Bioinformatics, 32(16), 2016, 2444–2450
doi: 10.1093/bioinformatics/btw184
Advance Access Publication Date: 8 April 2016
Original Paper

Structural Bioinformatics

Memdock: an α-helical membrane protein docking algorithm

Naama Hurwitz[1,*], Dina Schneidman-Duhovny[2] and Haim J. Wolfson[1,*]

[1]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel and [2]Department of Bioengineering and Therapeutic Sciences and Department of Pharmaceutical Chemistry, University of California, San Francisco, San Francisco, CA, USA

*To whom correspondence should be addressed.

Associate Editor: Anna Tramontano

Received on December 29, 2015; revised on March 17, 2016; accepted on April 3, 2016

Abstract

Motivation: A wide range of fundamental biological processes are mediated by membrane proteins. Despite their large number and importance, less than 1% of all 3D protein structures deposited in the Protein Data Bank are of membrane proteins. This is mainly due to the challenges of crystallizing such proteins or performing NMR spectroscopy analyses. All the more so, there is only a small number of membrane protein–protein complexes with known structure. Therefore, developing computational tools for docking membrane proteins is crucial. Numerous methods for docking globular proteins exist, however few have been developed especially for membrane proteins and designed to address docking within the lipid bilayer environment.

Results: We present a novel algorithm, Memdock, for docking α-helical membrane proteins which takes into consideration the lipid bilayer environment for docking as well as for refining and ranking the docking candidates. We show that our algorithm improves both the docking accuracy and the candidates ranking compared to a standard protein–protein docking algorithm.

Availability and Implementation: http://bioinfo3d.cs.tau.ac.il/Memdock/

Contacts: namih@tau.ac.il or wolfson@tau.ac.il

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Analysis of the complete genomic sequences for several organisms indicates that 30% of all genes code for membrane proteins (Jones, 1998; Wallin and von Heijne, 1998). Due to their central role in a wide range of fundamental biological processes, membrane proteins constitute around 60% of approved drug targets (Yildirim *et al.*, 2007). However, despite their large number and importance and even though there is an increasing number of solved membrane protein structures, they are still under-represented in the Protein Data Bank (PDB)(Berman *et al.*, 2000). In July 2015 only 1722 PDB entries were of membrane proteins (http://blanco.biomol.uci.edu/mpstruc/). This is owing to the difficulties involved in expressing, purifying and crystallizing these proteins (Carpemter *et al.*, 2008). In light of this deficit of empirical information, it is particularly important to develop efficient computational methods for predicting the structure, interactions and docking of membrane proteins.

Docking algorithms aim to predict the atomic resolution structure of a protein–protein complex given the structure of the separate proteins. Binding of two proteins to form a complex often results in conformational changes of the proteins. Therefore, there is a distinction between docking of ‘bound’ and ‘unbound’ protein structures. The conformation of the ‘bound’ structures is identical to their conformation in the complex, whereas the ‘unbound’ structure conformations are either the native conformation of the separate proteins in solution or their conformation in other complexes. Docking ‘unbound’ structures is a significantly more difficult task, since, as mentioned above, the ‘unbound’ proteins might undergo conformational changes upon binding (Janin, 2010).

The pipeline of docking algorithms usually includes three steps: (i) rigid docking; (ii) conformational refinement and (iii) docking candidates re-ranking by an energy score (Halperin *et al.*, 2002).

Numerous methods exist for the first step of rigid protein docking (Janin, 2010, Sousa *et al.*, 2013). These include Fast Fourier Transform based methods (ZDOCK, Chen *et al.*, 2003; GRAMM-X, Tovchigrechko and Vakser, 2005; Kozakov *et al.*, 2006), shape complementarity search by Geometric Hashing (PatchDock, Duhovny *et al.*, 2002; SymmDock, Schneidman-Duhovny *et al.*, 2005), Monte-Carlo based algorithms (RosettaDock, Gray *et al.*, 2003; ATTRACT, Zacharias, 2005) or information driven methods (e.g. HADDOCK, Dominguez *et al.*, 2003).

In the first step, of rigid-body docking, minor steric clashes between the receptor and the ligand structures are penalized liberally, thus tolerating conformational inaccuracies, which usually do not affect the protein backbone, but could have a significant impact on the side chain conformations. Consequently, the calculated energy value of a docking candidate could be high even if it is a near native one. Thus, it is necessary in the next step of refinement to remove steric clashes from the structures by exploring the conformational space of the side chains, while allowing minor movements in the backbone, as well (Chaudhury *et al.*, 2007; Fitzjohn and Bates, 2003; Krol *et al.*, 2007; Lindahl and Delarue, 2005; Mashiach *et al.*, 2009; May and Zacharias, 2007; Wang *et al.*, 2007).

