performance as GraphRank, both outperforming HADDOCK on the top2 to top4, reaching 33% success rate for top 5 clusters. For all the other model sets, iScore outperforms GraphRank. It shows a better scoring performance than the original scoring functions of pyDock (Fig. 3C) and ZDock (Fig. 3D), while the original SwarmDock scoring function remains the best in terms of scoring performance (Fig. 3B). iScore reaches a success rate of 36% and 60% (top 5 clusters) on pyDock and ZDock models, respectively, which is clearly a large improvement.

The scoring performance of iScore competes with that of IRaPPA (Moal *et al.*, 2017), a state-of-the-art machine learning based scoring function, on BM5 docking models generated by SwarmDock, pyDock and ZDock (Moal *et al.*, 2017) (for comparisons for each complex see Supplementary Table S2, and for overall performances see Table 1). IRaPPA identifies at least one hit for more cases than GraphRank and iScore, while iScore identifies higher-quality models for more cases than IRaPPA (Table 1). Specifically, iScore is successful in its top 10 models for 10, 6 and 6 cases for SwarmDock, pyDock and ZDock models, respectively, while IRaPPA for 12, 10 and 8, respectively. However, iScore appears to be more sensitive to high- or medium-quality models than IRaPPA: iScore and IRaPPA obtain 2 and 1 high-quality complexes on SwarmDock models, respectively and 5 and 3 medium-quality models on ZDock models, respectively. Considering that iScore was trained exclusively on BM4 HADDOCK models using a small number of features (1 for GraphRank, 4 for iScore) it performs well compared to IRaPPA, which exploits using 91 features and was separately trained on docking models generated by SwarmDock, ZDock and PyDock, respectively.

3.3 iScore ranks among the top scorers on the CAPRI score set

The scoring set from the CAPRI scoring experiments (Lensink and Wodak, 2014) is a valuable resource for evaluating scoring functions.

Table 1. Comparison of GraphRank and iScore with IRaPPA on docking program-specific models of BM5 protein–protein complexes

Docking models	#Complexes	GraphRank	iScore	IRaPPA
SwarmDock	18	7/1***/6**	10/2***/6**	12/1***/6**
pyDock	14	5/3**	6/3**	10/3**
ZDock	10	4/3**	6/5**	8/3**

Note: 10 models are selected and evaluated. The scoring performance for each complex is reported as the number of acceptable or better models (hits), followed by the number of high (indicated with ***) or medium quality models (**). The overall performance of each method on all complexes is reported here. For example, 7/1***/6** means that a scoring function is successful in 7 complexes, 1 complex out of the 7 complexes has at least a *** model and 6 out of 7 have at least a ** model in the top 10.

CAPRI is a community-wide experiment for evaluating docking programs (started in 2001) (Janin, 2002) and scoring functions (from 2005 on). The CAPRI score set consists of 15 targets, 13 of which have near-native docking models. Each target has a mixture of 500–2000 models from the various docking programs used in the CAPRI prediction challenges (Table 2). This represents an ideal set for evaluating scoring functions *independently of docking programs*.

We benchmarked iScore and GraphRank on the models from the CAPRI score set and compared their performance with the reported performance of the various scoring functions/groups which participated to the CAPRI scoring experiments. Following the CAPRI assessment protocol, we selected only the top 10 ranked models for assessing the performance of iScore and GraphRank. This was done by selecting the top two models from each of the top five clusters for each target.

The scoring performance of iScore and GraphRank on the 13 CAPRI targets containing near-native models is summarized in Table 2, together with the performance of the best scoring function/group in CAPRI for each target. Details of the performance of the various scoring functions compared for these targets are available in Supplementary Table S3. Again, iScore outperforms GraphRank (Table 2) demonstrating the synergistic effects of conservation information and the interacting energies in differentiating near-native models from docking artefacts. Further, iScore selected near-native models on the top10 for 9 out of 13 targets, with 2 targets having high-quality models and 5 having medium-quality models. As a comparison, selecting for each target the best CAPRI scoring function/group resulted in 10 out of 13 correctly predicted targets, with 4 and 3 targets having at least one high-quality and medium-quality models, respectively.

Overall, iScore ranks among the top scorers on these 13 CAPRI scoring targets (Table 3). In total 37 scoring functions/groups were assessed (Supplementary Table S3), but only those that participated to at least 5 targets are shown in Table 3. When considering the common submitted targets (Supplementary Table S3), iScore still competes with the Weng group (8/2***/4** versus 8/3***/2**), the Bonvin group (8/2***/4** versus 8/2***/3**) and the Bates group (8/2***/4** versus 8/1***/4**). It should be noted that the CAPRI scoring groups, e.g. Weng and Bonvin groups, selected the 10 models with help of human expertise, while our selections were only generated from iScore and GraphRank without manual selection. Furthermore, the results clearly demonstrate the importance of the PSSM feature: GraphRank, using only the PSSM feature, already performs quite well (ranked in the 4th position).

