ELSEVIER

Teaser This review discusses current search strategies and evaluation methods for investigating protein–protein docking, two important issues that are quite different from those of protein–ligand docking.

Search strategies and evaluation in protein–protein docking: principles, advances and challenges

Sheng-You Huang[1,2,3]

[1] Research Support Computing, University of Missouri, Columbia, MO 65211, USA
[2] University of Missouri Bioinformatics Consortium, University of Missouri, Columbia, MO 65211, USA
[3] Department of Computer Science, University of Missouri, Columbia, MO 65211, USA

Protein–protein docking is attracting increasing attention in drug discovery research targeting protein–protein interactions, owing to its potential in predicting protein–protein interactions and identifying 'hot spot' residues at the protein–protein interface. Given the relative lack of information about binding sites and the fact that proteins are generally larger than ligand, the search algorithms and evaluation methods for protein–protein docking differ somewhat from those for protein–ligand docking and, hence, require different research strategies. Here, we review the basic concepts, principles and advances of current search strategies and evaluation methods for protein–protein docking. We also discuss the current challenges and limitations, as well as future directions, of established approaches.

Sheng-You Huang has been working in the molecular docking field since 2003, and on protein–protein docking in particular since 2007. He is currently a system administrator in the Research Support Computing and the University of Missouri Bioinformatics Consortium at the University of Missouri. He is also a research assistant professor in the Department of Computer Science at the University. His research interests include molecular modeling, bioinformatics, and computational biophysics and/or biology and their application to drug discovery. He actively develops novel docking algorithms and energy-scoring functions for protein–ligand interactions, protein–protein interactions, protein–RNA interactions, and modeling of quantitative structure-function relationships of therapeutically important proteins. He obtained his PhD in 2003 from the Wuhan University where he studied computer simulations in aspects of soft matter and biological physics. His research has resulted in the publication of scientific software, book chapters and more than 60 peer-reviewed journal articles.

Given that protein–protein interactions have an important role in many biological functions in living organisms [1], determination of the structure of the protein–protein complexes involved in these interactions is vital for revealing biological process pathways, to investigate the mechanisms interacting between proteins and to identify the crucial 'hot spot' residues in interactions that are important for drug discovery [2–9]. With the rapid development of structural proteomics projects, the 3D structures of many protein–protein complexes have been determined using various techniques, such as X-ray crystallography and nuclear magnetic resonance spectroscopy, and have been deposited in the Protein Data Bank (PDB) [10]. However, compared with the progress achieved for individual proteins, development in experimentally determining the complex structures between proteins is still limited because of the technical difficulties and high cost of the experimental methods involved [11]. Therefore, computational tools, such as protein–protein docking, which predicts the binding mode and free energy between individual protein structures, are needed to complement the experimental methods for the identification of protein–protein interactions and determination of their complex structures. Since pioneering work by Janin and Wodak [12], the protein–protein docking field has advanced considerably and many protein–protein docking algorithms have been developed over the past two decades [13–22].

E-mail address: huangshe@missouri.edu.

1359-6446/06/$ - see front matter http://dx.doi.org/10.1016/j.drudis.2014.02.005

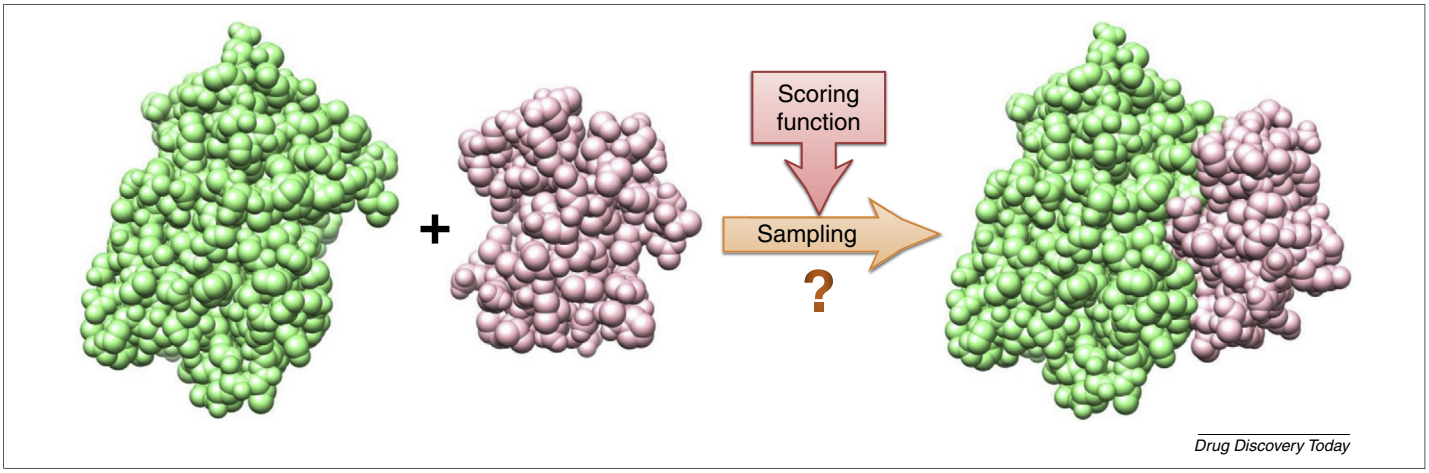

FIGURE 1

An illustration of protein–protein docking where the binding complex of two individual proteins (PDB code 1UDI) is constructed by sampling putative binding conformations that are evaluated and ranked by a scoring function.

Similar to protein–ligand (small molecule) docking [23], protein–protein docking also comprises two important components: sampling and scoring (Figure 1) [18]. These two processes can be coupled together during the docking process or can occur in different stages as adopted in post-docking approaches. Sampling is a search process that generates possible binding orientations and/or conformations (i.e. modes) between two molecules. It can be further divided into (i) rigid-body sampling of binding orientations; and (ii) conformational sampling of molecules, whereby rigid-body sampling is performed by the orientational search algorithm and conformational sampling is achieved by explicit protein flexibility consideration. Scoring is the measurement using a scoring function of the binding tightness and/or score between two molecules in a binding mode. The evaluated binding modes are then ranked according to their binding scores so that a set number of top binding modes can be selected as the final docking solutions. Therefore, in molecular docking, up to three aspects (orientational search, protein flexibility, and scoring functions) can be involved in a docking process.

Although sampling and scoring are the two main components for both protein–protein and protein–ligand docking, they do not necessarily use the same algorithm. For scoring, both protein–protein docking and protein–ligand docking use similar types of scoring method, which can normally be grouped into three basic categories: (i) force-field based; (ii) knowledge based; and (iii) empirical, as well as a combination of two or all of them [21,24]. In addition, because of the same type of targets (i.e., proteins) in both protein–protein and protein–ligand docking, algorithms that consider protein flexibility are also similar in both docking types; these normally include side chain and/or backbone flexibility, loop rearrangements, domain movements, and so on [14–18]. However, because less information is available relating to binding sites and the large size of proteins in protein–protein docking [18,23], the orientational search algorithm often requires strategies for protein–protein docking that are different from those for protein–ligand docking. Thus, many global and/or local search strategies have been developed for various protein–protein docking algorithms. In addition, given the larger size of proteins and larger binding interface, the evaluation method for protein–protein docking is also different from that for protein–ligand docking. Here, we give a detailed overview of the basic concepts, principles and specific features of current search strategies and evaluation methods in protein–protein docking. We also discuss challenges and limitations in existing algorithms and make suggestions for potential future research directions.

Protein–protein docking: an overview of search strategies

The search strategies in currently available protein–protein docking algorithms can be grouped into three basic categories [(i) exhaustive global search; (ii) local shape feature matching; and (iii) randomized search] and one broad category of post-docking approaches (Table 1).

Exhaustive global search

As mentioned above, because of a lack of information about binding sites, the investigation of protein–protein docking requires a global search for the binding orientations over six degrees of freedom (3D translational plus 3D rotational). Therefore, theoretically, the computational cost for an exhaustive global search has an order of $O(N^6)$, where $O(N^3)$ is from a 3D translational search and $O(N^3)$ is for a 3D rotational search. In an actual docking, one protein is normally fixed (the so-called 'static molecule') and the other protein is moved around the static protein (the so-called 'moving molecule'). Search over three rotational degrees of freedom is often separated from that over three translational degrees of freedom, in that the moving molecule is first rotated by an Euler angle in 3D rotational space. Then, for the rotation, an exhaustive search is carried out for the moving protein relative to the static protein in the complete 3D translational space. The above process is repeated until the entire 3D rotational space is sampled completely.

Given the typical size of approximately 60 Å for a protein, the search for the relative translations of two proteins will need to cover approximately $120 \times 120 \times 120$ Å^3 in the 3D translational space for a single rotation. If a grid spacing of 1.2 Å is used for discretizing the translational space during the search, there will be

TABLE 1

Search strategies in protein–protein docking

Search algorithms	Examples of docking programs	Refs
Exhaustive global search		
FFT-based search	FTDock, GRAMM, DOT, ZDOCK, MolFit, PIPER, F2DOCK, SDOCK, ASPDock, Cell-Dock	[25–41]
Spherical Fourier transform-based search	HEX, FRODOCK	[45–47]
Direct search in Cartesian space	SOFTDOCK, BIGGER, SKE-DOCK	[49–51]
Local shape feature matching		
Distance geometry algorithm	DOCK	[52]
Geometric hashing	PatchDock, SymmDock, LZerD	[53–56]
Genetic algorithm	GAPDOCK	[57]
Randomized search		
Monte Carlo search	RosettaDock, ICM-DISCO, ATTRACT, HADDOCK	[61–71]
Particle swarm optimization	SwarmDock	[72]
Genetic algorithm	AutoDock	[73]
Post-docking approach		
Using advanced scoring functions	RPScore, ZRANK, PyDock, EMPIRE, DARS, DECK, SIPPER, PIE, MDockPP, etc.	[81–94]
Considering protein flexibility	MultiDock, SmoothDock, RDOCK, FireDock, FiberDock, EigenHex, etc.	[95–104]
Other ranking protocols	SDU, CyClus, CONSRANK, etc.	[105–111]

a total of approximately $100 \times 100 \times 100 = 10^6$ relative translations. For the search in 3D rotational space, there will be a total of approximately 10 000 rotations for the moving protein if an Euler angle interval of 12° is used. Thus, the total binding orientations over a 6D rotational and translational space will be approximately $10^6 \times 10\,000 = 10^{10}$. The number will be larger if a finer grid spacing or Euler angle interval is used for higher-resolution docking. Therefore, it is computationally expensive and possibly even prohibitive to evaluate such a huge number of binding complexes using a standard search method. To reduce the computational cost, two types of approach have been developed for this type of exhaustive global search: fast Fourier transform (FFT) correlation and direct search algorithms.

FFT correlation

Despite a huge number of binding positions, the search over the whole 3D translational space can be completed in one go in the imaginary space through several FFT calculations, which were first introduced by Katchalski-Katzir and colleagues [25]. The FFT-based algorithm accelerates the search process over three translational degrees of freedom and reduces the computational time from conventional $O(N^3)$ to an order of $O[\log(N^3)]$. This results in a total search time of $O[N^3 \log(N^3)]$ over the complete 6D space, making an exhaustive global docking calculation practical on a personal computer. Given its high computational efficiency, this ground-breaking study resulted in a number of FFT-based docking programs, such as FTDock [26], GRAMM [27], DOT [28,29], ZDOCK [30–34], MolFit [25,35], PIPER [36], F2DOCK [37], SDOCK [38] and Cell-Dock [39], which are the largest group of global protein–protein docking algorithms in the field [11].

