

Integrative Modeling of Membrane-associated Protein Assemblies

Jorge Roel-Touris^{1†}, Brian Jiménez-García^{1†*} and Alexandre M.J.J. Bonvin^{1*}

¹Bijvoet Centre for Biomolecular Research, Faculty of Science-Chemistry, Utrecht University, Utrecht, The Netherlands.

[†]These authors contributed equally to this work.

^{*}To whom correspondence should be addressed.

E-mail:

Brian Jiménez-García: b.jimenezgarcia@uu.nl

Alexandre M.J.J. Bonvin: a.m.j.j.bonvin@uu.nl

Keywords: Macromolecular Docking; Artificial Intelligence; Coarse-graining; LightDock; HADDOCK; Membrane proteins

Abstract

Historically, membrane protein systems have been considered as one of the most challenging systems to study with experimental structural biology techniques. Over the past years, increased number of experimental structures of membrane proteins have become available thanks in particular to advances in solid-state NMR spectroscopy and cryo-electron microscopy. This has opened the route to modeling the complexes that those membrane proteins form by methods such as docking. Most approaches developed to date are, however, not capable of incorporating the topological information provided by the membrane into the modeling process. Here, we present an integrative computational protocol for the modeling of membrane-associated protein assemblies, specifically complexes consisting of a membrane-embedded protein and a soluble partner. It combines efficient, artificial intelligence-based rigid-body docking by LightDock with a flexible final refinement with HADDOCK to remove potential clashes at the interface. We make use of an equilibrated coarse-grained lipid bilayer to represent the information encoded in the membrane in the form of artificial beads, which allows to target the docking towards the binding-competent regions. We demonstrate the performance of this membrane-driven protocol on eighteen membrane-associated complexes, whose interface lies between the membrane and either the cytosolic or periplasmic regions. In addition, we evaluate how different membrane definitions impact the performance of the docking protocol and provide a comparison, in terms of success rate, to another state-of-the-art docking software, ZDOCK. Finally, we discuss the quality of the generated models and propose possible future developments. Our membrane docking protocol should allow to shed light on the still rather dark fraction of the interactome consisting of membrane proteins.

Introduction

Membrane proteins (MPs) play crucial roles in many biological functions within the cell. Commonly, MPs are classified based on their association mode with biological membranes into two main groups: Peripheral membrane proteins that are located on either side of the membrane and are attached to it by non-covalent interactions, and integral membrane proteins (IMPs) that are inserted into the membrane and can be either exposed on only one side of the membrane (monotopic membrane proteins) or span the entire lipid bilayer. The latter, known as transmembrane proteins (TMs), are structurally categorized as α -helical bundles or β -barrels¹. TMs mostly function as regulators of complex biochemical pathways (receptors and transducers) and/or transporters of molecules (channels and carriers). Only transmembrane proteins can function at both sides of the membrane by forming larger complexes. As such they are not simply passive membrane spanning proteins but play important roles in protein-protein interactions (PPIs), thus making them valuable targets for drug discovery (around 60% of current drug targets are MPs². A well-known example are G-protein coupled receptors (GPCRs) which are involved in many diseases³. Those are collected in a specific database (GPCRdb; <https://www.gpcrdb.org/>)⁴.

Over the past years, development of cutting-edge technologies has facilitated the study of previously inaccessible MPs, advancing the field of membrane structural biology. Obtaining high-quality crystals suitable for X-ray crystallography is still far from trivial. Solid-state NMR spectroscopy, and especially cryo-electron microscopy (cryo-EM), reaching near-atomic resolution, have become central tools to study membrane-associated protein complexes^{5,6}. However, experimental conditions such as low expression profiles and/or high instability outside the native membrane still makes their

structural characterization challenging⁷. Despite their large representation in the proteome (in human, nearly a quarter of the genome encodes for MPs⁸), roughly only 1% of all deposited protein structures in the Protein Data Bank⁹ (PDB) corresponds to MPs, with 1099 unique protein entries as of July 2020: <https://blanco.biomol.uci.edu/mpstruc/>. Even less of those have been experimentally solved in complex with their counterpart(s). For all these reasons, membrane protein systems, which are increasingly attracting attention, have been traditionally considered as one of the most difficult type of systems to study by experimental structural biology techniques.