Table 2. Comparison of GraphRank and iScore with CAPRI best performing group per target on the CAPRI score set

CAPRI targets	GraphRank	iScore	CAPRI best	# Total models	#Near-native
T29	4	4	9/5**	1979	166
T30	0	0	0	1148	2
T32	4/1**	4/1**	2	599	15
T35	0	0	1	497	3
T37	2/1**	4/2**	6/1***	1364	97
T39	0	0	0	1295	4
T40	4/3**	4/1***	10/10***	1987	535
T41	8	10/2**	10/2***	1101	347
T46	3	4	4	1570	24
T47	8/5***/3**	10/6***/4**	10/10***	1015	608
T50	0	4/3**	7/6**	1447	133
T53	5/1**	5/1**	8/3**	1360	122
T54	0	0	0	1304	19
Total	8/1***/4**	9/2***/5**	10/4***/3**		

Note: 10 models are selected and evaluated. The values are labeled in green/red when the performance of our scoring functions is better/worse than the CAPRI best scoring group. The scoring performance for each target is reported as the number of acceptable or better models (hits), followed by the number of high (indicated with ***) or medium quality models (**). For example, 8/2** means that there are totally 8 hits among the top 10 models, 2 models out of which are medium-quality models. The overall performance of each method on all 13 targets (the last row) is reported in a similar way. For example, 9/2***/5** means that a scoring function is successful in 9 targets, 2 targets out of 9 have at least a *** model and 5 out of 9 have at least a ** model in the top 10. Note that the CAPRI best column consists of results from 37 different groups (refer to Table 3 for a comparison of the performance per group and Supplementary Table S3 per target).

Table 3. Rankings of GraphRank and iScore in comparison with the scorer groups on the CAPRI score set

	Performance	# Submitted targets
iScore	9/2***/5**	13
Weng	8/3***/2**	9
Bonvin	8/2***/3**	9
Bates	8/1***/4**	10
GraphRank	8/1***/4**	13
Zou	7/4***/1**	9
Wang	6/2***/3**	6
Fernandez-Recio	5/2***/3**	8
Elber	5/1***/1**	5
Wolfson	4/1***	5
Camacho	3/2***/1**	5
... and many others		

Note: In total 37 scorer groups were assessed (Supplementary Table S3), but only scorer groups that have submitted predictions for at least 5 out of the 13 CAPRI targets are shown here. The scoring functions/groups are ordered based on their performance. Number of targets with submitted predictions are shown for each function/group.

4 Discussion

We have developed a novel graph-kernel based scoring function, iScore, for scoring and ranking docking models of protein–protein complexes. By benchmarking on docking models from four different docking programs, iScore shows competitive or better success rate than the original scoring functions of those docking programs. Further, validation on CAPRI targets and comparison with CAPRI scorer groups highlight the high performance of iScore, which achieves the top success rate with acceptable or better models selected for 9 out of 13 CAPRI targets. It is worth noting that both GraphRank and iScore were trained on a rather small dataset, using a very limited set of features, only one for GraphRank and four for iScore. We can expect to further improve the performance of iScore, by increasing the size of the training set and enriching the node and edge labels of interface graphs. Our iScore software with MPI

(Message Passing Interface) and GPU supports can be freely downloaded from: <https://github.com/DeepRank/iScore>. Currently, it takes about 15 min to rank 1619 models of a recent CAPRI target (a 6 domain protein, ranging from 83 to 112 amino acids) using 12 CPU cores (data for this CAPRI round not published yet).

The usage of graph kernel on labeled graphs in iScore provides a novel way to score docking models. SPIDER (Khashan et al., 2012) is also a graph-based scoring function but is drastically different from our GraphRank hence also iScore. SPIDER identifies common interface residue patterns (i.e. interfacial graph motifs) in native complexes and rank a docking model by counting the frequency of the interfacial graph motifs. First of all, GraphRank is based on graph kernel functions to calculate the interface similarities between a docking model and the training complexes while SPIDER is based on the frequent graph mining technique to identify interfacial graph motifs. Second, and importantly, the graphs used in SPIDER has only node labels with amino acid identity, while our GraphRank framework can potentially explore not only the properties of individual interface residues with node labels, but also the features of contacts between residues with edge labels. While we have only used node labels in this work (residue conservation profiles), the concept can easily be extended to add labels to the graph edges, for example in the form of residue–residue interaction energies and coevolution information. Third, iScore uses multi-scale representations of docked interfaces by combining atom-level energy terms with residue-level graph similarities, which allows to account for both subtle differences in 3D space, interaction topology and residue conservations at the same time.