There is the same basic principle behind different FFT-based search algorithms. The differences among current FFT-based docking programs stem mainly from how the potentials of protein atoms are mapped onto a grid and what potentials and/or scoring functions are used to describe the interactions between proteins. Here, we describe how the search over 3D translational space can be achieved through an FFT-based algorithm using the most basic shape complementarity-scoring scheme (a 2D version is shown in Figure 2a) [25]. Given a pair of proteins, the static one is called the receptor protein (R) and the moving one is named the ligand protein (L) for easy reference. Both proteins are centered in a discretized grid of dimensions $N \times N \times N$ in Cartesian space. Energy terms for protein atoms are then projected onto the grid, constructing a grid representation for the receptor protein $\mathbf{R} = \{R(l, m, n), 1 \leq l, m, n \leq N\}$ with Eq. (1):

$$R(l,m,n) = \begin{cases} 1: \text{ on the surface of the protein} \\ \rho: \text{ inside the molecule} \\ 0: \text{ outside the protein} \end{cases}, \quad (1)$$

and a grid representation for the ligand protein $\mathbf{L} = \{L(l, m, n), 1 \leq l, m, n \leq N\}$ with Eq. (2):

$$L(l,m,n) = \begin{cases} 1: \text{ on the surface of the protein} \\ \delta: \text{ inside the molecule} \\ 0: \text{ outside the protein} \end{cases}, \quad (2)$$

where ρ is a number $\ll -1$ and δ has a small positive value between 0 and 1 [25]. Thus, the binding score $C(p, q, r)$ of the ligand protein relative to the receptor protein for a translation (p, q, r) in the 3D grid space can be calculated based on their molecular shape complementarity by summing up the overlap products $R(l, m, n) \times L(l + p, m + q, n + r)$ between the two grids as follows [25] (Eq. (3)):

$$C(p,q,r) = \sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N} R(l,m,n) \times L(l+p, m+q, n+r), \quad (3)$$

where periodic boundary conditions are applied during the summation when the indices fall beyond the grid. Then, the next step in the search over the complete 3D translational space will be to calculate the above correlation score in Eq. (3) for all possible translations of $1 \leq p, q, r \leq N$ in the 3D grid space, resulting in N^3 calculations with N^3 scores. Here, the search for the N^3 scores $\mathbf{C} = \{C(p, q, r), 1 \leq p, q, r \leq N\}$ can be completed in one go through three FFT calculations as follows (Figure 2a; Eq. (4)):

$$\mathbf{C} = FFT^{-1}(\overline{FFT(\mathbf{R})} \times FFT(\mathbf{L})), \quad (4)$$

where $FFT(\mathbf{X})$ represents applying a forward discrete Fourier transform to X, $FFT^{-1}(\mathbf{X})$ represents an inverse Fourier transform for $\mathbf{X}$, and $\bar{\mathbf{X}}$ is the conjugate of the complex number $\mathbf{X}$.

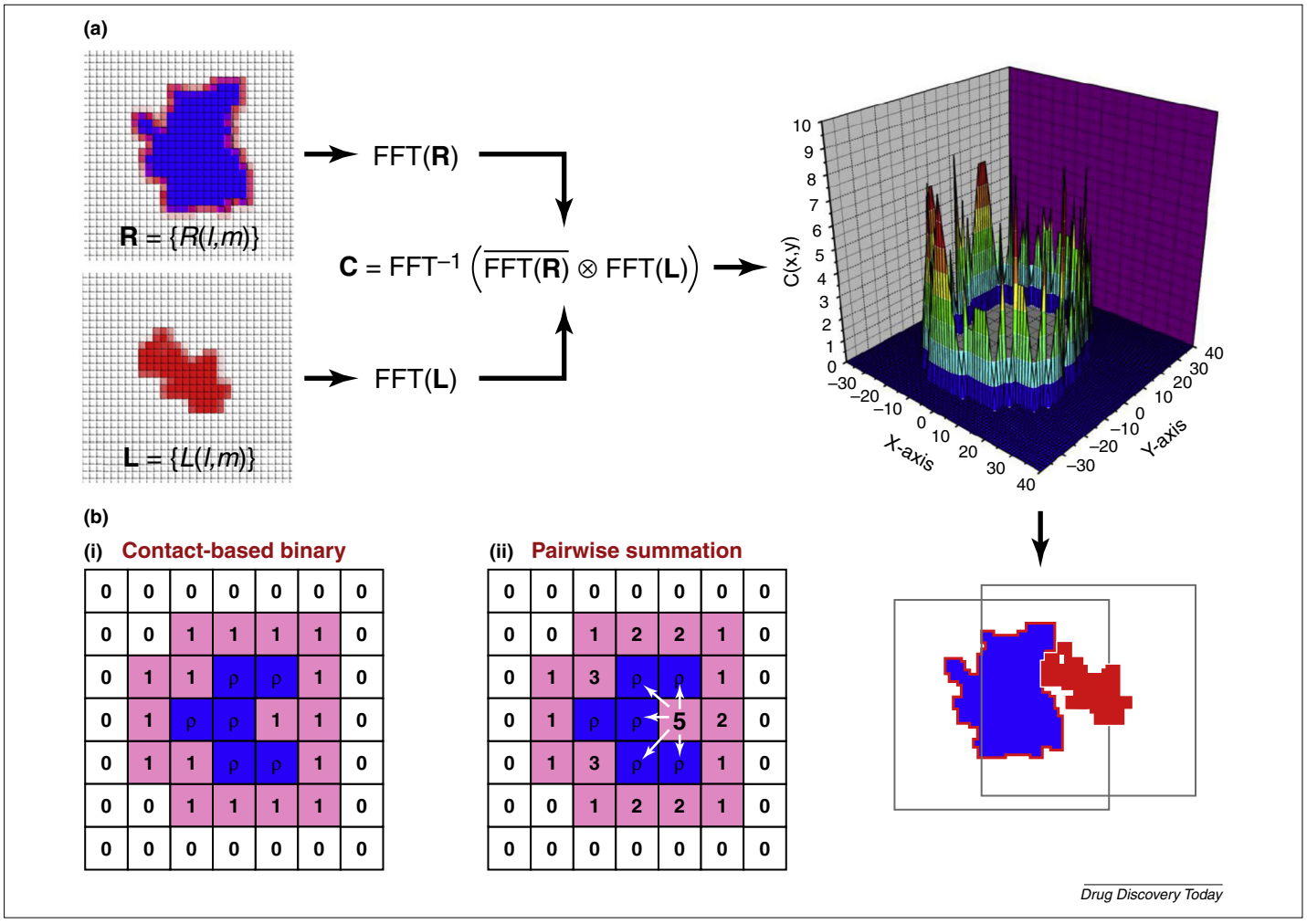

FIGURE 2

An illustration of fast Fourier transform (FFT)-based protein-protein docking algorithms. **(a)** An example of the FFT-based docking process in a 2D Cartesian space, where $\rho = -15$ and $\delta = 1$ for Eqs. (1) and (2), respectively. First, both the receptor (R) and ligand (L) proteins are mapped onto a grid of dimensions $N \times N$ as $\mathbf{R} = \{R(l, m), 1 \leq l, m \leq N\}$ and $\mathbf{L} = \{L(l, m), 1 \leq l, m \leq N\}$, respectively. The matching scores (or correlations) $\mathbf{C} = \{C(x, y), 1 \leq x, y \leq N\}$ for all relative translations between two grids can be obtained simultaneously through two forward and one backward FFT calculations. Here, the larger value of $C(x, y)$ means the better correlation or matching score between two proteins. A final complex with a score $C(x0, y0)$ can be constructed by applying the relative translations $x0$ and $y0$ to the ligand grid in the X- and Y-axes, respectively. **(b)** Two types of method for mapping the potentials of protein atoms onto a grid. Taking the shape mapping of the receptor protein as an example, the grid has a value of ρ for the points inside the protein (blue). (i) For the contact-based binary method, the other grid points have a value of 1 if they are in contact with the protein core (pink) and 0 if they are not (white). (ii) For the pairwise summation approach, the corresponding values will be the sum of the pairwise potentials from their neighboring grid points inside the protein within a distance cutoff, as the arrows indicate. Here, the distance cutoff to define a neighbor is one grid spacing. The contact-based binary approach is normally used for contact-based potentials, such as atomic contact energy, and the pairwise summation method is often needed for pairwise potentials, such as van der Waals energies, electrostatic interactions and knowledge-based potentials. An FFT-based docking algorithm can include one or both of the two mapping methods, depending on the used potentials.

The above FFT-based search process in 3D translational space is repeated for each of the rotations for the ligand protein until the complete 3D rotational space is sampled. It should be noted that, here, a simple shape complementarity scoring method is only used for illustration purposes. Information about the binding site [30–33,40,41] and other sophisticated scoring schemes, such as pairwise shape complementarity [31,32], electrostatic energy [26,28,35], atomic desolvation effect [30,32,37], knowledge-based potentials [33,34,36] and hydrophobic complementarity [27,42], can also be implemented in FFT-based search algorithms by mapping the information and/or potentials onto to the grid, as shown in Figure 2b. Improvement in both accuracy and running time can also be achieved by considering the symmetry information in homomultimer complexes [43,44].

In the above example, we introduced the FFT-based search in 3D Cartesian grid space. It is also the most commonly used algorithm for FFT-based docking programs, including FTDock [26], GRAMM [27], DOT [28,29], ZDOCK [30–34], MolFit [25,35], PIPER [36], F2DOCK [37], and so on [38–41]. In fact, the Fourier transform (FT) algorithm can also be applied to accelerate the search over 3D rotational space using spherical harmonics expansions, as used in FRODOCK [45] or in both rotational and translational space using spherical polar Fourier correlations, as implemented in HEX [46–48]. Furthermore, the computational efficiency of FT-based algorithms can be further accelerated with the help of advanced software packages, such as the 3D convolution library [34], and new hardware technologies, such as the graphics processing unit (GPU) [48] and Cell BE processor [39].

Direct search in Cartesian space

In addition to FFT-based algorithms, direct search methods can also be performed to find matches between two proteins in 3D Cartesian grid space with the help of some acceleration tactics, as used in SOFTDOCK [49], BIGGER [50] and SKE-DOCK [51]. Taking BIGGER as an example, the molecular shape of two proteins is first mapped onto a 3D grid in Cartesian space. Then, each grid point is given a simple value, such as '1' if it is occupied by the protein or '0' if it is not, for the sake of matching efficiency. This grid-based representation of the system is similar to that used in FFT-based search algorithms except it has simpler values on the grid. The shape matching is then directly performed in the Cartesian grid space to find the geometric fit between two proteins. Methods such as applying Boolean operators and heuristic rules are used to speed up the search process. Although the search efficiency for this type of direct search approach is lower than that for FFT-based search algorithms, the direct search approach is more controllable because of its direct operation in Cartesian space, which makes it easier to consider protein flexibility and incorporate biological information during the search process.