All refinement methods use energy based scoring functions for re-ranking the candidate complexes in the last step.

Membrane proteins have special properties which are determined by their hydrophobic environment. Since the membrane imposes strong folding constraints, the polar groups of the polypeptide backbone of the membrane protein participate in intra-molecular hydrogen bonds. This is typically accomplished through two structural motifs: the membrane-spanning alpha-helix bundle and the beta-barrel (White and Wimley, 1999). One of the advantages of membrane protein modeling is the smaller conformational search space which includes those two structural motifs. In the past decade there was a vast improvement in membrane protein modeling as outlined in a recent comprehensive review (Koehler Leman *et al.*, 2015)

Developing algorithms for membrane proteins docking is important for understanding various fundamental biological processes and, especially, for drug design. Kaczor *et al.* (2013) performed a systematic study on the suitability of some commonly used protein docking software designed for globular proteins, for modeling complexes of membrane proteins. Their results indicate that these algorithms can be used for obtaining relatively reliable models of membrane protein complexes. However, the protein–protein docking results for membrane proteins are worse than corresponding results for water-soluble proteins. They also conclude that the GRAMM-X algorithm (Tovchigrechko and Vakser, 2005) may be considered the best performing algorithm for membrane proteins. Kaczor et al. explain GRAMM-X success by noting that it takes into account evolutionary conservation in its scoring function. Casciari *et al.* (2006) developed an algorithm for predicting the quaternary structure of α-helical multi homo-oligomers based on the ZDOCK 2.1 docking suite (Chen *et al.*, 2003). In a post processing step they filter the results which do not conform with the expected orientation of the transmembrane segments vis-à-vis the membrane lipid bilayer. Kaczor *et al.* (2015) also developed a protocol for modeling GPCR dimers. In the protocol each monomer is rotated around its z-axis by increments of $\pi/6$ resulting in 144 dimer models compatible with membrane integration. In the next step, Rosetta protein–protein docking (Kaufmann *et al.*, 2010) is applied using local refinement mode, resulting in a final population of 1440 dimer models. The obtained dimer models are evaluated according to a set of scoring factors especially tailored for GPCR. The scoring function contains up to 11 different parameters, such as: potential energy, electrostatic interaction energy, free energy of binding, energy of hydrogen bond interactions, evolutionary conservation score, shape complementarity and more. The protocol was evaluated on six GPCR dimers. The results showed a correct indication for all six dimers. Alford *et al.* (2015) developed a tool, MPDock, which combines RosettaMP with the RosettaDock algorithm. The algorithm conducts a local docking refinement of protein–protein complex hypotheses. MPDock was tested on five protein–protein complexes. Out of 1000 models for each complex the scoring function was able to rank the native-like as a top scoring in about half of the targets. Recently Viswanath *et al.* (2015) developed an algorithm for docking membrane proteins. In their work they use the rigid body docking algorithm DOCK/PIER without adaptation to the membrane environment. Then the results are re-ranked with a new score developed for membrane proteins in which a new energy term is added representing the membrane transfer energy. The algorithm was applied to a benchmark of 30 protein complexes and was able to obtain a near-native structure in the top ten solutions for 36.7% of the complexes.

In the current paper we present an integrated docking algorithm which includes the docking, refinement and re-ranking stages designed for docking within the membrane. Our docking algorithm, Memdock, attempts to customize all three docking steps to the lipid bilayer environment. The first step of our algorithm is based on the PatchDock algorithm (Duhovny *et al.*, 2002). PatchDock is an efficient rigid 'unbound' soft docking algorithm, which ranks docking candidates mainly by geometric shape complementarity between the docked proteins. In the modified PatchDock (Mem-PatchDock) the hypothesized placement of the docked proteins is enforced to be in a membrane consistent orientation. Thus, the change in the angle formed between the protein helical axis and the membrane normal in the apo and holo forms of the protein is constrained. Likewise, the shift in the position of the protein helices along the membrane normal cannot change significantly as a result of the docking. In the second step, a refinement algorithm is used, based on the FiberDock algorithm (Mashiach *et al.*, 2009), which conducts flexible refinement of docking candidates. The method models both side-chain and backbone flexibility and performs rigid body optimization of the ligand orientation. In the modified FiberDock (Mem-FiberDock), perturbations of the ligand are enforced to be in a membrane consistent orientation, similarly to the rigid docking step. Finally, the refined models are re-scored and re-ranked according to an energy function that was designed specifically for membrane proteins. We evaluated our algorithm on an α-helical membrane proteins dataset and show that it performs well and much faster when docking membrane proteins compared with the original algorithms designed for the docking of globular proteins.