Computational methods offer an attractive alternative for studying membrane systems¹⁰. Many efforts have been made to develop efficient tools to computationally predict the 3D atomic structures of membrane-associated proteins and their complexes¹¹. Some rely on secondary structure or topology prediction and make use of either knowledge-based statistics or evolutionary information to generate 3D models^{12,13}. The simplest computational methods are based on homology modeling. In short, these approaches require a template structure (or multiple) with high sequence similarity to the target sequence, and usually produce very reliable “core” models (corresponding to the TM domains) and less accurate predictions for the extracellular loops. Methods such as MEDELLER¹⁴ and Memoir¹⁵ have greatly benefited from the increasing availability of cryo-EM derived structures in the PDB and are inspired on the well-known homology modeling tool MODELLER¹⁶.

Another representative subset of computational methods geared towards modeling complexes are docking-based approaches. Docking commonly includes two different steps, namely sampling and scoring. Sampling is usually referred to as the

process of generating (tens of) thousands of possible conformations of a given (bio)molecular complex. This can be done through a number of well-established techniques such as Fast Fourier Transformation (FFT)-based methods included in various docking software such as GRAMM-X^{17,18}, ClusPro¹⁹, pyDock²⁰ and ZDOCK²¹. These methods, however, do not allow for explicit flexibility of the modeled partners due to intrinsic limitations of the FFT sampling. Although this limitation can be partially solved by using ensembles of conformers, it implies higher computational cost. Energy minimization, in HADDOCK²² and ATTRACT²³ for example, Metropolis Monte Carlo optimization, e.g. in RosettaDock²⁴, or artificial intelligence-based algorithms, such as implemented in SwarmDock²⁵ and LightDock²⁶, are also used. The sampling process is often followed by a refinement of the docked models for which molecular dynamics- or Monte-Carlo-based protocols are the most commonly used. The generated models are scored with the aim of discriminating between biologically-relevant (native) and non-relevant models. This is typically done with a scoring function, which can be based on either physico-chemical properties and/or statistical potentials²⁷. Nowadays, with the increasing availability of large pools of docking models such as provided in the CAPRIDOCK²⁸ (http://cb.iri.univ-lille1.fr/Users/lensink/Score_set/), PPI4DOCK²⁹ (<http://biodev.cea.fr/interevol/ppi4dock/>) and DOCKGROUND³⁰ (<http://dockground.compbio.ku.edu/>) databases, machine(deep)-learning scoring functions are gaining interest³¹. Sampling and scoring might be coupled (scoring-driven sampling) or work as independent steps (sampling and then scoring). In the context of membrane protein docking, software such as Rosetta³², DOCK/PIERR³³ and Memdock³⁴ include built-in specific protocols to model transmembrane domains using implicit membrane potentials. Besides RosettaMP³⁵ (for membrane protein design), none of the

available membrane-specific computational methods allow for an explicit representation of the lipid bilayer and, therefore, cannot harvest the topological information encoded in it.

In this work, we present an integrative computational approach for modeling membrane-associated protein assemblies (complexes consisting of a membrane-embedded protein and a soluble partner) that combines an efficient, *swarm*-based rigid-body docking by LightDock with a flexible final refinement with HADDOCK to remove potential clashes at the interface. To introduce the topological information provided by the lipid bilayer we make use of an equilibrated coarse-grained membrane into the docking calculations. In that way we can focus the docking towards binding competent regions, excluding all irrelevant regions prior to the simulation. This membrane representation has been implemented within the LightDock framework²⁶. The sampling in LightDock is based on an artificial intelligence-based *swarm* approach that relies on the metaphor that, in nature, *glowworms* (which represent ligand poses) feel attracted to each other depending on the amount of emitted light (scoring, energetic value of a docking pose). In this way, the docking poses, which constitute the *swarm* of “*glowworms*” in LightDock, are optimized towards the energetically more favorable ones through the translational, rotational and Anisotropic Network Model (ANM) spaces. The latter is, however, not available in the membrane docking mode. Sampling and scoring in LightDock are tightly interconnected since the optimization process is score-driven. In its latest official release (version 0.8.0; pypi.org/project/lightdock), LightDock supports the use of information such as mutagenesis and/or bioinformatic predictions to bias the sampling³⁶. The LightDock-generated membrane protein models are then refined with HADDOCK via an efficient coarse-grained (CG) protocol³⁷. This

protocol, originally designed to backmap coarse-grained models to atomistic resolution by morphing atomistic models onto the coarse-grained ones using distance restraints, is very efficient in removing steric clashes while maintaining the original geometry of the docked models.