Both conservation profiles and intermolecular energies are important features for scoring of PPIs. Our scoring function GraphRank, using only conservation profiles of the interface residues as features, already shows a promising scoring performance. Physical energies have been widely used and identified as important features in state-of-the-art scoring functions and are complementary to evolutionary information. Considering the successful applications of intermolecular energies in existing scoring functions, in this work we simply combined three intermolecular energetic terms from HADDOCK with the conservation profiles-based GraphRank score. The resulting scoring function iScore outperforms GraphRank,

indicating the significance of considering both evolutionary and energetic information in characterizing PPIs.

When comparing the performance of iScore on models from different docking programs on BM5 new data, we do observe that iScore is able to improve the ranking over the original scoring functions for the rigid-body docking programs (pyDock and ZDock), while iScore does not really outperform the flexible docking programs like HADDOCK and SwarmDock which generate more optimized interfaces (Fig. 3). This might be related to the structure quality of the docking models. For docking models from flexible docking, their structures are already optimized to release steric clashes, while the rigid-body programs usually do not have such an optimization step, leading to unnatural interactions (clashes) within structures. To improve the structure quality of the docking models, we did apply a short energy minimization to optimize the structures before calculating intermolecular energies. With higher structure quality, like those coming out of SwarmDock and HADDOCK, the impact of this short minimization is smaller, and the resulting improvement of iScore versus the original scoring functions is less.

Note that the current version of iScore does not work on antibody-antigen complexes, because PSSMs do not capture interface conservation in such complexes. Incorporation of antibody-antigen specific features into iScore is a topic of our ongoing work.

By introducing the labeled graphs and graph kernel in our scoring function iScore, we pave the way for exploring more detailed features in the graph presentation of protein-protein complexes. Natural extensions of this work will be to include edge labels, for example residue-residue interaction energies and co-evolution. Considering graphs are natural representations of biomolecules, this general framework should be useful for other important macromolecular interaction related topics, such as binding affinity predictions, hot-spot predictions and rational design of protein interfaces.

Acknowledgements

We thank Dr Iain H. Moal (EBI Hinxton, UK) for providing IRaPPA ranking results and docking models of SwarmDock, pyDock and ZDock. We thank Dr Yasser EL-Manzalawy from Penn State University and MSc. Mick Walter from Utrecht University for helpful discussions. The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

Funding

This work was supported in part by the European H2020 e-Infrastructure grant BioExcel (grant no. 675728). C.G. acknowledges financial support from the China Scholarship Council (grant no. 201406220132). L.X. acknowledges financial support from the Netherlands Organisation for Scientific Research (Veni grant 722.014.005) and an Accelerating Scientific Discovery (ASDI) grant from the Netherlands eScience Center (grant no. 027016G04). The work of V.H. was supported in part by the National Center for Advancing Translational Sciences, National Institutes of Health through the grant UL1 TR000127 and TR002014, by the National Science Foundation, through the grants 1518732, 1640834 and 1636795, the Pennsylvania State University's Institute for Cyberscience and the Center for Big Data Analytics and Discovery Informatics, the Edward Frymoyer Endowed Professorship in Information Sciences and Technology at Pennsylvania State University and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science funded by the Pratiksha Trust at the Indian Institute of Science. YJ was supported in part by a research assistantship funded by the Center for Big Data Analytics and Discovery Informatics at Pennsylvania State University.

Conflict of Interest: none declared.