Local shape feature matching

In local shape feature-matching algorithms, proteins are represented by molecular shapes, such as the Connolly surface, which is calculated using a probe molecule and often called the 'solvent' excluded accessible surface. Algorithms are then used to find those matches that give a good local shape complementarity between two proteins (Figure 3). Given the fast speed of its matching process, the algorithm can often generate tens of thousands of binding orientations within minutes and, thus, is able to perform a global search within a practical length of time on a personal computer. In this type of algorithm, search over six degrees of freedom is not as explicitly presented as in exhaustive global search algorithms. The three translations and three rotations are implicitly included in the transformation matrix for a match and will only be calculated when a binding orientation between two

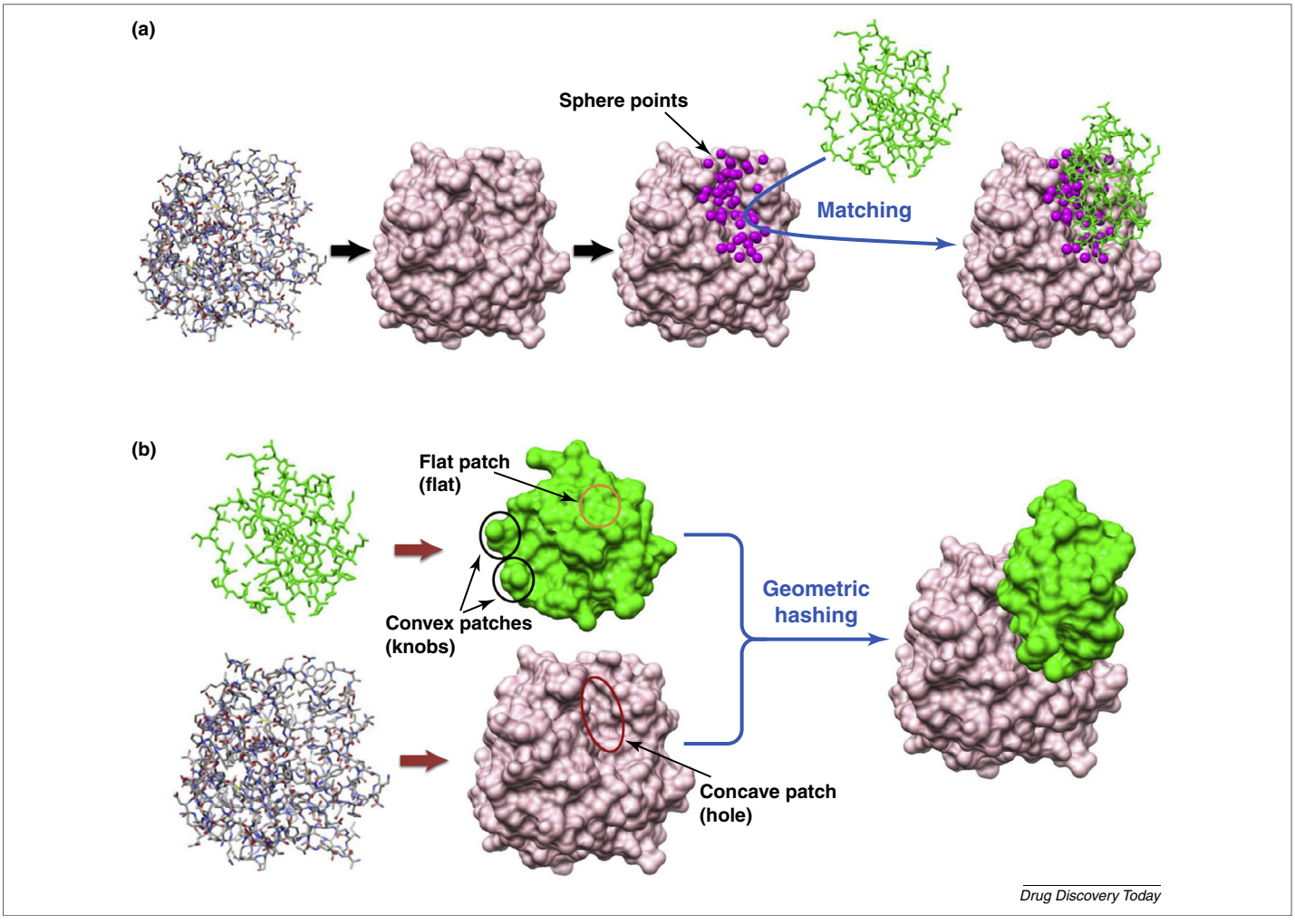

FIGURE 3

An illustration of two typical local shape feature-matching algorithms in protein–protein docking (PDB code 2SNI). **(a)** Distance geometry algorithm used in DOCK: first, the molecular surface is calculated for the receptor protein and then sphere points are generated to represent the shape of the binding site on the receptor protein. In the next step, possible complexes are constructed by matching the atoms of the ligand protein with the sphere points on the receptor protein using a distance geometry algorithm. **(b)** Geometric hashing used in PatchDock: first, the molecular surface is calculated for both the receptor and ligand proteins and a segmentation algorithm is then applied to detect three types of geometric patch of concave (or hole), convex (or knob) and flat surface pieces; possible complexes are then constructed by matching surface patches using the Geometric Hashing technique with the rule that knob (or convex) patches match hole (or concave) patches and flat patches can match any type of patch.

proteins is constructed through the match. Given the nature of local shape matching, many of the binding orientations generated by the algorithm include atomic clashes. As such, steric checking is often used as a first step to filter out those solutions with too many clashes. Local shape feature-matching algorithms also tend to generate more binding orientations towards those sites with good shape complementarity. Therefore, a post-clustering step is often necessary to remove the redundancy in the final solutions. Examples of the docking programs that use the local shape feature-matching algorithm include DOCK [52], PatchDock [53,54], LZerD [55,56] and GAPDOCK [57].

Although DOCK, which was first developed by Kuntz and colleagues [52], is a well-known protein–ligand docking program, it also has the ability to perform protein–protein docking computations (Figure 3a). There are several applications using DOCK for protein–protein docking [58–60]. Given that it is designed for protein–ligand docking, DOCK performs a local search around known binding sites and, therefore, is a local protein–protein docking search program. Given its local search nature, it is fast and can often finish a docking process within minutes or even seconds. It might be suitable for those cases where information is available about the binding site. In DOCK, only one of the proteins needs to be represented through the molecular surface. Specifically, the molecular surface of one protein is calculated using the DMS program. Sphere points, which are the negative images of the surface features, are generated using the SPHGEN program. A certain number of sphere points around the binding site are then selected to represent the molecular shape of the binding site. With the selected sphere points, a search is performed to find possible matches between the sphere points and the ligand atoms using a distance geometry algorithm. A match is considered successful if all the edges in a set of sphere points match those in a set of ligand atoms given a certain distance tolerance. The successful match is then used to construct a possible binding orientation. Steric checking is applied during the matching process. The docking process continues until a certain number of reasonable binding orientations are reached.

Geometric hashing algorithms perform a global protein–protein docking search by finding the local matches of shape descriptors, such as surface patches in PatchDock [53] or 3D Zernike descriptors in LZerD [55], between proteins (Figure 3b). Information can also be incorporated for symmetric docking of homo-multimers [54]. Taking PatchDock as an example, both proteins need to be represented by their molecular shapes through the molecular surface. Specifically, the Connolly surface of the protein is generated using the MS program. Next, a segmentation algorithm is carried out to detect three types of geometric patch: concave, convex and flat pieces on molecular surface. A graph is constructed with the points of the sparse surface. Each node in the graph is then categorized as a 'knob', 'flat' or a 'hole' based on its curvature using a shape function. With the molecular shape represented by a graph, surface patch matching is performed using the Geometric Hashing and Pose-Clustering techniques with the rules that knob (or convex) patches match hole (or concave) patches and flat patches can match any type of patch [53] (Figure 3b). Steric checking is applied to remove those binding complexes with severe penetrations of one protein to the other and a clustering procedure is used to diversify the binding solutions. The final orientations are ranked based on a score of shape complementarity.

Randomized search

Similar to local shape feature matching, randomized search is also commonly used for both local and global search in molecular docking. However, unlike in exhaustive global searches or local shape feature-matching algorithms, there is no need to use a special molecular representation, such as a grid or a shape, in randomized search approaches. Search at the atomic level is common in randomized search algorithms, even though a grid representation of the receptor protein or reduced models of proteins can be used to speed up the search process. The name 'randomized search' might have arisen from the random starting positions and/or the randomized movement of the ligand protein during the search process (Figure 4). Specifically, one protein is fixed that can be atom or grid based depending on the algorithms. Next, the other protein is randomly placed around the binding site for a local search or around the whole static protein molecule for a global search based on a certain number of rules. Algorithms can be used to optimize the placement procedure with information such as molecular shape and/or surface, to generate more reasonable initial binding orientations. Then, from their starting positions, each of the initially generated ligand-binding orientations is optimized and/or refined via a multistage sampling and/or multiscale modeling approach using stochastic algorithms, such as generic algorithms and/or Monte Carlo methods. Similar to local shape feature matching, randomized search does not exhaustively search the complete 6D space. Parameters of each binding orientation corresponding to six degrees of freedom are obtained from its initial placement and adjusted by the subsequent optimization process. Examples of randomized search include RosettaDock [61–64], ATTRACT [65–67], High Ambiguity Driven Protein–Protein Docking (HADDOCK) [68,69], ICM Docking and Interface Side-Chain Optimization (ICM-DISCO) [70,71] and so on [72,73].

RosettaDock normally performs a local protein–protein docking where information about the binding site is known, although a global search can also be achieved [61]. In RosettaDock, random starting positions of binding decoys are created by placing a random rotation of one protein around the binding site of the other protein, followed by translation along the line of protein centers to create glancing contact between the two proteins [61–63]. Symmetry information can be applied to construct homo-oligomeric protein assemblies [74]. The initially constructed binding orientations are then refined by a two-stage multiscale optimization algorithm using the rigid-body Monte Carlo search. In the first stage of low-resolution optimization, reduced models of the proteins are used where each residue is represented by four backbone atoms and one pseudo-atom located at the center of the side chain. In the second stage of high-resolution refinement, the residue-based reduced models are restored to the all-atom proteins by adding explicit side chains to the protein backbones using a backbone-dependent rotamer packing algorithm. Then, rigid-body optimization is performed to find the optimal binding orientation between the proteins with side chains using a Monte Carlo search. In this second stage, the side-chain packing and minimization operations are repeated 50 times to optimize simultaneously both the side-chain conformations and the rigid-body

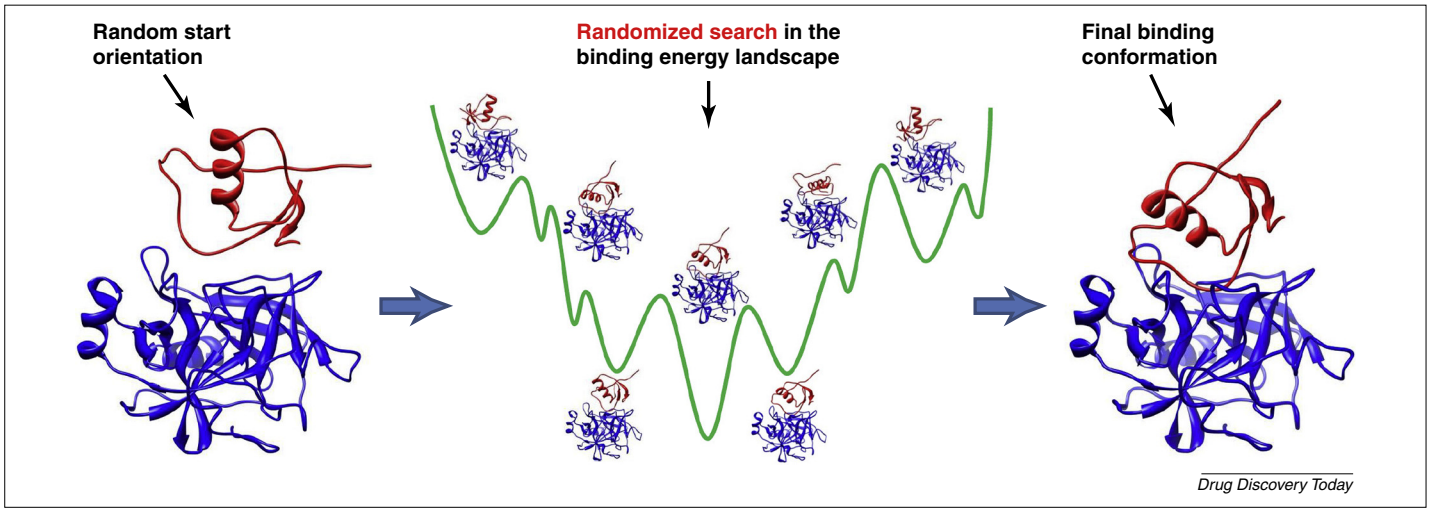

FIGURE 4

An illustration of randomized search algorithms in protein–protein docking (PDB code 1CGI). In this algorithm, the system is normally represented by atom-based models, which can be all-atom proteins or reduced protein models. The search begins by randomly placing the ligand protein with a random orientation somewhere away from the receptor protein. Then, the binding energy landscape is explored by optimizing both the relative orientation and backbone and/or side-chain conformations of two proteins via a multistage sampling and/or multiscale modeling protocol using randomized search algorithms, such as Monte Carlo or genetic algorithms, in the hope of finding the global minimum corresponding to the native binding state.

position. The search procedure is repeated until a certain number of binding conformations are created.