We demonstrate the efficiency and performance of this two-step (docking and refinement) membrane-driven protocol on the 18 membrane protein complexes from the MemCplxDB benchmark set³⁸ (<https://github.com/haddock/MemCplxDB>) whose interface lies between the membrane and either the cytosolic or periplasmic regions. We also evaluate how different choices for defining the membrane topology affect the sampling of our protocol, and assess the quality improvement of the generated models after the HADDOCK refinement step. We compare the success rate of this integrative approach and the quality of the generated models with that of another state-of-the-art docking software, ZDOCK²¹, for which we test several docking scenarios penalizing (“blocking”) regions during sampling and therefore explicitly accounting for the information provided by the membrane. Finally, we discuss the quality of the side-chains at the interface of the generated models and propose future developments that could be made for improving the current results.

Results

Integrative modeling approach for membrane-associated protein complexes

We have developed a computational approach for modeling the interaction of membrane-associated protein complexes that accounts for the topological information encoded in the membrane. First, we insert the atomistic transmembrane protein into a pre-equilibrated coarse-grained model of the protein in a lipid bilayer provided by the

MemProtDB database³⁹ (<http://memprotmd.bioch.ox.ac.uk/>) (see Material and Methods; *Preprocessing of input structures*) and then remove all lipid beads except those representing the phosphate groups. Using this beads layer, we automatically generate a group of independent simulations known as *swarms* over the solvent-exposed receptor surface. Using the capability of the LightDock framework, we thus discard irrelevant sampling regions (see Fig. 1-3 where the geometrical centers of the *swarms* are depicted as blue beads). Next, each *swarm* is populated with 200 starting random orientations of the soluble ligand (200 is the default number of *glowworms*, the agents of the sampling algorithm). This procedure effectively biases the sampling towards the binding-competent regions on the membrane protein (either cytosolic or periplasmic) and excludes those within the boundaries imposed by the membrane. While LightDock can allow for flexibility during docking through normal modes, this option is not supported for membrane-embedded proteins. The only limited flexibility introduced in the protocol is that of the final refinement using HADDOCK.