2 Materials and methods

2.1 Benchmark set

A benchmark dataset of α-helical membrane proteins was collected from the Orientations of Proteins in Membranes (OPM) database (Lomize *et al.*, 2006). OPM provides spatial arrangements of membrane proteins with respect to the hydrocarbon core of the lipid bilayer and includes all unique experimental structures of membrane proteins.

The initial set of PDB entries for proteins was filtered to include at least one α-helix within the membrane. Then each pair of chains was tested for binding within the lipid bilayer. A final non-redundant subset of 64 protein pairs was generated, with less than 40% sequence identity for at least one pair of components. The subset of protein pairs was divided to 21 protein pairs for evaluating the algorithm and 43 protein pairs for training. Eleven protein pairs (out of the 21 pairs used for benchmarking) were used for the 'unbound' evaluation (described below). The training set proteins and the benchmark proteins are listed in Tables A and B (respectively) of the Supplementary Materials.

Our algorithm was evaluated in the 'bound' and 'unbound' docking scenarios. The 'bound' structures were extracted from the original PDB complex. The 'unbound' structures were constructed by modeling the individual chains according to homologous sequences using MODELLER (Eswar *et al*., 2006). Homologous sequences (with identity of 55–85%) were retrieved using BLAST (Altschul *et al*., 1990). Only in 11 out of the 65 protein pairs homologues could be found which were not (structurally) identical to the native structures. Only these were used for the evaluation. The list of 'unbound' structures, their sequence identity percentage and RMSD from the native can be found in Table C of the Supplementary Materials.

2.2 Protein movement in the membrane environment is restricted – membrane protein docking principle

The guiding principle in designing membrane protein docking algorithm is that proteins in the membrane are restricted in their movement between their apo and holo structures. Specifically, given an apo structure, let's attach a 3D Euclidean coordinate frame XYZ to this structure, centered at its center of mass and having the Z axis parallel to the membrane normal. After docking this structure will be transformed to the holo structure with the coordinate frame $X'Y'Z'$. Due to the above mentioned restrictions the center of mass of the holo structure can have only a limited translation in the Z direction and the angle between the Z and Z' axes is limited as well (see Fig. 1).

2.3 The rigid docking step

In this step the docked protein structures are maintained rigid. One of the protein molecules, which will be named 'receptor', remains fixed and the other protein molecule, which will be named 'ligand', is moving with respect to the receptor to find the docking candidates.

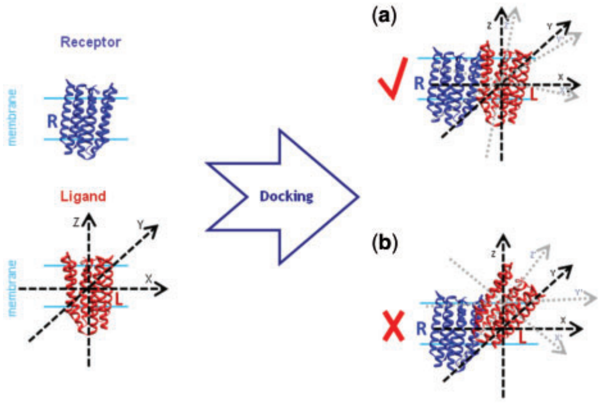

Fig. 1. The guiding principle in docking membrane proteins. Attaching a Euclidean coordinate frame *XYZ* to the ligand structure (in black), centered at its center of mass and having the *Z* axis parallel to the membrane normal. The movement of the ligand is shown by drawing the gray Euclidean coordinates ($X'Y'Z'$). (**a**) A membrane consistent orientation docking and (**b**) a membrane inconsistent orientation docking (Color version of this figure is available at *Bioinformatics* online.)

Our rigid membrane protein docking algorithm is based on the PatchDock algorithm (Duhovny *et al*., 2002). PatchDock is a geometry-based molecular docking algorithm, which detects docking candidates that yield good molecular shape complementarity. The PatchDock algorithm divides the surface of the proteins molecules into concave, convex and flat patches. Then, complementary patches are matched in order to generate docking candidates. An RMSD (root mean square deviation) clustering is applied to the docking candidates in order to discard redundant solutions.

Customizing PatchDock to membrane proteins, using Mem-PatchDock, is done by enforcing membrane consistent poses for the docking candidates. This is carried out during the patch matching stage by limiting the translations along the Z axis up to a threshold of 8 Å and the rotation tilt angle from the original Z axis up to a threshold of 0.4 radians. Thus, in the patch matching stage both complementarity between the patches and the proper position within the membrane bilayer is being thought. Limitations of the translation along the Z-axis to 8 Å was motivated by the findings of White and Wimley (1999) that the average length of helices in α-helix bundle membrane proteins is 38 ± 7 Å. Casciari *et al.* (2006) used a 6 Å translation limitation in their work.