ATTRACT is another global protein–protein docking program that adopts reduced protein models in the initial search process [65,75]. In ATTRACT, each residue of the proteins is represented by up to three pseudo-atoms. Backbone atoms are represented by one pseudo-atom located at the Cα position. Side-chain atoms are represented by another pseudo-atom at the geometric mean of side-chain atoms for small amino acids (Ala, Asp, Asn, Ser, Thr, Val and Pro) or two pseudo-atoms for larger and flexible amino acids. To do a systematic docking, a series of start positions is first generated on the surface of the receptor protein with a probe radius that is slightly larger than the radius of the ligand protein. For each of the starting positions, various initial ligand orientations are constructed by rotating the ligand protein centered at the starting position, and then optimized by a series of energy minimizations using a Monte Carlo search where explicit side-chain and backbone flexibility can be considered [66,67]. A collecting procedure is then carried out on all the optimized binding solutions from different starting positions, such that every docking minimum occurs just once.

Unlike other protein–protein docking programs, HADDOCK [68,69] is specially designed to make use of biochemical and/or biophysical interaction information, such as chemical shift perturbation data from nuclear magnetic resonance (NMR) titration experiments or mutagenesis data. The docking process in HADDOCK is driven by the so-called 'Ambiguous Interaction Restraint' (AIR), which is defined as an ambiguous distance between all residues involved in the interaction. The docking protocol begins with randomization of the initial orientations, where the two proteins are positioned at 150 Å away from each other with random rotations, followed by a series of rigid-body energy minimizations for the orientation. Next, a series of semi-rigid simulated annealing processes is performed to further optimize the binding conformations where both side chains and backbones at the interface are allowed to move. Last, a gentle refinement in Cartesian space is carried out using molecular dynamics (MD) simulations where explicit solvents are included. Final results are clustered to remove the redundancy in the binding conformations.

Although ICM-DISCO [70,71] can be used to perform a global energy optimization from multiple starting positions of the ligand using a randomized search algorithm, it is more representative of a post-docking approach that includes two-step docking processes of rigid body-docking simulations and side-chain refinement of ligand-binding residues [71]. First, multiple starting positions of the ligand protein, which are defined as a series of points with an average distance of 15 Å between neighbors, are generated around the receptor protein. Next, the sampling of the ligand protein starting from each of the starting positions is carried out in a grid of soft receptor potentials by a pseudo-Brownian Monte Carlo minimization. Then, a certain number of low-energy binding orientations from the previous step are further refined to consider the induced changes of ligand-binding residues using a global energy optimization with fully flexible side chains.

Post-docking approaches

Strictly speaking, post-docking approaches are not a true category of search algorithms, but a type of docking-refinement protocol that does not have to have its own search algorithm. A post-docking approach normally has a hierarchical structure including at least two stages of procedure. First, possible binding orientations and/or conformations are sampled by an initial docking program, which can be any of three search algorithms as discussed above. Then, a certain number of top binding solutions from the first step, which can range from hundreds to tens of thousands of binding orientations and/or conformations, are further optimized and reranked by a more sophisticated scoring technique, where explicit protein flexibility and biological information can be incorporated. The protocol of separating sampling and scoring

significantly simplifies the computational process. The rationale behind post-docking algorithms is that initial protein–protein docking programs are able to generate at least one near-native binding mode that are called 'hits' in a certain number of binding orientations and/or conformations. Given the reasonable success of current protein–protein docking programs in generating hits in a certain number of top orientations and/or conformations, post-docking algorithms have made significant progress and achieved promising success in the community-wide Critical Assessment of PRedicted Interaction (CAPRI) experiments [76–80]. Examples of post-docking algorithms include using more sophisticated scoring functions, such as RPScore [81], ZRANK [82], PyDock [83], EMPIRE [84], DARS [85], DECK [86], SIPPER [87], PIE [88], MDockPP [89,90] and so on [91–94]; explicitly considering protein flexibility, such as MultiDock [95], SmoothDock [96], RDOCK [97], FireDock [98], FiberDock [99], EigenHex [100] and so on [101–104], and other refinement and/or ranking protocols, such as SDU [105], CyClus [106], and so on [107–111].

Challenges and future directions

Although search strategies in protein–protein docking have received considerable attention and have developed significantly in both their level of sophistication and variety, current search algorithms are still far from mature (Table 2). As one of the basic elements in molecular docking, a search algorithm is also closely interconnected with other aspects, such as scoring and flexibility during docking calculations. Therefore, challenges and/or limitations exist not only in developing new search algorithms or improving current search strategies, but also in optimizing the interadaptation between a search algorithm and other elements, such as scoring functions and protein flexibility in the search process, which highlights future research directions for search algorithms in protein–protein docking.

Integration of scoring functions

It was thought that the integration of sampling and scoring can substantially improve docking results, even though decoupling the two steps can simplify the computational problem [20]. Therefore, on-the-fly scoring during sampling would be ideal. One challenge is how to integrate efficiently the scoring function into the search algorithm during the docking process. Strictly speaking, among all the protein–protein docking strategies, the exhaustive search algorithm is the only truly global search approach in which sampling covers every position of the 6D space. However, exhaustive search-based docking methods normally require grid representation for the system. Therefore, to implement an exhaustive global search, the scoring function must be able to be projected on the grids of two proteins. This brings about two challenges. On the one hand, given the discrete nature of a grid, the scoring function might lose some of its accuracy when it is discretized and mapped onto the grid points [112]. The accuracy will not be affected much for a simple scoring method, such as shape complementarity, but loss of the accuracy might be significant for more sophisticated scoring functions, such as van der Waals interactions and electrostatic energies. Moreover, some scoring energy terms, such as hydrophobic effects, desolvation energies and entropic effects, which have more complicated terms cannot be projected onto a grid. Therefore, much effort has recently been made to improve the accuracy of FFT-based docking algorithms by integrating more sophisticated scoring functions, such as atomic solvation parameters for ASPDock [41], stepwise force-field potentials for SDOCK [38] and knowledge-based potentials for PIPER [36] and ZDOCK 3.0 [33]. However, how to keep the original accuracy of scoring functions when mapping them onto grids and how to design the proper form of some scoring functions so that they can be projected onto grids are still two of the challenges and future research directions for exhaustive global search algorithms.

Similar issues also apply to local shape feature-matching algorithms because the energy terms of scoring functions also need to be mapped onto the molecular surface or a grid that is used for local shape feature matching. Although randomized search algorithms might not lose the accuracy of scoring functions because the proteins are normally represented at the atomic level during the search process, they might have the expensive computational efficiency because of the higher computational cost of sampling and scoring at the atomic level compared with the grid level. Therefore, how to improve the computational efficiency without loss of accuracy is a challenge in randomized search algorithms.

Inclusion of information about binding

Although an ab initio docking search is common because of the lack of information about protein–protein interactions, some proteins do have biological information about binding available from experimental data or a domain interaction database, such as KBDOCK [113]. There are two ways to use information about binding depending on the property of the information: restricting the sampling space and template-based docking.

Restricting the sampling space

This is a general way and also the most commonly used approach to incorporate the biological information in a docking process. It can utilize all types of information, from mutagenesis data and

TABLE 2

Features of search algorithms in protein–protein docking

Search algorithms	Exhaustive search	Global search	Local search	Rigid docking	Flexible docking	Molecular representation	Computational cost
FFT-based correlation	√	√		√		Grid based	Low
SFT-based search	√	√		√		Harmonic surface	Low
Direct global search	√	√	√	√		Grid based	Medium–high
Local shape matching		√	√	√		Grid or surface	Medium
Randomized search		√	√	√	√	Atom based	High

NMR constraints to evolutionary data, during the search process in initial docking algorithms or through the filtering step in post-docking approaches.

Given the nature of Fourier transform in Cartesian grid space, the search over the entire 3D translational space will be always performed during FFT-based search algorithms, and there is no efficient way to just search part of the space that is of interest or leave out some of the space that is not significant for binding. However, this is important when information about binding sites is already known. In that case, confining the search around the binding site according to the information will not only speed up the search process because of the smaller search space, but also increase the probability of finding correct binding modes because of fewer possible binding orientations. Some FFT-based docking algorithms, such as MolFit and ZDOCK, do have an option to include information about binding sites during or after searching [30–33,40,41,114], but the search would still be performed over the entire translational space and, thus, the computational cost is still not reduced. Although some efforts have been made to reduce the computational cost by approaches such as a partial scan in MolFit [115] or a two-step hybrid-resolution sampling procedure in F2Dock [37] and ZDOCK [116] in rotational space, how to include efficiently information about binding sites in the search process is still an important issue in the Cartesian FFT-based algorithms.

Information about the binding site can be included in spherical Fourier transform (SFT)-based algorithms by restraining the search angles for one or two of the proteins in rotational space [45,46], although the SFT-based algorithms suffer from the loss of accuracy in molecular representation because the radial functions for spherical harmonic surface fall off quickly after a certain distance from the origin. Therefore, docking of larger proteins is a challenge for SFT-based docking algorithms. The docking results can also depend on the initial relative orientation and intermolecular distance between two proteins if a spherical polar Fourier correlation is used to further speed up the search in both rotational and translational space, as used in HEX [46,47]. This will introduce an additional difficulty for setting up initial locations and orientations of the proteins.

Unlike in FFT-based docking algorithms, a local search around the binding site can be easily achieved in direct search, local shape feature-matching or randomized search approaches by restricting the matching areas or initial orientations and/or distances between proteins during the search or subsequent optimization process. A typical docking algorithm is HADDOCK [68,69], which was specially developed to utilize the biochemical or biophysical information from experimental data, such as NMR restraints, during the search process. Therefore, when information about binding is available, direct search, local shape feature-matching or randomized search algorithms might be one of the choices for such protein–protein docking cases. However, one limitation in direct search algorithms is its relatively expensive computational cost. It should also be kept in mind that both local shape feature-matching and randomized search algorithms do not perform an exhaustive search over every single point of 6D rotational and translational space. Therefore, more binding orientations might need to be sampled to cover as much binding space as possible for such algorithms.

Template-based docking

In addition to including the available biological information in a regular search process, 'docking' two protein structures by aligning them onto a homologous complex, so-called 'template-based docking', is another way to incorporate information about the binding interface and has recently received significant attention [117–123]. The basic idea is that, instead of constructing putative protein–protein binding structures through a regular docking procedure, the approach is to search a complex of two proteins that are homologous to the two proteins to be docked using information such as sequence similarity, evolutionary conservation and interface similarity [117,120]. Then, the complex can be constructed by superimposing two individual protein structures onto the homologous complex or directly modeling the complex structures using the homologous complex as a template. The 'docked' complex structure by template-based docking can be further refined with explicit consideration of protein flexibility by a post-docking approach or a randomized search algorithm. Despite encouraging success [117], the protocol is limited by the availability of the homologous protein–protein complexes and the reliability of the homologous complexes, because two pairs of homologous proteins might not necessarily bind at the same interface. With increasing amounts of information and numbers of structures for protein–protein interactions available, the template-based docking approach is expected to have an increasing role in the protein-docking field.