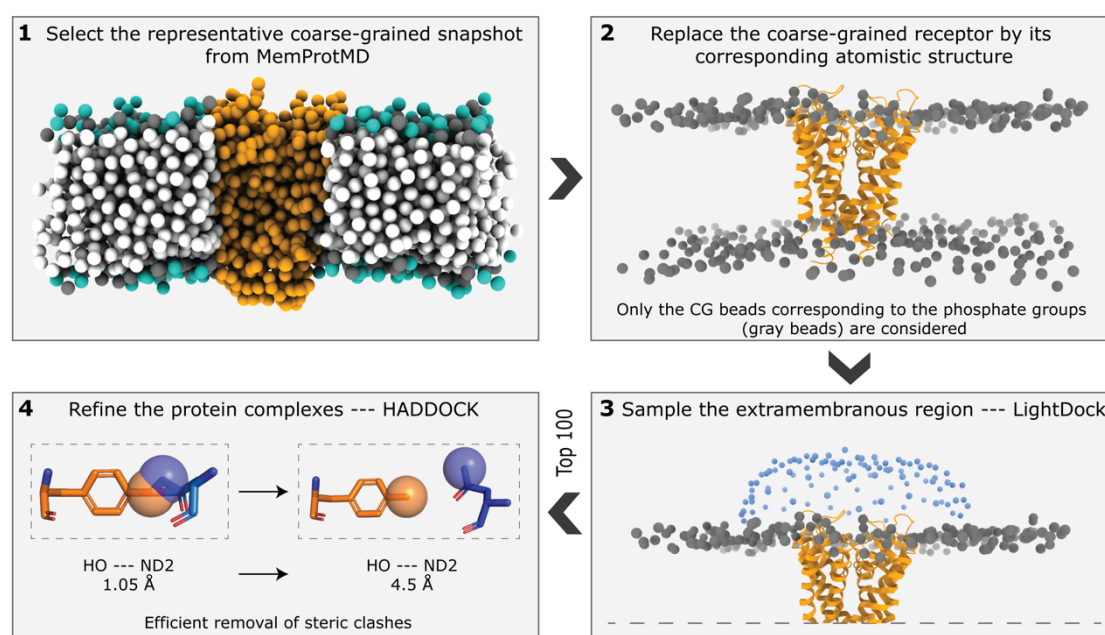


Fig. 1. Membrane protein integrative modeling workflow. **1)** The representative coarse-grained membrane snapshot from the MemProtMD database is selected³⁹ (<http://memprotmd.bioch.ox.ac.uk/>) **2)** The coarse-grained transmembrane receptor is replaced by its corresponding atomistic structure. **3)** The binding-competent regions are sampled with LightDock using the membrane defined by beads corresponding to the phosphate positions. The resulting top 100 docked models are selected for final refinement. **4)** Refinement with HADDOCK following a coarse-grained to all atom protocol and final scoring.

For the scoring during the docking simulation with LightDock, we use an adapted version of the DFIRE⁴⁰ scoring function that penalizes models penetrating the membrane, specifically those overlapping with any membrane bead (see Material and Methods; *Implementation of an explicit membrane representation into LightDock*). We select the top 100 models from the optimization of all *swarms* for a final refinement stage with HADDOCK in order to remove clashes at the interface. This is achieved using an efficient coarse-grained refinement protocol: In short, we first generate the corresponding MARTINI-based⁴¹ coarse-grained representation for each of the docked models to be refined; then, by a combination of energy-minimizations and short molecular dynamics stages, the protocol³⁷ fits the atomistic structure of each of the

components onto the generated CG model of the complex and optimizes the system to remove clashes. This final refinement is performed in the absence of the membrane. The resulting models are then scored and ranked according to the HADDOCK score.

Although in this work our protocol only makes use of the membrane as information source during the modeling, it is fully compatible with the use of a variety of experimental data in the form of residue restraints if this source of information is provided⁴².

Overall performance on the membrane docking dataset

We have tested the performance of our membrane-driven protocol on the 18 transmembrane-soluble protein complexes of the MemCplxDB benchmark (see Material and Methods; *Membrane docking dataset* and Fig. 2) and compared it with the results of a full sampling in the absence of the topological information provided by the membrane (i.e. *Blind* docking – see next section). The docking was performed starting from the unbound structures of each constituent, except for *2bs2*, *2vpz* and *4huq* for which no unbound state structures are available. The success rate was defined as the percentage of cases for which an acceptable or better model was obtained within the top *N* ranked models (see Material and Methods; *Metrics for the evaluation of model quality and success rate*).