The input to the Rigid Docking step are two separate proteins structures oriented so that the Z axis is parallel to the membrane plane normal and the centroid of the proteins is at zero.

2.4 The refinement and re-ranking steps

For the refinement step we present an algorithm based on the FiberDock algorithm (Mem-FiberDock). In FiberDock (Mashiach *et al*., 2009) each candidate complex is subsequently refined by restricted interface side-chain rearrangement and by soft rigid-body optimization. The side-chain flexibility is modeled by rotamers and the obtained combinatorial optimization problem is solved by integer linear programming (Kingsford *et al*., 2005). Following rearrangement of the side-chains, a normal mode analysis (NMA) based backbone refinement procedure is applied. Finally, the relative position of the docking partners is refined by Monte Carlo minimization of the binding score function.

Customizing FiberDock to membrane environment, using Mem-FiberDock, is done in two stages: first, a preprocessing step of filtering membrane inconsistent docking candidates' complexes generated in the rigid docking step is performed. This stage is conducted for cases in which the input to the refinement algorithm was not produced by Mem-PatchDock. This preliminary stage is carried out by limiting the Z-axis offset of the ligand, and tilt angle from the original structure, up to 12 Å and 0.8 radians respectively. These are evaluated on the Z axis of the Center Of Mass (COM) of the ligand.

During refinement membrane orientation is re-evaluated before each rigid body perturbation by limiting the Z coordinate perturbation movement up to an 8 Å shift and an 0.4 radians angle rotation.

The refined candidates are re-ranked by the energy function designed for membrane proteins, as described below.

2.5 Energy function

We have developed a scoring function (Memscore) based on statistics derived from known 3D structures of α-helical membrane proteins. The energy function takes into account the lipid bilayer environment, distinguishing polar and hydrophobic layers.

The function was adopted partially from the FireDock method (Andrusier *et al*., 2007) with additional membrane parameter E_{env} (environment energy), while deleting the desolvation ACE (Atomic

Contact Energy) based component and the density energy. The function includes Atomic softened van der Waals interactions energy (E_{repVdW} – van der Waals repulsion energy, E_{attVdW} – van der Waals attraction energy), partial electrostatics (E_{repElect} – electrostatic repulsion energy, E_{attElect} – electrostatic attraction energy) and estimations of the binding free energy (E_{HB} – Hydrogen bond energy, E_{aliph} – aliphatic interaction energy and E_{pipi} – π–π interaction energy). E_{env} potential score is analogous to the environment score used in the Rosetta membrane protein modeling prediction method (Yarov-Yarovoy *et al.*, 2006).

The calculated energy score is a linear combination of the individual energy terms (see Equation 1), where *a–h* are user defined weights. The weights were calculated on a 43 membrane proteins training set (Table A of the Supplementary Materials). Training was conducted using a genetic programming algorithm (Cramer, 1985).

$$E_{\text{total}} = aE_{\text{env}} + bE_{\text{HB}} + cE_{\text{aliph}} + dE_{\text{pipi}} + eE_{\text{repVdW}} + fE_{\text{attVdW}} + gE_{\text{repElect}} + hE_{\text{attElect}} \quad (1)$$

E_{env} potential score was calculated by dividing the membrane into horizontal layers. The layers, similarly to the layers chosen by Yarov-Yarovoy *et al.* (2006) for membrane protein modeling, approximately corresponding to the water-exposed, polar, interface and hydrophobic layers of the membrane (described by White and Wimley,1999) (layer 1: -12 Å $\leq Z$ coordinate $\leq$ 12 Å, layer 2: 12 Å$< Z$ coordinate $\leq$ 18 Å, layer 3: 18 Å $< Z$ coordinate $\leq$ 24 Å, layer 4: 24 Å $< Z$ coordinate, layer 5: -18 Å $\leq Z$ coordinate $<$ -12 Å, layer 6: -24 Å $\leq Z$ coordinate $<$ -18 Å, layer 7: Z coordinate $<$ -24 Å). The environment potential was defined as:

$$E_{\text{env}} = \sum_{i} -\ln\left(\frac{N(aa_i|L,B)}{N(aa_i)}\right) \quad (2)$$

where i is a residue index, $N(aa_i|L,B)$ is the number of amino acid type aa_i in a layer/burial (L,B) state and $N(aa_i)$ is the frequency of amino acid type aa_i (20 amino acid types) in all layer/burial states. The buried state of a residue presents the number of neighboring residues around it. It was defined as the number of residue centroids within 10 Å of the examined residue centroid (define in Yarov-Yarovoy *et al.*, 2006), classified into four states (neighbors number $\leq$ 5, 6 $\leq$ neighbors number $\leq$ 11, 12 $\leq$ neighbor s number $\leq$ 17, neighbors number $\geq$ 18).