Consideration of protein flexibility

Protein flexibility is one of the challenging issues involved in both the sampling and scoring processes of protein–protein docking [14–18]. In FFT-based algorithms, the grid-based representations of two proteins cannot be changed during the search process. Therefore, both FFT-based and SFT-based algorithms perform rigid-body protein–protein docking. Only part of the protein conformational change can be implicitly considered by allowing a certain degree of penetration between proteins or softening of their molecular surfaces. The implicit consideration of protein flexibility, which is at the expense of the accuracy of molecular representation of proteins, depends on the size of the grid spacing for regular FFT-based algorithms [25] or the highest polynomial power in the spherical harmonic Fourier expansion for SFT-based approaches [45,46]. Therefore, consideration might be needed relating to the optimal grid spacing and the highest polynomial power in the spherical harmonic Fourier expansion in FFT-based and SFT-based docking algorithms, respectively. Moreover, given its inherent limitation, the implicit approach is not able to consider large conformational changes, such as backbone rearrangements and domain movements. Therefore, how to consider large conformational changes is another challenge when using FFT-based algorithms for protein–protein docking.

Similar to FFT-based algorithms, direct search and local shape feature-matching algorithms also suffer from large conformational changes in proteins because of their rigid-body docking nature and, therefore, have similar challenges and/or limitations. Possible solutions include ensemble docking, where several different conformations of proteins are used for multiple docking calculations [63,124,125], or post-docking approaches, where protein flexibility is explicitly considered during the optimization and/or refinement stage [95–104].

Theoretically, randomized search algorithms can handle any degree of protein flexibility from small side-chain fluctuations to large domain movements. First, their initial ligand-binding orientations are placed away from the receptor protein and, thus, are not very sensitive to the molecular shape. Then, from the starting binding orientation, protein flexibility can be explicitly considered by allowing side chains and/or backbones to change during the optimization process through the Monte Carlo search. Therefore, randomized search algorithms, such as RosettaDOCK [61–63], HADDOCK [68,69], ATTRACT [65–67] and ICM-DISCO [70,71], might be the first choice when large conformational changes occur upon binding. However, in this situation, challenges come from the expensive computation cost in evaluating different binding conformations at the atomic level and the increased sampling space resulting from the introduction of additional degrees of freedom for conformational sampling. The high number of degrees of freedom can also result in the generation of more false positives because of the limitations of current scoring functions. Although ensemble docking and information-driven sampling approaches have been well used to reduce the sampling space in randomized search algorithms [17], how to improve the computational efficiency in both scoring and sampling is still a challenge to consider for large protein conformational changes upon binding during the randomized search process. With the increasing computational power and the development of more sophisticated approaches, such as coarse-grain modeling [126–130], the randomized search algorithm is expected to have a more important role in flexible protein–protein docking.

Docking of homology-modeled structures

Another important issue in protein–protein docking is the docking of homology-modeled structures because the number of the proteins with experimentally determined 3D structures is limited compared with the number of proteins with known sequences. As shown in the last CAPRI rounds [79,80], more than half of the targets involved modeling of protein structures. Docking of homology-modeled structures is more challenging than docking of regular unbound structures because of the dependence of the modeled structures on the quality of the templates used for homology modeling.

There are two difficulties associated with the task. One is the selection of an appropriate template, especially when there are multiple structures in the PDB [10] that are homologous to the protein to be modeled. In this situation, priority can be given to those structures in protein–protein complexes, because the protein structure in a complex might have included the induced fit owing to binding with another protein structure; in addition, its binding interface might also be a potential site of binding because of the conservation of residues involved in protein–protein interactions, although the sequence similarity is still a major factor when choosing a template because it is a direct measurement of the quality of the templates for homology modeling.

The other difficulty is to consider the conformational change upon binding because the modeled structure might be different from the 'unbound' confirmation of the true protein because of the sequence difference, not to mention the bound conformation in the binding state. Although implicit consideration of protein flexibility has achieved many successes in docking homology-modeled structures [89,131], explicit consideration of protein conformational changes is often necessary because of the reduced accuracy of a modeled structure compared with an unbound structure. There are two ways of considering explicitly the protein flexibility in docking modeled structures. One is to use multiple conformations that are generated from one homology-modeling run using a single template or from multiple homology-modeling runs based on multiple templates if available, in the hope that the multiple modeled structures might include the conformations close to the one in the bound form [89]. Another way is to sample the protein conformations during the docking process, as in the randomized search algorithms such as RosettaDock [64] and ATTRACT [132]. The limitation of this method is its expensive computational cost resulting from the huge degrees of freedom and the strong dependence on the accuracy of scoring functions, as mentioned above. However, with the constraints relating to information about binding, sampling efficiency and accuracy could be considerably improved using information-driven protein-docking algorithms, such as HADDOCK [69]. The information about binding is also valuable for filtering the binding decoys from rigid-body docking algorithms and to guide the refinement process during the post-docking stage.

Evaluation of docking algorithms

Given the large number of protein–protein docking algorithms with different search strategies, one crucial issue is how to evaluate objectively a docking algorithm with a certain search strategy so as to test the performance of the approach, identify possible limitations and further improve the search algorithm based on the test results.

The docking benchmark

To evaluate a developed algorithm, a benchmark including known protein–protein complexes with certain features is necessary so that comparative evaluation of different algorithms can be performed on the same basis. First, the benchmark should be diverse to cover as many types of protein–protein interaction as possible so that the evaluation will not be biased towards certain complexes. Second, the complexes in the benchmark should be taken from experimentally determined high-resolution structures instead of molecular modeling to avoid the introduction of computational errors. Third, the benchmark should include both the bound and unbound structures of protein–protein complexes to reflect realistic conformational changes upon binding. Meeting these criteria, several well-prepared protein–protein docking benchmarks [133–138] have been published and widely used to evaluate docking algorithms in the community. In addition, several searchable databases of protein complexes, such as PIBASE [139], 3D-Complex [140], DOCKGROUND [141], SCOPPI [142], I2I [143] and PROTCOM [144], have been constructed for various systematic studies of protein–protein interactions, which would also provide valuable resources for the construction of new docking benchmarks. The benchmarks including both bound and unbound structures are a valuable resource for testing docking algorithms in a realistic binding environment where the conformations of proteins change upon binding.

However, challenges still exist with current benchmarks. Given the limitation of experimental data, some atoms, residues and even segments are often missing in many experimentally determined structures. Therefore, there might be an inconsistency between the bound and unbound structures of the same protein on the backbone and/or side chains. This will lead to a challenge in analyzing the test results. That is, it is unclear whether failure would be the result of the conformational change between the bound and unbound structures or those missing or extra residues and/or atoms if a docking algorithm fails to predict near-native binding modes. Ways for removing this kind of inconsistency include deleting extra residues and/or atoms based on a bound–unbound mapping or adding those missing residues and/or atoms using molecular modeling tools in the bound and/or unbound structures. Both ways suffer from loss of accuracy in the structures. The deletion also cuts the structural completeness of the protein, whereas the addition introduces some structural uncertainties because of the computational modeling. Therefore, how to prepare a diverse benchmark with consistent residues and atoms between bound and unbound structures is an important issue and possibly one of the future research directions in developing a new protein–protein docking benchmark.

The evaluation criteria

Another important issue in the evaluation of protein–protein docking algorithms is how to measure the performance of an algorithm. This includes two aspects: the first is the criteria used to measure the quality of binding modes; the other is to quantify the performance of a docking algorithm based on the measured qualities of the binding modes.

The quality of binding modes

Currently, there are two types of criterion that are widely used to measure the quality of a binding mode. One is the CAPRI criterion, which is based on the combination of three parameters [76–80]: the ligand root mean square deviation (L_{rmsd}) after the superimposition of the receptor proteins, the interface RMSD (I_{rmsd}) after the optimal superimposition of the backbone atoms between the native and predicted interfaces, and the percentage of predicted native contacts (f_{nat}). According to these three parameters, the accuracy of predictions can be divided into four categories of accuracies: high, medium, acceptable and incorrect (Table 3), where a prediction above acceptable accuracy can be defined as a hit [89,145]. The CAPRI criteria can also be adapted by only using a combination of two parameters, such as I_{rmsd} and f_{nat}, to define hits [33].

Another type of success criterion is based on a single RMSD parameter, such as L_{rmsd} or I_{rmsd}. In this category, the commonly used criteria to define a success or a hit may be $L_{rmsd} < 10$ Å as used in RosettaDock [61] and PIPER [36] or $I_{rmsd} < 2.5$ Å as adopted in ZDOCK [30,31], in which the RMSD cutoff for success can be adjusted depending on specific evaluation purpose [128] (Table 3). Of these two types of criterion, the CAPRI criteria are used mainly in the CAPRI experiments, whereas the single RMSD criterion is more common in the validation studies of developed docking or scoring algorithms. They are qualitatively similar and should be interchangeable in ranking the performances of different algorithms based on the same test set. The advantage of the single-ligand RMSD criterion is that its calculation does not involve superimposition between native and predicted binding interfaces and, thus, is easy to calculate. Its limitation is that it might miss some successful predictions if there is a significant conformational change in the region of the unbound ligand protein away from the binding interface. By contrast, the CAPRI criterion will normally not miss a successful prediction, but has a cost of more complicated computations. In general, the single-ligand RMSD criterion is acceptable for use in the validation of algorithms because, compared with the large test set, a single missout would not change the results statistically. However, the CAPRI criteria will be necessary for the CAPRI experiments because every single binding mode makes a difference for the participants who are only allowed to submit ten models for each target.

TABLE 3

Criteria to measure the quality of predictions in protein–protein docking

Accuracy category	Criteria
CAPRI criteria[a]	
High	$f_{nat} \geq 0.5$ and ($L_{rmsd} \leq 1.0$ or $I_{rmsd} \leq 1.0$)
Medium	$f_{nat} \geq 0.3$ and ($1.0 < L_{rmsd} \leq 5.0$ or $1.0 < I_{rmsd} \leq 2.0$)
Acceptable	$f_{nat} \geq 0.1$ and ($5.0 < L_{rmsd} \leq 10.0$ or $2.0 < I_{rmsd} \leq 4.0$)
Incorrect	$f_{nat} < 0.1$ or ($L_{rmsd} > 10.0$ and $I_{rmsd} > 4.0$)
Ligand RMSD criterion[b]	
Successful	$L_{rmsd} < 10$ Å
Incorrect	Otherwise
Interface RMSD criterion[b]	
Successful	$I_{rmsd} < 2.5$ (or 4.0) Å
Incorrect	Otherwise

[a] The criteria can be modified by using a combination of two of the three CAPRI parameters.

[b] The RMSD cutoff for success can be adjusted depending on specific applications.

The performance of docking algorithms

With the quality definition of binding modes, one is able to quantify the performance of a docking algorithm in binding mode prediction given a test set or benchmark. There are often two ways to measure the performance. One is the success rate, which is defined as the percentage of the targets whose native structures are successfully predicted for a given test set. During the calculation of success rates, the top-ranked binding mode may be considered, as used in protein–ligand docking [24]. Namely, the native structure of a target is successfully predicted if the top-ranked binding mode for the target is a hit. However, given the challenge in protein–protein docking, the top ten binding modes are often used when calculating success rates; that is, the native structure of a target is successfully predicted if any of the top ten binding modes is a hit. Similarly, one can calculate the success rates for any number of top considered binding modes (e.g. top five, top 100, and top 1000).