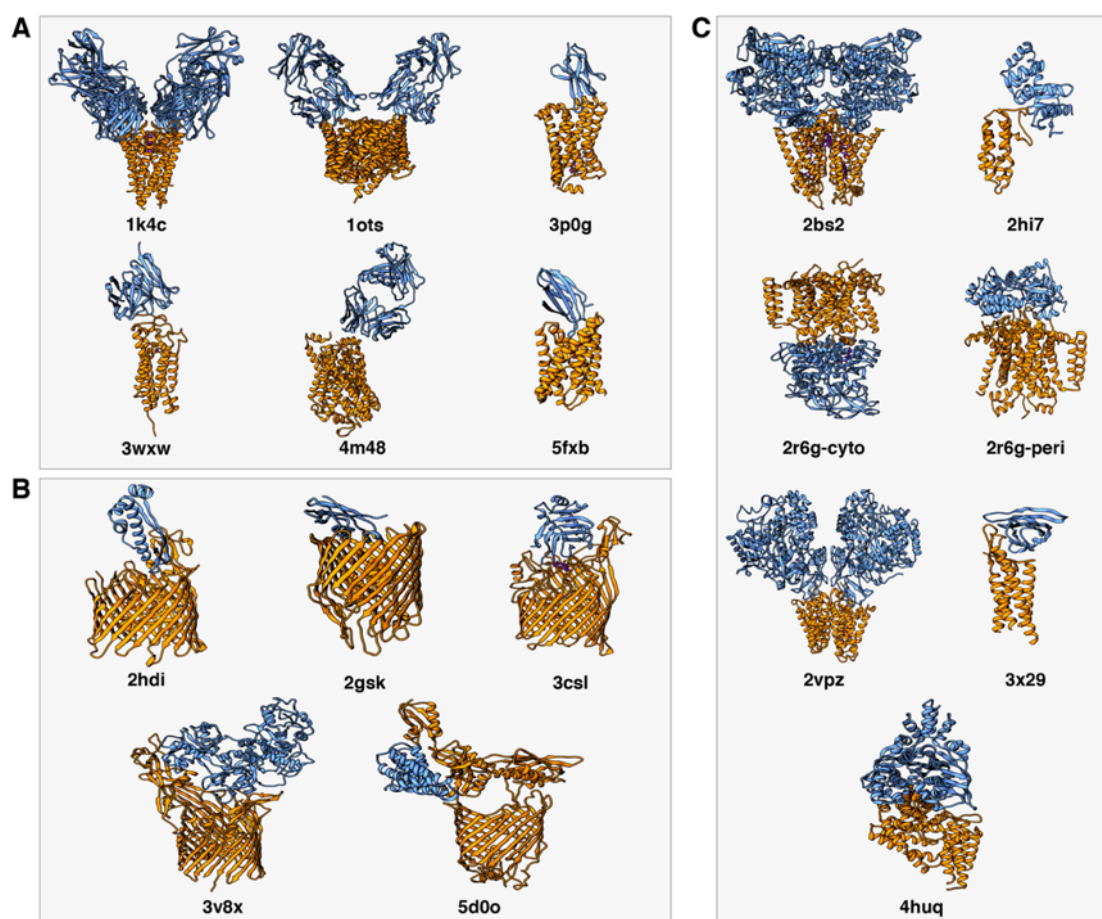


Fig. 2. View of the 18 transmembrane-soluble protein complexes of the MemCplxDB database³⁸ (<https://github.com/haddocking/MemCplxDB>) used in this work. The cases are classified as: **A)** Antibodies (the soluble partner is either an antibody or a nanobody), **B)** β -Barrel (the transmembrane receptor is a β -sheet barrel) and **C)** α -Helical (the transmembrane receptor consists of a bundle of α -helices). The receptors (the membrane proteins) are depicted in orange and the ligands in blue.

For the two most representative top N ($T5$ and $T10$), our *Membrane* protocol shows an overall success rate of 61.1%, 11 out of 18 complexes, with 3 cases having medium quality models as shown in Fig. 3. It reaches a maximum of 88.9% for the top 100 predictions. High quality models are obtained for one α -Helical case within the top 20 (3x29) with the best docking pose (ranked at the 12th position) having 70% of the native contacts and 1.0Å/2.0Å i-RMSD/l-RMSD from the reference crystal structure. The highest success rate for either $T5$ or $T10$ is achieved for α -Helical complexes. For those,

acceptable or higher quality models are generated for 71.4% of the complexes (5 out of 7 cases). This performance, however, drops for the *β -Barrel* category with 40% success rate for *T5/10* and 80% for *T100* (4 out of 5 cases). Not surprisingly, our protocol fails to deliver near-native models for the case with the largest conformational change (*3v8x*; i-RMSD of 3.42Å between unbound and bound structures), classified as *β -Barrel*. For *Antibodies* (6 cases), we used the CDR loops to pre-orient the molecules at the *setup* step⁴², but these were not specifically used for the scoring. For these cases, our protocol generates acceptable and medium quality models for all complexes (100% success rate for *T50*) with a 33.3% success rate considering the top ranked model (*T1*) and 66.7% for *T5/10*.

The membrane integrative modeling protocol outperforms blind predictions

For the *Blind*, membrane-free predictions, LightDock-HADDOCK reaches an overall success rate of 16.7% for the *T100* (11.1% of medium quality models), with a moderate performance for *T5* and *T10* (5.5% and 11.1%, respectively). For one bound case (*4huq*) and one with the second lowest conformational change (*3x29*; 0.67Å), the *Blind* protocol does manage to generate models with more than the 30% of the native contacts. For the remaining cases, acceptable models are found only for *2gsk* with the first near-native model ranked at the 39th position with a l-RMSD of 9.47Å. The top1 and top10 performance are similar to what has been reported for HADDOCK using a blind docking scenario³⁸. Altogether, the results of the *Blind* predictions are considerable worse than that of our *Membrane* protocol in terms of both overall performance and CAPRI-based quality of the generated models. This clearly shows that the use of the membrane

topological information to drive the modeling process has a significant impact on the docking performance.