2.6 Evaluating the docking candidates

The parameters used to evaluate the quality of the predicted docking candidates were as in the CAPRI challenge (Janin and Wodak, 2007). The docking results are classified into three categories: 'High', 'Medium' and 'Acceptable' according to the values of the parameters *I*rms (Interface RMSD), *L*rms (*L*igand RMSD) and *f*nat (fraction of native contacts):

- 'High': *f*nat $\geq$ 0.5 AND (*L*rms $\leq$ 1.0 Å OR *I*rms $\leq$ 1.0 Å)
- 'Medium': ((0.3 $\leq$ *f*nat $<$ 0.5) AND (*L*rms $\leq$ 5.0 Å OR *I*rms $\leq$ 2.0 Å)) OR (*f*nat $\geq$ 0.5 AND *L*rms $>$ 1.0 Å AND *I*rms $>$ 1.0 Å)
- 'Acceptable': ((0.1 $\leq$ *f*nat $<$ 0.3) AND (*L*rms $\leq$ 10.0 Å OR *I*rms $\leq$ 4.0 Å)) OR (*f*nat $\geq$ 0.3 AND *L*rms $>$5.0 Å AND *I*rms $>$2.0Å)

2.7 Comparison with PatchDock/FiberDock and GRAMM-X algorithms

Memdock performance was compared with the performance of the algorithms on which it is based: using PatchDock for rigid body docking followed by FiberDock for the refinement. These were run with the default 'unbound' setting. For the refinement stage up to 1000 docking candidates were chosen.

In addition, Memdock performance was compared with that of GRAMM-X algorithm, which includes both rigid docking and refinement stages. GRAMM-X was chosen as it was shown in previous work (Kaczor *et al.*, 2013) to perform the best from numerous docking algorithms on a membrane proteins benchmark. GRAMM-X algorithm was run via its server with the default parameters and 100 outputs.

3 Results

The Memdock algorithm is particularly efficient. Mem-PatchDock running time on membrane proteins is by an order of magnitude shorter than that of the original algorithm, PatchDock (Table 1). Moreover, the number of docking candidates drops significantly with Memdock and the percentage of complexes with an at least 'Acceptable' docking candidate increases. The results are presented in Figures 2 and 3 and in Table D of the Supplementary Materials.

3.1 Docking results

3.1.1 'Bound' structures results

Our algorithm was evaluated on 21 'bound' structures. Comparison between the performance of PatchDock/FiberDock, Memdock and GRAMM-X was conducted. In addition, docking candidates were ranked using FiberDock score and the Memscore (Memdock score). The docking results are shown in Figure 2.

Memdock protocol succeeds to sample and rank structures of 'Acceptable' or higher accuracy among the top ten docking candidates in 71% of the benchmark complexes compared to only 33%

Table 1. Mem – PatchDock vs. PatchDock: runtime and docking candidates' number

Algorithm	Speed		Docking Candidates num.	
	Mean	Standard dev.	Mean	Standard dev.
Mem – PatchDock	27 sec.	9.7	172	138.9
PatchDock	789 sec.	545.5	6909	5863.3

The average run time of PatchDock and Mem-PatchDock for two input membrane proteins on a single 1.0 GHz PC processor under the Linux operating system and the average docking candidates' number.

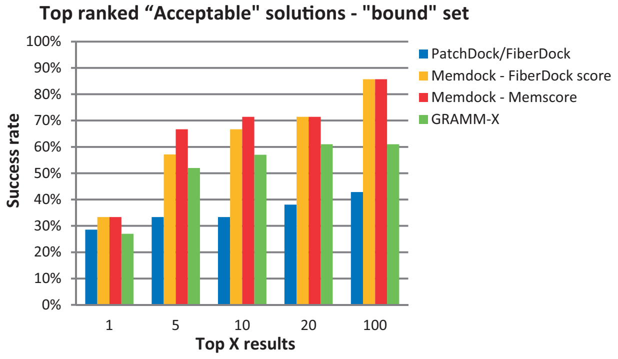

Fig. 2. Comparison between the performance of PatchDock/FiberDock algorithm, the membrane proteins customized algorithm – Memdock, and GRAMM-X algorithm. Additionally, Memscore is compared to the FiberDock score. The success rate is the percentage of 'Acceptable' (CAPRI category) docking candidates in the X ranked candidate. Validated on the 'bound' set (Color version of this figure is available at *Bioinformatics* online.)

in the original algorithm designed for standard docking using PatchDock/FiberDock and 57% in GRAMM-X algorithm.

3.1.2 ‘Unbound’ structures results

Our algorithm was evaluated on 11 ‘unbound’ structures with available template homologous sequences. The docking results are shown in Figure 3. In this experiment an ‘Acceptable’ solution was ranked at the top in 4 of the cases and was found among the top 100 candidates in 8 of the cases when using the Memdock protocol.