The other way to measure the performance of a docking algorithm is to calculate the number of hits in a certain number of top-ranked binding modes. This measurement is especially useful if a docking algorithm is used as a first step to generate initial binding decoys for post-docking approaches that depend on the availability of hits. The more hits found in the top binding modes generated by an initial docking program, the higher chance the post-docking approach might have to rank a hit at the top of the list.

Among these two parameters, the success rate is more of a general way to measure the performance of a docking algorithm

in binding mode predictions, whereas the number of hits is more commonly used in protein–protein docking as a general guideline for post-docking approaches to choose initial docking algorithms. Both measurements depend on the distribution of sampled binding modes in the conformational space. Therefore, their values might not be comparable among different docking algorithms if more than one top binding mode is considered during the calculation. One solution is to cluster the ranked binding modes for each of the tested docking algorithms with the same RMSD cutoff before measuring the performance. It should also be noted that the number of hits is target dependent and the degree of dependence is algorithm dependent. This might bring about a possibility that a docking algorithm obtains a larger average number of hits but has a lower success rate when a certain number of top binding modes are considered. Therefore, one should be careful when choosing an initial docking algorithm for a post-docking approach. A basic principle might be that the success rate is more important initially than the average number of hits. This principle might also apply to measuring the overall performance of a docking algorithm in general.

The CAPRI experiment

The protein–protein docking benchmarks provide an ideal environment to validate a docking algorithm because the experimentally determined structures are already known. The known structural information is able to give immediate feedback relating to the correction and improvement of the tested docking algorithm. However, this might bring some biases to the developed algorithm that is to predict correctly more native structures in a benchmark. It might affect the general applicability of the developed algorithm because of the imperfectness of the benchmark, as discussed above. Therefore, a blind prediction test of protein–protein interactions would be desired for a truly objective evaluation of developed docking algorithms. To meet this need, the CAPRI challenge [76–80] was launched in 2001 to serve as a valuable platform for evaluating docking and scoring protocols on realistic applications, and has had a significant role in promoting the development and/or improvement of docking and scoring algorithms since then.

Table 4 lists the top-performing groups in the last five CAPRI editions from 2001 to 2012. Several notable features can be observed from Table 4, which might indicate some of the trends in the development of docking and scoring algorithms. First, there are new docking and/or scoring algorithms coming in each new CAPRI edition in addition to some traditional docking programs, such as 3D-DOCK and MolFit [26,35]. Examples of these new algorithms include RosettaDock [61] in rounds 3–5, ZRANK [82] in rounds 6–12, and PIPER [36] and MDockPP [89] in rounds 13–19 (Table 4). Second, the composition of the top-performing groups and the rankings of some algorithms vary significantly in different CAPRI editions, suggesting the diversity of top-performing algorithms and the fast progress in the development and improvement of algorithms. Third, although randomized search algorithms such as ICM-DISCO, ATTRACT and HADDOCK, have performed well in the CAPRI challenge because of their explicit consideration of protein flexibility, many top-performing groups are based on FFT-based search strategies, such as 3D-DOCK, MolFit, DOT, ZDOCK, ClusPro and PIPPER (Table 4), indicating the robustness of FFT-based approaches. Last, the docking algorithms have evolved significantly to accommodate the emerging challenges in real applications, and have become a composite package of sophisticated approaches for scoring function, consideration of protein flexibility, post-docking processing and use of information about binding [76–80,146–155], which might suggest a future research direction in the development of new docking and/or scoring algorithms.

TABLE 4

List of top-performing groups in the previous CAPRI experiments

Algorithms	Group	Summary[f]
Rounds 1–2 (seven targets, 2001–2003)		
ICM-DISCO	Abagyan	3/2**/1***
SmoothDock	Camacho	3/2***
MolFit	Eisenstein	3/1***
3D-DOCK/MULTIDOCK	Sternberg	3/1***
DOT	Ten-Eyck	3/1**
Rounds 3–5 (nine targets, 2003–2005)		
ICM-DISCO	Abagyan	8/4**/2***
PatchDock/FlexDock	Wolfson	8/3**
ZDOCK	Weng	7/3**/3***
Modified 3D-DOCK, MultiDock	Bates	7/3**
RosettaDock	Baker[a]	6/2**/4***
Rounds 6–12 (seven targets, 2005–2007)		
ZDOCK, ZRANK	Weng	5/2(**)
HADDOCK	Bonvin	4/2(**)
MolFit	Eisenstein	3/1(***)
MolFit, 3D-Dock, RosettaDock, SMD refinement	Smith	3/2(**)
ClusPro	Vajda	3/2(**)
Rounds 13–19 (13 targets, 2007–2009)		
Cluspro, PIPER, SDU	Vajda	6/4***/2**
ATTRACT	Zacharias	6/4***/1**
MDockPP	Zou[b]	6/3***/2**
MolFit	Eisenstein[c]	6/3***/1**
PatchDock, FlexDock, FiberDock	Wolfson[c]	6/3***/1**
ZDOCK, ZRANK	Weng[d]	6/2***/2**
meta-PPISP, ZDOCK, ZRANK, CHARMM, HADDOCK	Zhou[d]	6/2***/2**
Rounds 20–27 (ten targets, 2010–2012)		
HADDOCK	Bonvin	9/1***/3**
SwarmDock + Markov-chain model	Bates	8/2**
Template-based docking	Vakser	7/1***
ClusPro 2.0/PIPPER	Vajda	6/2***/3**
pyDock	Fernandez-Recio[e]	6/1***/3**
ClusPro 2.0 + SDU	Shen[e]	6/1***/3**

[a] The Baker group did not submit models for one target (T08).
[b] The Zou group did not participate in the rounds 13 and 14 for three targets (T29, T30 and T31).
[c] The Eisenstein and Wolfson groups obtained the same performance, ranked #4.
[d] The Weng and Zhou groups gave the same performance, ranked #5.
[e] The Fernandez-Recio and Shen groups achieved the same performance, ranked #5.
[f] The summary column lists the total number of acceptable predictions, followed by the number of predictions of medium (**) and high (***) accuracy, respectively.

The most important and attractive aspect of CAPRI might be its blind prediction feature that guarantees the objective evaluation of the platform. However, there are also challenges in CAPRI. One comes from the fact that participants are free to use any biological information, such as mutation data and evolutionary analyses, in the CAPRI experiments. Although the biological information significantly improves the success rate of CAPRI predictions [76–79], it also poses two difficulties with the algorithms for CAPRI because the use of biological information is arbitrary and strongly subjective depending on the CAPRI participants. On the one hand, it

TABLE 5

List of available protein–protein docking servers

Server	Algorithm	Website
ClusPro	FFT based	http://cluspro.bu.edu/
GRAMM-X	FFT based	http://vakser.bioinformatics.ku.edu/resources/gramm/grammx/
ZDOCK	FFT based	http://zdock.umassmed.edu/
3D-Garden	FFT based	http://www.sbg.bio.ic.ac.uk/~3dgarden/
HEX Server	SFT based	http://hexserver.loria.fr/
PatchDock	Geometric hashing	http://bioinfo3d.cs.tau.ac.il/PatchDock/
HADDOCK	Randomized search	http://haddock.science.uu.nl/services/HADDOCK/
RosettaDock	Randomized search	http://rosettadock.graylab.jhu.edu/
SwarmDock	Randomized search	http://bmm.cancerresearchuk.org/SwarmDock/
DOCK-PIE	Post-docking	http://clsb.ices.utexas.edu/web/dock.html
FiberDock	Post-docking	http://bioinfo3d.cs.tau.ac.il/FiberDock/
FireDock	Post-docking	http://bioinfo3d.cs.tau.ac.il/FireDock/
pyDockWeb	Post-docking	http://life.bsc.es/servlet/pydock/home/

is unclear whether a success should be attributed to the original docking algorithm or the filtering step with the biological information. This might sometimes hide some of the limitations in the original docking and/or scoring algorithm, which is not desirable for the improvement of the algorithm. On the other hand, the results might not be reproducible because of the subjectivity of using biological information. Solutions for these difficulties include introducing additional categories, such as the web server challenge [79], into the CAPRI experiments. For automatic prediction purposes, many docking algorithms have already developed their own web server versions, such as ClusPro [156], HADDOCK [157], RosettaDock [158], GRAMM-X [159], 3D-Garden [160], FiberDock [161], FireDock [161], HEX server [162], SwarmDock [163], ZDOCK server [164], PatchDock [53] and pyDockWeb [165] (Table 5), which would help overcome the arbitrary effect from human interference in docking calculations.

Another challenge of CAPRI is its limited number of targets because only newly determined protein–protein complex structures can be used for the blind prediction experiment. The challenge could be relieved with the fast development of proteomics projects and the CAPRI becoming widely known to the community of experimental structural biologists. As a whole, despite some few challenges, the CAPRI experiment is still, and will continue to be for some time, the most widely used platform for the objective evaluation of developed docking and scoring algorithms in the research community.

Concluding remarks

Although molecular docking has been studied for more than three decades, protein–protein docking has only developed significantly over the past two decades, and now has an increasing role in predicting protein–protein interactions, revealing the interacting mechanism between proteins and identifying the 'hot-spot' residues for drug discovery. Given the lack of information about binding sites and the additional degrees of freedom in proteins, search algorithms in protein–protein docking are often different from those in protein–ligand docking; thus, a variety of search algorithms have been developed for protein–protein docking over the past two decades. These algorithms, which differ in methodology, search range, speed and molecular representation, have all achieved successes in some categories of docking, but have their own challenges and/or limitations that leave room for future improvement. Given the abundance of existing search algorithms and their successes, future strategies might include compounding current algorithms in addition to developing new search methods and improving existing algorithms by optimizing the interplay effects between the search and other factors, such as scoring and flexibility in docking. Given various docking and post-docking algorithms, how to evaluate their performances remains an important topic in validating developed algorithms, detecting potential limitations and improving them further. In this aspect, challenges and trends for future research might include the construction of more sophisticated docking benchmarks, the proper use of evaluation criteria and the test of such developments using realistic docking experiments, such as CAPRI.

Acknowledgments

Computations for this work were performed on the HPC resources at the University of Missouri Bioinformatics Consortium (UMBC). The support of the UMBC is greatly appreciated.