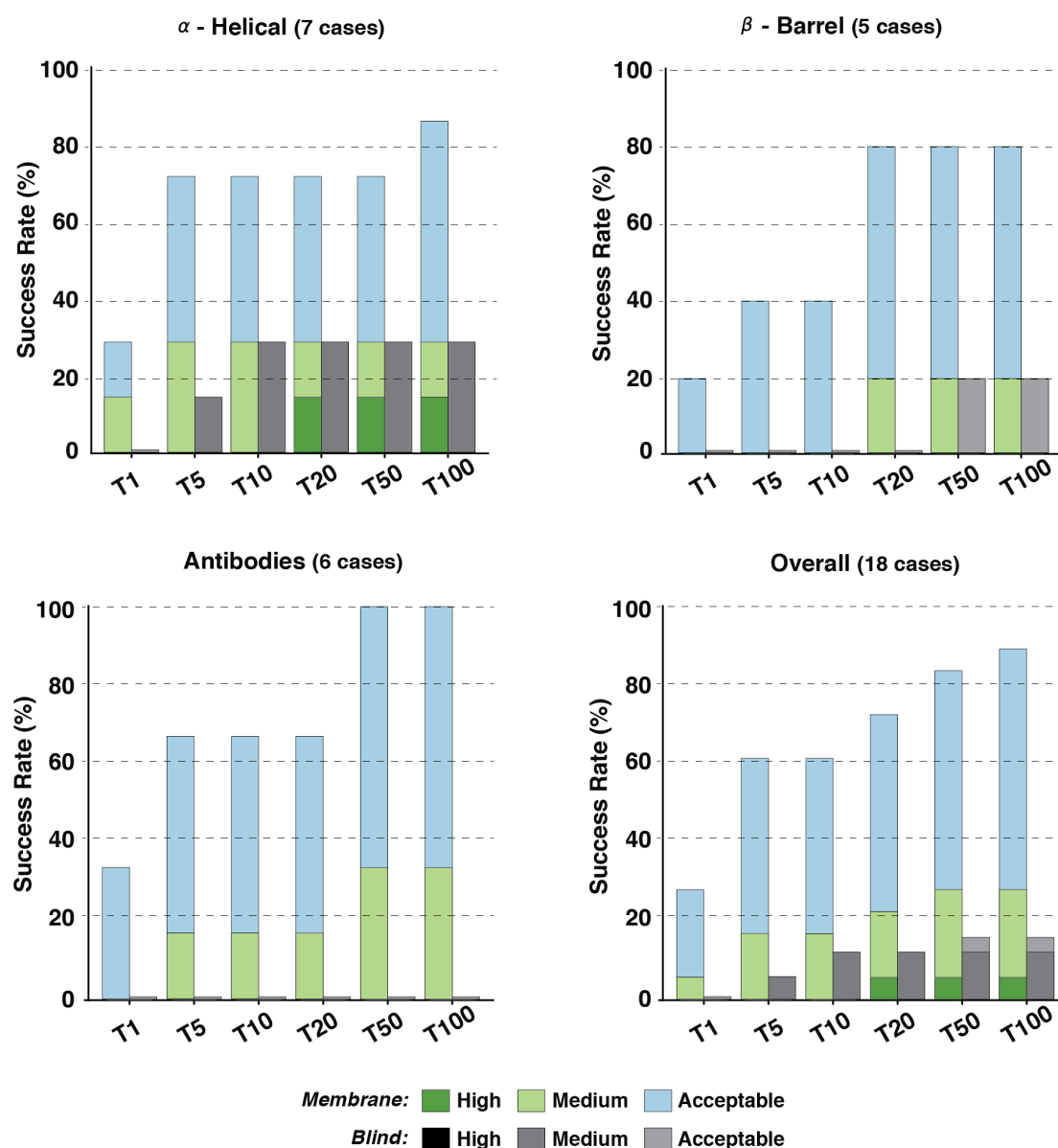


Fig. 3. Performance of the membrane protein integrative modeling protocol on the 18 cases of the membrane docking benchmark. Success rates are presented for each of the benchmark categories (α -Helical, β -Barrel and Antibodies) as well as Overall (including all three different categories). Color coding from blue to green (Membrane) and grayscale (Blind) indicates the model quality (from acceptable to high) as defined based on CAPRI criteria.