On our benchmark Memdock implemented with the membrane specific Memscore did not outperform in the ‘unbound’ cases application of Memdock with the non-specific Fiberdock score. However, when PatchDock/Fiberdock was first used for docking the membrane proteins and the resulting candidates have been reranked with Memscore it contributed significantly to the improvement of the results (Fig. S1 in Supplementary Materials). This might indicate that the membrane consistent translations and rotations enforced by Memdock and the membrane oriented layer scoring of Memscore are essentially based on the same information.

3.1.3 Further docking analysis

Further analysis was performed in order to check whether the structures in which GRAMM-X succeeds in docking have a conserved binding site, since GRAMM-X uses an evolutionary conservation term in its scoring function. This analysis was performed by ConSurf (Glaser *et al.*, 2003). Indeed, the complexes in which GRAMM-X predicted an ‘Acceptable’ or higher accuracy among the top ten docking candidates had a conserved binding site (Tables B and D, Supplementary Materials). In fact in most of our benchmark complexes there is certain level of conservation in the binding site (Table B, Supplementary Materials).

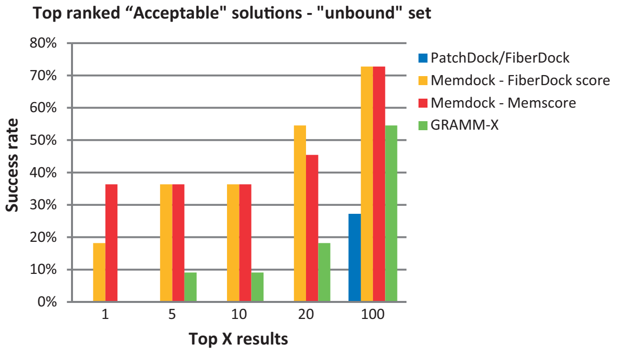

Fig. 3. Comparison between the performance of PatchDock/ FiberDock algorithm, Memdock, and GRAMM-X algorithm. Additionally, Memscore is compared to the FiberDock score. The success rate is the percentage of ‘Acceptable’ (CAPRI catego) docking candidates in the X ranked candidate. Validated on the ‘unbound’ set (Color version of this figure is available at *Bioinformatics* online.)

The docking success of Memdock is influenced by the size of the binding site (Tables B and D of the Supplementary Materials). One can see that most of the proteins pairs in which the docking was less successful had a small binding site. In addition, the docking was not influenced by the size or the protein or the tilt angle of the helices.

3.2 Docking and refinement success in using the membrane algorithms

We present in this section several examples of membrane protein complexes which highlight the advantage of using the membrane docking algorithm - Memdock (Fig. 4) on the ‘unbound’ dataset.

The enzyme cytochrome c oxidase (1M56) controls the last step of food oxidation. Cytochrome c oxidase takes electrons and attaches them to an oxygen molecule converting molecular oxygen to two molecules of water. Docking was performed with comparative models based on templates of 97.6% (PDB 3OMI) and 72% (PDB 1QLE) sequence identity for chains A and B, respectively. PatchDock/FiberDock protocol did not find in any ‘Acceptable’ docking candidate. Memdock, however, found an ‘Acceptable’ accuracy docking candidate ranked 11th (Fig. 4a).

Multisubunit cytochrome *b* (1Q90) interacting with the complex Photosystems I and II which reaction centres capture light energy in order to derive oxygenic photosynthesis. Docking was performed with comparative models based on templates of 85% (PDB 2ZT9) sequence identity for chains B and S. PatchDock/FiberDock protocol found an ‘Acceptable’ docking candidate ranked 525. Memdock, however, found an ‘Acceptable’ accuracy docking candidate ranked first (Fig. 4b).

Pentameric formate channel (3KLY) enables formate transport across the inner membrane which is a critical step in anaerobic bacterial respiration. Docking was performed with comparative models based on templates of 55.3% (PDB 3KCU) sequence identity for chains A and B. PatchDock/FiberDock protocol did not find in any ‘Acceptable’ docking candidate. Remarkably, Memdock, found a ‘High’ accuracy docking candidate ranked first (Fig. 4c).