References

1 Waksman, G. (2005) *Proteomics and Protein–Protein Interactions: Biology, Chemistry, Bioinformatics, and Drug Design*. Springer

2 Kann, M.G. (2007) Protein interactions and disease: computational approaches to uncover the etiology of diseases. *Brief. Bioinform.* 8, 333–346

3 Arkin, M.R. and Wells, J.A. (2004) Small-molecule inhibitors of protein–protein interactions: progressing towards the dream. *Nat. Rev. Drug Discov.* 3, 301–317

4 Fuller, J.C. *et al.* (2009) Predicting druggable binding sites at the protein–protein interface. *Drug Discov. Today* 14, 155–161

5 Grosdidier, S. and Fernandez-Recio, J. (2012) Protein–protein docking and hot-spot prediction for drug discovery. *Curr. Pharm. Des.* 18, 4607–4618
6 Bienstock, R.J. (2012) Computational drug design targeting protein–protein interactions. *Curr. Pharm. Des.* 18, 1240–1254
7 Schreiber, G. and Fleishman, S.J. (2013) Computational design of protein–protein interactions. *Curr. Opin. Struct. Biol.* 23, 903–910
8 Huang, S.-Y. *et al.* (2009) Molecular modeling of the heterodimer of human CFTR's nucleotide-binding domains using a protein-protein docking approach. *J. Mol. Graph. Model.* 27, 822–828
9 Zhang, G. *et al.* (2010) Ion sensing in the RCK1 domain of BK channels. *Proc. Natl. Acad. Sci. U. S. A.* 107, 18700–18705
10 Berman, H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.* 28, 235–242
11 Janin, J. (2010) Protein–protein docking tested in blind predictions: the CAPRI experiment. *Mol. Biosyst.* 6, 2351–2362
12 Wodak, S.J. and Janin, J. (1978) Computer analysis of protein–protein interaction. *J. Mol. Biol.* 124, 323–342
13 Smith, G.R. and Sternberg, M.J. (2002) Prediction of protein–protein interactions by docking methods. *Curr. Opin. Struct. Biol.* 12, 28–35
14 Gray, J.J. (2006) High-resolution protein–protein docking. *Curr. Opin. Struct. Biol.* 16, 183–193
15 Bonvin, A.M. (2006) Flexible protein–protein docking. *Curr. Opin. Struct. Biol.* 16, 194–200
16 Andrusier, N. *et al.* (2008) Principles of flexible protein–protein docking. *Proteins* 73, 271–289
17 Zacharias, M. (2010) Accounting for conformational changes during protein–protein docking. *Curr. Opin. Struct. Biol.* 20, 180–186
18 Ritchie, D.W. (2008) Recent progress and future directions in protein–protein docking. *Curr. Protein Pept. Sci.* 9, 1–15
19 Vajda, S. and Kozakov, D. (2009) Convergence and combination of methods in protein–protein docking. *Curr. Opin. Struct. Biol.* 19, 164–170
20 Vajda, S. *et al.* (2013) Sampling and scoring: a marriage made in heaven. *Proteins* 81, 1874–1884
21 Moal, I.H. *et al.* (2013) Scoring functions for protein–protein interactions. *Curr. Opin. Struct. Biol.* 23, 862–867
22 Vakser, I.A. (2013) Low-resolution structural modeling of protein interactome. *Curr. Opin. Struct. Biol.* 23, 198–205
23 Huang, S-Y. and Zou, X. (2010) Advances and challenges in protein–ligand docking. *Int. J. Mol. Sci.* 11, 3016–3034
24 Huang, S-Y. *et al.* (2010) Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Phys. Chem. Chem. Phys.* 12, 12899–12908
25 Katchalski-Katzir, E. *et al.* (1992) Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. U. S. A.* 89, 2195–2199
26 Gabb, H.A. *et al.* (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. *J. Mol. Biol.* 272, 106–120
27 Vakser, I.A. (1997) Evaluation of GRAMM low-resolution docking methodology on the hemagglutinin-antibody complex. *Proteins* 29 (Suppl. 1), 226–230
28 Mandell, J.G. *et al.* (2001) Protein docking using continuum electrostatics and geometric fit. *Protein Eng.* 14, 105–113
29 Roberts, V.A. *et al.* (2013) DOT2: macromolecular docking with improved biophysical models. *J. Comput. Chem.* 34, 1743–1758
30 Chen, R. and Weng, Z.P. (2002) Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins* 47, 281–294
31 Chen, R. and Weng, Z. (2003) A novel shape complementarity scoring function for protein–protein docking. *Proteins* 51, 397–408
32 Chen, R. *et al.* (2003) ZDOCK: an initial-stage protein docking algorithm. *Proteins* 52, 80–87
33 Mintseris, J. *et al.* (2007) Integrating statistical pair potentials into protein complex prediction. *Proteins* 69, 511–520
34 Pierce, B.G. *et al.* (2011) Accelerating protein docking in ZDOCK using an advanced 3D convolution library. *PLoS ONE* 6, e24657
35 Heifetz, A. *et al.* (2002) Electrostatics in protein–protein docking. *Protein Sci.* 11, 571–587
36 Kozakov, D. *et al.* (2006) PIPER: an FFT-based protein docking program with pairwise potentials. *Proteins* 65, 392–406
37 Bajaj, C. *et al.* (2011) F2Dock: fast Fourier protein–protein docking. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 45–58
38 Zhang, C. and Lai, L. (2011) SDOCK: a global protein–protein docking program using stepwise force-field potentials. *J. Comput. Chem.* 32, 2598–2612
39 Pons, C. *et al.* (2012) Cell-Dock: high-performance protein–protein docking. *Bioinformatics* 28, 2394–2396
40 Ben-Zeev, E. and Eisenstein, M. (2003) Weighted geometric docking: incorporating external information in the rotation-translation scan. *Proteins* 52, 24–27
41 Li, L. *et al.* (2011) ASPDock: protein–protein docking algorithm using atomic solvation parameters model. *BMC Bioinformatics* 12, 36
42 Berchanski, A. *et al.* (2004) Hydrophobic complementarity in protein–protein docking. *Proteins* 56, 130–142
43 Berchanski, A. *et al.* (2005) Modeling oligomers with Cn or Dn symmetry: application to CAPRI target 10. *Proteins* 60, 202–206
44 Pierce, B. *et al.* (2005) M-ZDOCK: a grid-based approach for Cn symmetric multimer docking. *Bioinformatics* 21, 1472–1478
45 Garzon, J.I. *et al.* (2009) FRODOCK: a new approach for fast rotational protein–protein docking. *Bioinformatics* 25, 2544–2551
46 Ritchie, D.W. and Kemp, G.J. (2000) Protein docking using spherical polar Fourier correlations. *Proteins* 39, 178–194
47 Ritchie, D.W. *et al.* (2008) Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics* 24, 1865–1873
48 Ritchie, D.W. and Venkatraman, V. (2010) Ultra-fast FFT protein docking on graphics processors. *Bioinformatics* 26, 2398–2405
49 Jiang, F. and Kim, S.H. (1991) 'Soft docking': matching of molecular surface cubes. *J. Mol. Biol.* 219, 79–102
50 Palma, P.N. *et al.* (2000) BiGGER: a new (soft) docking algorithm for predicting protein interactions. *Proteins* 39, 372–384
51 Terashi, G. *et al.* (2007) The SKE-DOCK server and human teams based on a combined method of shape complementarity and free energy estimation. *Proteins* 69, 866–872
52 Kuntz, I.D. *et al.* (1982) A geometric approach to macromolecule–ligand interactions. *J. Mol. Biol.* 161, 269–288
53 Duhovny, D. *et al.* (2002) Efficient unbound docking of rigid molecules. *Lect. Notes Comp. Sci.* 2452, 185–200
54 Schneidman-Duhovny, D. *et al.* (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Res.* 33, W363–W367
55 Venkatraman, V. *et al.* (2009) Protein–protein docking using region-based 3D Zernike descriptors. *BMC Bioinformatics* 10, 407
56 Esquivel-Rodriguez, J. *et al.* (2012) Multi-LZerD: multiple protein docking for asymmetric complexes. *Proteins* 80, 1818–1833
57 Gardiner, E.J. *et al.* (2001) Protein docking using a genetic algorithm. *Proteins* 44, 44–56
58 Shoichet, B.K. and Kuntz, I.D. (1991) Protein docking and complementarity. *J. Mol. Biol.* 221, 327–346
59 Shoichet, B.K. and Kuntz, I.D. (1996) Predicting the structure of protein complexes: a step in the right direction. *Chem. Biol.* 3, 151–156
60 Lorber, D.M. *et al.* (2002) Protein–protein docking with multiple residue conformations and residue substitutions. *Protein Sci.* 11, 1393–1408
61 Gray, J.J. *et al.* (2003) Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J. Mol. Biol.* 331, 281–299
62 Wang, C. *et al.* (2007) Protein–protein docking with backbone flexibility. *J. Mol. Biol.* 373, 503–519
63 Chaudhury, S. and Gray, J.J. (2008) Conformer selection and induced fit in flexible backbone protein–protein docking using computational and NMR ensembles. *J. Mol. Biol.* 381, 1068–1087
64 Chaudhury, S. *et al.* (2011) Benchmarking and analysis of protein docking performance in Rosetta v3.2. *PLoS ONE* 6, e22477
65 Zacharias, M. (2003) Protein–protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Sci.* 12, 1271–1282
66 Zacharias, M. (2005) ATTRACT: protein–protein docking in CAPRI using a reduced protein model. *Proteins* 60, 252–256
67 Bastard, K. *et al.* (2006) Accounting for loop flexibility during protein–protein docking. *Proteins* 62, 956–969
68 Dominguez, C. *et al.* (2003) HADDOCK: a protein–protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* 125, 1731–1737
69 de Vries, S.J. *et al.* (2007) HADDOCK versus HADDOCK: new features and performance of HADDOCK2.0 on the CAPRI targets. *Proteins* 69, 726–733
70 Abagyan, R. *et al.* (1994) ICM: a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. *J. Comput. Chem.* 15, 488–506
71 Fernandez-Recio, J. *et al.* (2003) ICM-DISCO docking by global energy optimization with fully flexible side-chains. *Proteins* 52, 113–117
72 Moal, I.H. and Bates, P.A. (2010) SwarmDock and the use of normal modes in protein–protein docking. *Int. J. Mol. Sci.* 11, 3623–3648
73 Morris, G.M. *et al.* (1998) Automated docking using a Lamarckian genetic algorithm and empirical binding free energy function. *J. Comput. Chem.* 19, 1639–1662