Impact of different membrane definitions on the docking performance

The results presented so far have been obtained by either defining the membrane based on the phosphate beads positions taken from MemProtDB (*Membrane*) or by fully blind predictions (i.e. without any membrane). We investigate here how different definitions of the membrane might impact our docking protocol. For that purpose, we have generated two additional artificial bead representations of the membrane based on the average (*Average*) or minimum (*Minimum*) *z-axis* coordinate provided by the equilibrated MemProtDB membrane model. We have compared the docking performance of those different membrane scenarios on the 18 cases from the membrane docking benchmark. As previously, we assess the performance in terms of success rate for each of the selected *N* tops (See Material and Methods; *Metrics for the evaluation of model quality and success rate*). For the sake of simplicity, we only report the success rate for acceptable or better models.

On average, in the *Membrane* scenario our simulations have 99 ± 34 starting swarms (ranging from 32 to 170), while for *Average* and *Minimum* this increases to 134 ± 28 and 164 ± 34 , respectively. This roughly translates into an increase of 7,000 and 13,000 in the number of *glowworms* (agents of the algorithm representing possible ligand poses) that are handled by the optimization algorithm during the sampling and scoring processes as compared to the *Membrane* scenario. As shown in Fig. 4A, for the two most representative tops (*T5* and *T10*) the success rates drop from 61.1% to 44.4% and 27.8% for *Average* and to 50% and 33.3% for *Minimum*. This pattern is observed along all selected top *N* models, which suggests that there is a negative correlation between the number of *swarms* (and *glowworms*) and the docking performance. This effect is

expected, since the larger the pool of generated poses, the larger the number of possible false positives which can be selected by the scoring function. For this reason, the optimization of the poses might not always converge towards biological relevant states.

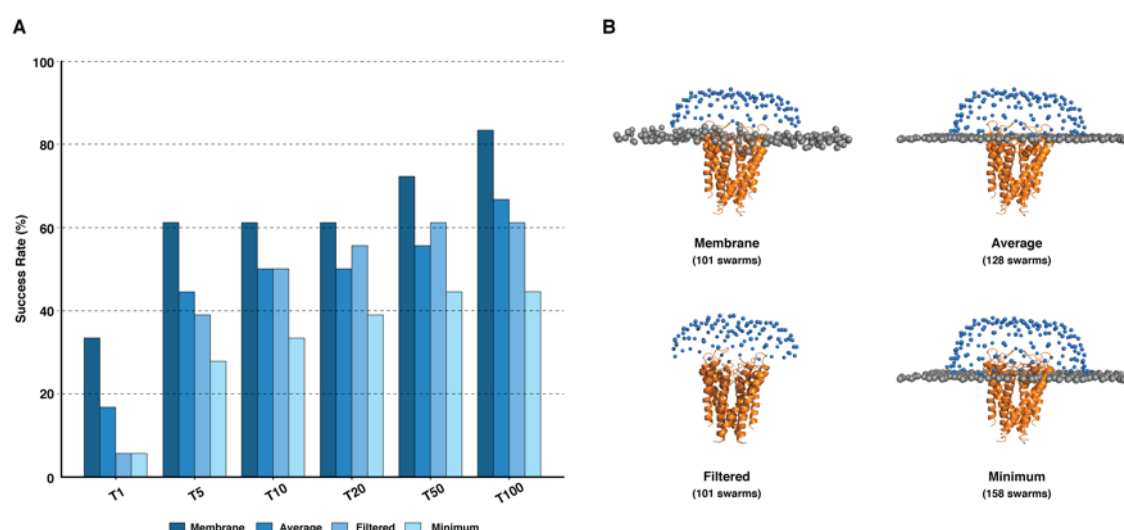


Fig. 4. Analysis of the impact of different membrane definitions onto the docking performance. **A)** Bar plot of the performance of the different membrane setups on the 18 cases of the membrane protein docking dataset (i.e. before refinement with HADDOCK). The success rate is defined as the percentage of cases for which an acceptable or higher quality model was found within the selected top N . **B)** Illustration of the different membrane setups on a representative case of the benchmark (1k4c).