3.3 Efficient rigid docking

The original PatchDock is a fast and efficient algorithm. The main reason behind PatchDock’s high efficiency is its fast transformational search, which is driven by local feature matching rather than brute

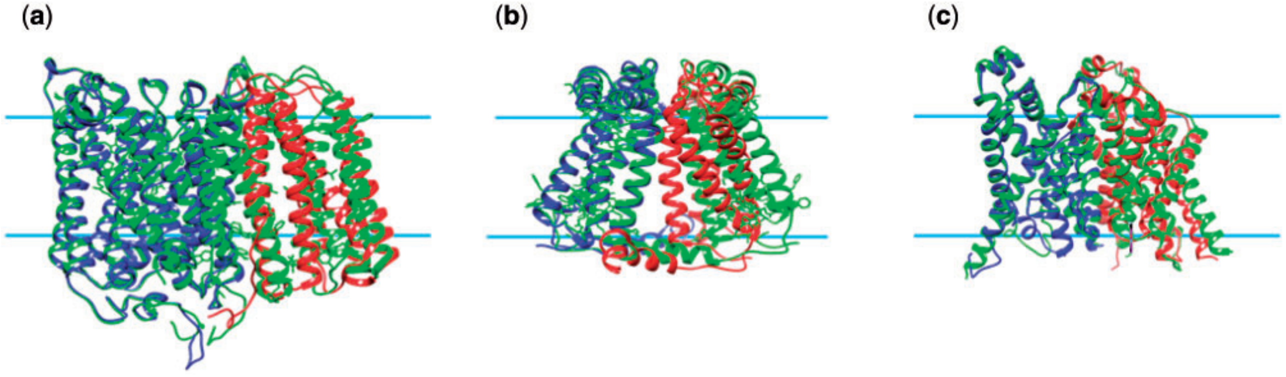

Fig. 4. Docking candidate (blue – receptor, red – ligand) superimposed on the native complex (light green). The light blue lines represent the membrane (**a**) Cytochrome *c* oxidase , chains A and C (1M56), (**b**) Cytochrome *b*, chains B and S (1Q90) and (**c**) Pentameric formate channel, chains A and B (3KLY)

force searching of the six-dimensional transformation space. It further speeds up the computational processing time by utilizing advanced data structures and spatial pattern detection techniques, such as geometric hashing and pose clustering (Duhovny *et al.*, 2002).

The new developed algorithm designed for membrane proteins is even more efficient than the original one because it further limits the transformation search. In addition, the number of docking candidates decreases significantly (Table 1).

There is no significant difference in the refinement speed between Mem-FiberDock and FiberDock, however, the number of docking candidates entering refinement is significantly reduced by Memdock.

4 Discussion

We introduced an algorithm with excellent performance for docking and refinement of α-helical membrane proteins.

This improvement is achieved by introducing sampling and scoring specific for membrane proteins. First, we enforce membrane consistent orientation in both docking and refinement steps as an integral part of the algorithms rather than a post-processing step. Second, the refined docking candidates undergo re-ranking using a scoring function formulated from combination of knowledge-based potentials derived from statistical analyses of known 3D structures of α-helical membrane proteins and the FireDock scoring function terms which are based on the binding forces between the receptor and the ligand.

The algorithm was applied to a benchmark set of 21 'bound' proteins structures and 11 'unbound' protein structures. In 33% of the 'bound' benchmark set and 36% of the 'unbound' benchmark set an 'Acceptable' docking candidate was ranked at the top (Figs 2 and 3). In 71% of the 'bound' benchmark set and 45% of the 'unbound' benchmark set an 'Acceptable' docking candidate was predicted among the top 20 ranking results (Figs 2 and 3).

Thus, the algorithm is shown to perform well both on the 'bound' and on the 'unbound' examples. The algorithm significantly improves docking accuracy and the speed compared to the standard docking.

A web server for the Memdock algorithm is being currently implemented and will be available at http://bioinfo3d.cs.tau.ac.il/Memdock/.

Acknowledgements

We thank David Jones for allowing us to use his genetic programming based learning algorithm.

Funding

This research was supported by the Israel Science Foundation (Grant No. 1112/12), the I-CORE program of the Budgeting and Planning Committee and the Israel Science Foundation (center No. 1775/12) and by the Hermann Minkowski Minerva Geometry Center. N.H. acknowledges the Edmond J.Safra Bioinformatics Center fellowship.

Conflict of Interest: none declared.

References

Alford,R.F. *et al* (2015) An integrated framework advancing membrane protein modeling and design. *PLoS Comput. Biol.*, **11**, e1004398.

Altschul,S.F. *et al* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Andrusier,N. *et al* (2007) FireDock: fast interaction refinement in molecular docking. *Proteins*, **69**, 139–159.

Berman,H.M. *et al* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Carpemter,E.P. *et al* (2008) Overcoming the challenges of membrane protein crystallography. *Curr. Opin. Struct. Biol.*, **18**, 581–586.

Casciari,D. *et al* (2006) Quaternary structure predictions of transmembrane proteins starting from the monomer: a docking-based approach. *BMC Bioinf.*, **7**, 340.

Chaudhury,S. *et al* (2007) Incorporating biochemical information and backbone flexibility in RosettaDock for CAPRI rounds 6-12. *Proteins*, **69**, 793–800.