References

- Aloy, P. and Russell, R.B. (2006) Structural systems biology: modelling protein interactions. *Nat. Rev. Mol. Cell Biol.*, **7**, 188–197.
- Altschul, S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
- Andreani, J. and Guerois, R. (2014) Evolution of protein interactions: from interactomes to interfaces. *Arch. Biochem. Biophys.*, **554**, 65–75.
- Andreani, J. *et al.* (2013) InterEvScore: a novel coarse-grained interface scoring function using a multi-body statistical potential coupled to evolution. *Bioinformatics*, **29**, 1742–1749.
- Borgwardt, K.M. *et al.* (2005) Protein function prediction via graph kernels. *Bioinformatics*, **21**, i47–i56.
- Bourquard, T. *et al.* (2011) A collaborative filtering approach for protein-protein docking scoring functions. *PLoS One*, **6**, e18541.
- Brünger, A.T. *et al.* (1998) Crystallography & NMR system: a new software suite for macromolecular structure determination. *Acta Crystallogr. Sect. D Biol. Crystallogr.*, **54**, 905–921.
- Bunke, H. and Riesen, K. (2011) Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recogn.*, **44**, 1057–1067.
- Chang, C.-C. and Lin, C.-J. (2011) LIBSVM. *ACM Trans. Intell. Syst. Technol.*, **2**, 1–27.
- Chang, S. *et al.* (2008) Amino acid network and its scoring application in protein-protein docking. *Biophys. Chem.*, **134**, 111–118.
- Cheng, T.M.K. *et al.* (2007) pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins Struct. Funct. Bioinform.*, **68**, 503–515.
- Croft, W.B. *et al.* (2010) *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Boston.
- de Oliveira, S. and Deane, C. (2017) Co-evolution techniques are reshaping the way we do structural bioinformatics. *F1000Research*, **6**, 1224.
- de Vries, S.J. and Bonvin, A.M.J.J. (2011) CPORT: a consensus interface predictor and its performance in prediction-driven docking with HADDOCK. *PLoS One*, **6**, e17695.
- Dominguez, C. *et al.* (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.*, **125**, 1731–1737.
- Fernández-Recio, J. *et al.* (2004) Identification of protein-protein interaction sites from docking energy landscapes. *J. Mol. Biol.*, **335**, 843–865.
- Fink, F. *et al.* (2011) PROCOS: computational analysis of protein-protein complexes. *J. Comput. Chem.*, **32**, 2575–2586.
- Gärtner, T. *et al.* (2003) On graph kernels: hardness results and efficient alternatives. In: Schölkopf, B. and Warmuth, M.K. (eds) *Learning Theory and Kernel Machines, Lecture Notes in Computer Science*. Springer, Berlin, pp. 129–143.
- Ghosh, S. *et al.* (2018) The journey of graph kernels through two decades. *Comput. Sci. Rev.*, **27**, 88–111.
- Grosdidier, S. *et al.* (2007) Prediction and scoring of docking poses with pyDock. *Proteins Struct. Funct. Bioinform.*, **69**, 852–858.
- Halperin, I. *et al.* (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins*, **47**, 409–443.
- Hopf, T.A. *et al.* (2014) Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife Sci.*, **3**, 65.
- Huang, S.-Y. (2014) Search strategies and evaluation in protein-protein docking: principles, advances and challenges. *Drug Discov. Today*, **19**, 1081–1096.
- Hwang, H. *et al.* (2010) Protein-protein docking benchmark version 4.0. *Proteins Struct. Funct. Bioinform.*, **78**, 3111–3114.
- Janin, J. (2002) Welcome to CAPRI: a Critical Assessment of PRedicted Interactions. *Proteins Struct. Funct. Bioinform.*, **47**, 257–257.
- Jiménez-García, B. *et al.* (2013) pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring. *Bioinformatics*, **29**, 1698–1699.
- Jorgensen, W.L. and Tirado-Rives, J. (1988) The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.*, **110**, 1657–1666.