74 Andre, I. *et al.* (2008) Emergence of symmetry in homooligomeric biological assemblies. *Proc. Natl. Acad. Sci. U. S. A.* 105, 16148–16152
75 Saladin, A. *et al.* (2009) PTools: an opensource molecular docking library. *BMC Struct. Biol.* 9, 27
76 Janin, J. *et al.* (2003) CAPRI: a critical assessment of predicted interactions. *Proteins* 52, 2–9
77 Mendez, R. *et al.* (2005) Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. *Proteins* 60, 150–169
78 Lensink, M.F. *et al.* (2007) Docking and scoring protein complexes: CAPRI 3rd edition. *Proteins* 69, 704–718
79 Lensink, M.F. and Wodak, S.J. (2010) Blind predictions of protein interfaces by docking calculations in CAPRI. *Proteins* 78, 3085–3095
80 Lensink, M.F. and Wodak, S.J. (2013) Docking, scoring and affinity prediction in CAPRI. *Proteins* 81, 2082–2095
81 Moont, G. *et al.* (1999) Use of pair potentials across protein interfaces in screening predicted docked complexes. *Proteins* 35, 364–373
82 Pierce, B. and Weng, Z. (2007) ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins* 67, 1078–1086
83 Cheng, T.M. *et al.* (2007) pyDock: electrostatics and desolvation for effective scoring of rigid-body protein–protein docking. *Proteins* 68, 503–515
84 Liang, S. *et al.* (2007) A simple reference state makes a significant improvement in near-native selections from structurally refined docking decoys. *Proteins* 69, 244–253
85 Chuang, G.Y. *et al.* (2008) DARS (Decoys As the Reference State) potentials for protein–protein docking. *Biophys. J.* 95, 4217–4227
86 Liu, S. and Vakser, I.A. (2011) DECK: distance and environment-dependent, coarse-grained, knowledge-based potentials for protein–protein docking. *BMC Bioinformatics* 12, 280
87 Pons, C. *et al.* (2011) Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein–protein docking. *J. Chem. Inf. Model.* 51, 370–377
88 Ravikant, D.V. and Elber, R. (2010) PIE-efficient filters and coarse grained potentials for unbound protein–protein docking. *Proteins* 78, 400–419
89 Huang, S-Y. and Zou, X. (2010) MDockPP: a hierarchical approach for protein–protein docking and its application to CAPRI rounds 15–19. *Proteins* 78, 3096–3103
90 Huang, S.-Y. and Zou, X. (2008) An iterative knowledge-based scoring function for protein-protein recognition. *Proteins* 72, 557–579
91 Vreven, T. *et al.* (2011) Integrating atom-based and residue-based scoring functions for protein–protein docking. *Protein Sci.* 20, 1576–1586
92 Schneider, S. and Zacharias, M. (2012) Scoring optimisation of unbound protein–protein docking including protein binding site predictions. *J. Mol. Recognit.* 25, 15–23
93 Zacharias, M. (2013) Combining coarse-grained nonbonded and atomistic bonded interactions for protein modeling. *Proteins* 81, 81–92
94 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
95 Jackson, R.M. *et al.* (1998) Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. *J. Mol. Biol.* 276, 265–285
96 Camacho, C.J. and Gatchell, D.W. (2003) Successful discrimination of protein interactions. *Proteins* 52, 92–97
97 Li, L. *et al.* (2003) RDOCK: refinement of rigid-body protein docking predictions. *Proteins* 53, 693–707
98 Andrusier, N. *et al.* (2007) FireDock: fast interaction refinement in molecular docking. *Proteins* 69, 139–159
99 Mashiach, E. *et al.* (2010) FiberDock: flexible induced-fit backbone refinement in molecular docking. *Proteins* 78, 1503–1519
100 Venkatraman, V. and Ritchie, D.W. (2012) Flexible protein docking refinement using pose-dependent normal mode analysis. *Proteins* 80, 2262–2274
101 Krol, M. *et al.* (2007) Implicit flexibility in protein docking: cross-docking and local refinement. *Proteins* 69, 750–757
102 Lorenzen, S. and Zhang, Y. (2007) Monte Carlo refinement of rigid-body protein docking structures with backbone displacement and side-chain optimization. *Protein Sci.* 16, 2716–2725
103 Liang, S. *et al.* (2009) Refining near-native protein–protein docking decoys by local re-sampling and energy minimization. *Proteins* 76, 309–316
104 Viswanath, S. *et al.* (2013) Improving ranking of models for protein complexes with side chain modeling and atomic potentials. *Proteins* 81, 592–606
105 Shen, Y. *et al.* (2008) Protein docking by the underestimation of free energy funnels in the space of encounter complexes. *PLoS Comput. Biol.* 4, e1000191
106 Omori, S. and Kitao, A. (2013) CyClus: a fast, comprehensive cylindrical interface approximation clustering/reranking method for rigid-body protein–protein docking decoys. *Proteins* 81, 1005–1016
107 Oliva, R. *et al.* (2013) Ranking multiple docking solutions based on the conservation of inter-residue contacts. *Proteins* 81, 1571–1584
108 Qin, S. and Zhou, H.X. (2007) A holistic approach to protein docking. *Proteins* 69, 743–749
109 Lorenzen, S. and Zhang, Y. (2007) Identification of near-native structures by clustering protein docking conformations. *Proteins* 68, 187–194
110 Kozakov, D. *et al.* (2008) Discrimination of near-native structures in protein–protein docking by testing the stability of local minima. *Proteins* 72, 993–1004
111 Kowalsman, N. and Eisenstein, M. (2009) Combining interface core and whole interface descriptors in postscan processing of protein–protein docking models. *Proteins* 77, 297–318
112 Kowalsman, N. and Eisenstein, M. (2007) Inherent limitations in protein–protein docking procedures. *Bioinformatics* 23, 421–426
113 Ghoorah, A.W. *et al.* (2014) KBDOCK 2013: a spatial classification of 3D protein domain family interactions. *Nucleic Acids Res.* http://dx.doi.org/10.1093/nar/gkt1199
114 Li, L. *et al.* (2013) How to use not-always-reliable binding site information in protein–protein docking prediction. *PLoS ONE* 8, e75936
115 Ben-Zeev, E. *et al.* (2003) Prediction of the unknown: inspiring experience with the CAPRI experiment. *Proteins* 52, 41–46
116 Vreven, T. *et al.* (2013) Exploring angular distance in protein–protein docking algorithms. *PLoS ONE* 8, e56645
117 Szilagyi, A. and Zhang, Y. (2014) Template-based structure modeling of protein–protein interactions. *Curr. Opin. Struct. Biol.* 24, 10–23
118 Kundrotas, P.J. *et al.* (2012) Templates are available to model nearly all complexes of structurally characterized proteins. *Proc. Natl. Acad. Sci. U. S. A.* 109, 9438–9441
119 Mukherjee, S. and Zhang, Y. (2011) Protein–protein complex structure predictions by multimeric threading and template recombination. *Structure* 19, 955–966
120 Zhang, Q.C. *et al.* (2012) Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature* 490, 556–560
121 Guerler, A. *et al.* (2013) Mapping monomeric threading to protein–protein structure prediction. *J. Chem. Inf. Model.* 53, 717–725
122 Tuncbag, N. *et al.* (2011) Predicting protein–protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM. *Nat. Protoc.* 6, 1341–1354
123 Ghoorah, A.W. *et al.* (2011) Spatial clustering of protein binding sites for template based protein docking. *Bioinformatics* 27, 2820–2827
124 Huang, S-Y. and Zou, X. (2007) Ensemble docking of multiple protein structures: considering protein structural variations in molecular docking. *Proteins* 66, 399–421
125 Trellet, M. *et al.* (2013) A unified conformational selection and induced fit approach to protein–peptide docking. *PLoS ONE* 8, e58769
126 Baaden, M. and Marrink, S.J. (2013) Coarse-grain modelling of protein–protein interactions. *Curr. Opin. Struct. Biol.* 23, 878–886
127 Basdevant, N. *et al.* (2013) Modeling protein–protein recognition in solution using the coarse-grained force field SCORPION. *J. Chem. Theory. Comput.* 9, 803–813
128 Emperador, A. *et al.* (2013) Efficient relaxation of protein–protein interfaces by discrete molecular dynamics simulations. *J. Chem. Theory. Comput.* 9, 1222–1229
129 Solernou, A. and Fernandez-Recio, J. (2011) pyDockCG: new coarse-grained potential for protein–protein docking. *J. Phys. Chem. B* 115, 6032–6039
130 de Vries, S.J. and Zacharias, M. (2012) ATTRACT-EM: a new method for the computational assembly of large molecular machines using Cryo-EM maps. *PLoS ONE* 7, e49733
131 Vreven, T. *et al.* (2013) Performance of ZDOCK in CAPRI rounds 20–26. *Proteins* 81, 2175–2182
132 de Vries, S. and Zacharias, M. (2013) Flexible docking and refinement with a coarse-grained protein model using ATTRACT. *Proteins* 81, 2167–2174
133 Chen, R. *et al.* (2003) A protein–protein docking benchmark. *Proteins* 52, 88–91
134 Mintseris, J. *et al.* (2005) Protein–protein docking benchmark 2.0: an update. *Proteins* 60, 214–216
135 Hwang, H. *et al.* (2008) Protein–protein docking benchmark version 3.0. *Proteins* 73, 705–709
136 Hwang, H. *et al.* (2010) Protein–protein docking benchmark version 4.0. *Proteins* 78, 3111–3114
137 Kastritis, P. *et al.* (2011) A structure-based benchmark for protein–protein binding affinity. *Protein Sci.* 20, 482–491
138 Gao, Y. *et al.* (2007) DOCKGROUND system of databases for protein recognition studies: unbound structures for docking. *Proteins* 69, 845–851
139 Davis, F.P. and Sali, A. (2005) PIBASE: a comprehensive database of structurally defined protein interfaces. *Bioinformatics* 21, 1901–1907
140 Levy, E.D. *et al.* (2006) 3D complex: a structural classification of protein complexes. *PLoS Comput. Biol.* 2, e155
141 Douguet, D. *et al.* (2006) DOCKGROUND resource for studying protein–protein interfaces. *Bioinformatics* 22, 2612–2618

142 Winter, C. *et al.* (2006) SCOPPI: a structural classification of protein–protein interfaces. *Nucleic Acids Res.* 34, D310–D314

143 Mintz, S. *et al.* (2005) Generation and analysis of a protein–protein interface data set with similar chemical and spatial patterns of interactions. *Proteins* 61, 6–20

144 Kundrotas, P.J. and Alexov, E. (2007) PROTCOM: searchable database of protein complexes enhanced with domain–domain structures. *Nucleic Acids Res.* 35, D575–D579

145 Moal, I.H. *et al.* (2013) The scoring of poses in protein–protein docking: current capabilities and future directions. *BMC Bioinformatics* 14, 286

146 Wiehe, K. *et al.* (2007) The performance of ZDOCK and ZRANK in rounds 6-11 of CAPRI. *Proteins* 69, 719–725

147 Kozakov, D. *et al.* (2010) Achieving reliability and high accuracy in automated protein docking: ClusPro, PIPER, SDU, and stability analysis in CAPRI rounds 13–19. *Proteins* 78, 3124–3130

148 Mashiach, E. *et al.* (2010) An integrated suite of fast docking algorithms. *Proteins* 78, 3197–3204

149 Rodrigues, J.P. *et al.* (2013) Defining the limits of homology modeling in information-driven protein docking. *Proteins* 81, 2119–2128

150 Torchala, M. *et al.* (2013) A Markov-chain model description of binding funnels to enhance the ranking of docked solutions. *Proteins* 81, 2143–2149

151 Kundrotas, P.J. and Vakser, I.A. (2013) Global and local structural similarity in protein–protein complexes: implications for template-based docking. *Proteins* 81, 2137–2142

152 Kozakov, D. *et al.* (2013) How good is automated protein docking? *Proteins* 81, 2159–2166

153 Pallara, C. *et al.* (2013) Expanding the frontiers of protein–protein modeling: from docking and scoring to binding affinity predictions and other challenges. *Proteins* 81, 2192–2200

154 Shen, Y. (2013) Improved flexible refinement of protein docking in CAPRI rounds 22–27. *Proteins* 81, 2129–2136

155 Huang, S.-Y. *et al.* (2013) Inclusion of the orientational entropic effect and low-resolution experimental information for protein-protein docking in CAPRI. *Proteins* 81, 2183–2191

156 Comeau, S.R. *et al.* (2004) ClusPro: a fully automated algorithm for protein–protein docking. *Nucleic Acids Res.* 32, W96–W99

157 de Vries, S.J. *et al.* (2010) The HADDOCK web server for data-driven biomolecular docking. *Nat. Protoc.* 5, 883–897

158 Lyskov, S. and Gray, J.J. (2008) The RosettaDock server for local protein–protein docking. *Nucleic Acids Res.* 36, W233–W238

159 Tovchigrechko, A. and Vakser, I.A. (2006) GRAMM-X public web server for protein–protein docking. *Nucleic Acids Res.* 34, W310–W314

160 Lesk, V.I. and Sternberg, M.J. (2008) 3D-Garden: a system for modelling protein–protein complexes based on conformational refinement of ensembles generated with the marching cubes algorithm. *Bioinformatics* 24, 1137–1144

161 Mashiach, E. *et al.* (2010) FiberDock: a web server for flexible induced-fit backbone refinement in molecular docking. *Nucleic Acids Res.* 38, W457–W461

162 Macindoe, G. *et al.* (2010) HexServer: an FFT-based protein docking server powered by graphics processors. *Nucleic Acids Res.* 38, W445–W449

163 Torchala, M. *et al.* (2013) SwarmDock: a server for flexible protein–protein docking. *Bioinformatics* 29, 807–809

164 Pierce, B.G. *et al.* (2014) ZDOCK server: interactive docking prediction of protein-protein complexes and symmetric multimers. *Bioinformatics* http://dx.doi.org/10.1093/bioinformatics/btu097

165 Jimenez-Garcia, B. *et al.* (2013) pyDockWEB: a web server for rigid-body protein–protein docking using electrostatics and desolvation scoring. *Bioinformatics* 29, 1698–1699