Chen,R. *et al* (2003) ZDOCK: an initial-stage protein docking algorithm. *Proteins*, **52**, 80–87.

Cramer,N.L. (1985) A Representation for the Adaptive Generation of Simple Sequential Programs. In: *Proceedings of the 1st International Conference on Genetic Algorithms.*

Dominguez,C. *et al* (2003) HADDOCK: a protein–protein docking approach based on biochemical and/or biophysical information. *J. Am. Chem. Soc.*, **125**, 1731–1737.

Duhovny,D. *et al* (2002) Efficient Unbound Docking of Rigid Molecules. In: *Second International Workshop, WABI 2002*, pp. 185–200.

Eswar,N. *et al* (2006) Comparative protein structure modeling with MODELLER. *Curr. Protoc. Bioinf.*, **15**, 5.6.1–5.6.30.

Fitzjohn,P.W. and Bates,P.A. (2003) Guided docking: first step to locate potential binding sites. *Proteins*, **52**, 28–32.

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.

Glaser,F. *et al* (2003) ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics*, **19**, 163–164.

Halperin,I. *et al* (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins*, **47**, 409–443.

Janin,J. and Wodak,S. (2007) The third CAPRI Assessment Meeting Toronto, Canada. *Structure*, **15**, 755–759.

Janin,J. (2010) Protein–protein docking tested in blind predictions: the CAPRI experiment. *Mol. BioSyst.*, **6**, 2351–2362.

Jones,D.T. (1998) Do transmembrane protein superfolds exist? *FEBS Lett.* **27**, **423**, 281–285.

Kaczor,A.A. *et al* (2013) Modeling complexes of transmembrane proteins: systematic analysis of protein–protein docking. *Mol. Inf.*, **32**, 717–733.

Kaczor,A.A. *et al* (2015) Multi-component protein–protein docking based protocol with external scoring for modeling dimers of g protein-coupled receptors. *Mol. Inf*, **34**, 246–255.

Kaufmann,K.W. *et al* (2010) Practically useful: what the Rosetta protein modeling suite can do for you. *Biochemistry (Mosc.)*, **49**, 2987–2998.

Kingsford,C.L. *et al* (2005) Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, **21**, 1028–1036.

Koehler Leman,J. *et al* (2015) Computational modeling of membrane proteins. *Proteins*, **83**, 1–24.

Kozakov,D. *et al* (2006) PIPER: An FFT-based protein docking program with pairwise potentials. *Proteins*, **65**, 392–406.

Krol,M. *et al* (2007) Implicit flexibility in protein docking: cross-docking and local refinement. *Proteins*, **69**, 750–757.

Lindahl,E. and Delarue,M. (2005) Refinement of docked protein–ligand and protein–DNA structures using low frequency normal mode amplitude optimization. *Nucleic Acids Res.*, **33**, 4496–4506.

Lomize,M.A. *et al* (2006) OPM: orientations of proteins in membranes database. *Bioinformatics*, **22**, 623–625.

Mashiach,E. *et al* (2009) Fiber Dock: flexible induced-fit backbone refinement in molecular docking. *Proteins*, **78**, 1503–1519.

May,A. and Zacharias,M. (2007) Protein–protein docking in CAPRI using ATTRACT to account for global and local flexibility. *Proteins*, **69**, 774–780.

Schneidman-Duhovny,D. *et al* (2005) Geometry based flexible and symmetric protein docking. *Proteins*, **60**, 224–231.

Sousa,S.F. *et al* (2013) Protein–ligand docking in the new millennium – a retrospective of 10 years in the field. *Curr. Med. Chem.*, **20**, 2296–2314.

Tovchigrechko,A. and Vakser,I.A. (2005) Development and testing of an automated approach to protein docking. *Proteins*, **60**, 296–301.

Viswanath,S. *et al* (2015) Extension of a protein docking algorithm to membranes and applications to amyloid precursor protein dimerization. *Proteins*, **83**, 2170–2185.

Wallin,E. and von Heijne,G. (1998) Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms. *Protein Science.*, **7**, 1029–1038.

Wang,C. *et al* (2007) Protein–protein docking with backbone flexibility. *J. Mol. Biol.*, **373**, 503–519.

White,S.H. and Wimley,W.C. (1999) Membrane protein folding and stability: physical principles. *Ann. Rev. Biophys. Struct.*, **28**, 319–365.

Yildirim,M.A. *et al* (2007) Drug–target network. *Nat. Biotechnol.*, **25**, 1119–1126.

Yarov-Yarovoy,V. *et al* (2006) Multipass membrane protein structure prediction using Rosetta. *Proteins*, **62**, 1010–1025.

Zacharias,M. (2005) ATTRACT: protein–protein docking in using a reduced protein model. *Proteins*, **60**, 252–256.