- Karaca, E. et al. (2013) On the usefulness of ion-mobility mass spectrometry and SAXS data in scoring docking decoys. *Acta Crystallogr. Sect. D Biol. Crystallogr.*, **69**, 683–694.
- Khashan, R. et al. (2012) Scoring protein interaction decoys using exposed residues (SPIDER): a novel multibody interaction scoring function based on frequent geometric patterns of interfacial residues. *Proteins Struct. Funct. Bioinform.*, **80**, 2207–2217.
- Kiel, C. et al. (2008) Analyzing protein interaction networks using structural information. *Annu. Rev. Biochem.*, **77**, 415–441.
- Lensink, M.F. and Wodak, S.J. (2010) Docking and scoring protein interactions: cAPRI 2009. *Proteins Struct. Funct. Bioinform.*, **78**, 3073–3084.
- Lensink, M.F. and Wodak, S.J. (2013) Docking, scoring, and affinity prediction in CAPRI. *Proteins Struct. Funct. Bioinform.*, **81**, 2082–2095.
- Lensink, M.F. and Wodak, S.J. (2014) Score_set: a CAPRI benchmark for scoring protein complexes. *Proteins Struct. Funct. Bioinform.*, **82**, 3163–3169.
- Lensink, M.F. et al. (2007) Docking and scoring protein complexes: cAPRI 3rd edition. *Proteins Struct. Funct. Bioinform.*, **69**, 704–718.
- Lensink, M.F. et al. (2017) Modeling protein–protein and protein–peptide complexes: cAPRI 6th edition. *Proteins Struct. Funct. Bioinform.*, **85**, 359–377.
- Melquiond, A.S.J. et al. (2012) Next challenges in protein–protein docking: from proteome to interactome and beyond. *Wiley Interdiscipl. Rev. Comput. Mol. Sci.*, **2**, 642–651.
- Moal, I.H. et al. (2017) IRaPPA: information retrieval based integration of biophysical models for protein assembly selection. *Bioinformatics*, **33**, 1806–1813.
- Moal, I.H. et al. (2013) Scoring functions for protein–protein interactions. *Curr. Opin. Struct. Biol.*, **23**, 862–867.
- Moont, G. et al. (1999) Use of pair potentials across protein interfaces in screening predicted docked complexes. *Proteins Struct. Funct. Bioinform.*, **35**, 364–373.
- Pierce, B. and Weng, Z. (2007) ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins Struct. Funct. Bioinform.*, **67**, 1078–1086.
- Pierce, B.G. et al. (2014) ZDOCK server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, **30**, 1771–1773.
- Pons, C. et al. (2011a) Prediction of protein-binding areas by small-world residue networks and application to docking. *BMC Bioinformatics*, **12**, 378.
- Pons, C. et al. (2011b) Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein–protein docking. *J. Chem. Inf. Model.*, **51**, 370–377.
- Rodrigues, J.P.G.L.M. and Bonvin, A.M.J.J. (2014) Integrative computational modeling of protein interactions. *FEBS J.*, **281**, 1988–2003.
- Rodrigues, J.P.G.L.M. et al. (2012) Clustering biomolecular complexes by residue contacts similarity. *Proteins Struct. Funct. Bioinform.*, **80**, 1810–1817.
- Shoemaker, B.A. and Panchenko, A.R. (2007) Deciphering protein–protein interactions. Part I. experimental techniques and databases. *PLoS Comput. Biol.*, **3**, e42.
- Soni, N. and Madhusudhan, M.S. (2017) Computational modeling of protein assemblies. *Curr. Opin. Struct. Biol.*, **44**, 179–189.
- Stein, A. et al. (2011) Three-dimensional modeling of protein interactions and complexes is going 'omics. *Curr. Opin. Struct. Biol.*, **21**, 200–208.
- Torchala, M. et al. (2013) SwarmDock: a server for flexible protein–protein docking. *Bioinformatics*, **29**, 807–809.
- Tress, M. et al. (2005) Scoring docking models with evolutionary information. *Proteins Struct. Funct. Bioinform.*, **60**, 275–280.
- van Zundert, G.C.P. et al. (2016) The HADDOCK2.2 web server: user-friendly integrative modeling of biomolecular complexes. *J. Mol. Biol.*, **428**, 720–725.
- Vangone, A. and Bonvin, A.M. (2015) Contacts-based prediction of binding affinity in protein–protein complexes. *eLife Sci.*, **4**, e07454.
- Vangone, A. et al. (2017) Prediction of biomolecular complexes. In: Rigden, D.J. (ed) *From Protein Structure to Function with Bioinformatics*. Springer, Dordrecht, Netherlands, pp. 265–292.
- Vangone, A. et al. (2016) Sense and simplicity in HADDOCK scoring: lessons from CASP-CAPRI round 1. *Proteins Struct. Funct. Bioinform.*, **85**, 417–423.
- Vapnik, V.N. et al. (2013) *The Nature of Statistical Learning Theory*. Springer Science & Business Media, New York, NY.
- Vento, M. (2015) A long trip in the charming world of graphs for pattern recognition. *Pattern Recogn.*, **48**, 291–301.
- Vishwanathan, S.V.N. et al. (2010) Graph Kernels. *J. Mach. Learn. Res.*, **11**, 1201–1242.
- Vreven, T. et al. (2015) Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *J. Mol. Biol.*, **427**, 3031–3041.
- Wang, Y. et al. (2013) A Theoretical Analysis of NDCG Type Ranking Measures. In: *Conference on Learning Theory*, pp. 25–54.
- Xue, L.C. et al. (2015) Computational prediction of protein interfaces: a review of data driven methods. *FEBS Lett.*, **589**, 3516–3526.
- Xue, L.C. et al. (2014) DockRank: ranking docked conformations using partner-specific sequence homology-based protein interface prediction. *Proteins Struct. Funct. Bioinform.*, **82**, 250–267.
- Zacharias, M. (2003) Protein–protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Sci.*, **12**, 1271–1282.
- Zhou, H. and Zhou, Y. (2002) Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.*, **11**, 2714–2726.