Penalizing models penetrating the membrane leads to better predictions

We have also investigated the effect of the scoring penalty on the optimization algorithm during the docking. To do so, we have designed an additional scenario (*Filtered*), in which the membrane was only used to initially filter the swarms over the receptor surface, but not considered for penalizing models penetrating the membrane (see Fig. 4B). In this case, the success rate of the top 10 is similar to that of *Average* scenario (50%). However, for higher tops such as *T1* and *T5*, the *Filtered* scenario performs considerably worse as compared to *Membrane* (5.5% and 38.8% vs 33.3% and

61.1% respectively). This clearly suggests that, while the membrane plays an important role to narrow the conformational search it has also a big impact on the scoring: First, it guides the optimization protocol towards more binding-competent regions and second, it helps identifying near-native states out of the pool of generated docked models.

The structural quality of the docked models improves after HADDOCK refinement

We have assessed the quality of the docked models in terms of intermolecular steric clashes. To do so, we have quantified and compared the number of clashes (See Material and Methods; *Metrics for the determination of steric clashes in a protein complex*) present in our docked models before (LightDock only) and after refinement with HADDOCK. On average, the top 100 LightDock models have a significant number of clashes (28.5 ± 10.0) compared to those after refinement (0.6 ± 0.5) as shown in Fig. 5A. For some cases, *2bs2*, *2gsk*, *3csl* and *1ots*, few refined models ranked at positions ≥ 95 still have a moderate number of those (> 25), but these are penalized at the level of the HADDOCK score which ensures that clashing models will never be ranked at top ranking positions. Overall, this coarse-grained refinement protocol is able to refine and remove more than the 98% of the total number of clashes. As an example, a model before and after refinement is shown in Fig. 5B.

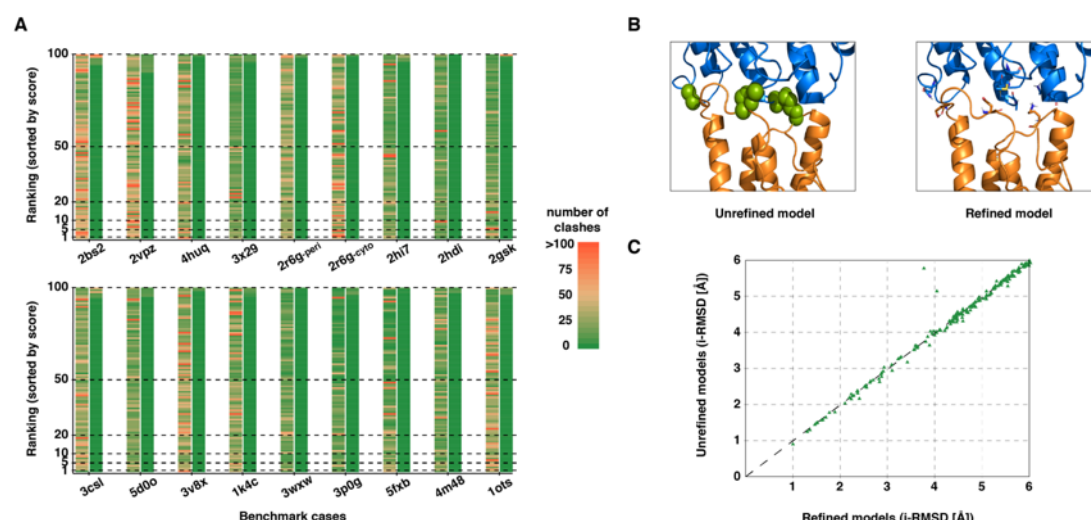


Fig. 5. Analysis of the quality of the membrane-associated protein models before and after refinement with HADDOCK. **A)** Stacked bar plot of the top 100 generated models for each of the benchmark cases (18 in total) ranked by their respective score (left – LightDock DFIRE docking score, right - HADDOCK score). For each complex the left bar corresponds to the unrefined models and the right bar to the refined models. The color coding (from green to red) indicates the number of clashes. **B)** Illustration of a complex before and after refinement. Green spheres represent atomic clashes. The corresponding side chains are shown as sticks in the refined model. **C)** i-RMSD comparison of all models with an i-RMSD $\leq 6\text{\AA}$ before and after refinement (183 in total). Points above the diagonal indicate an improvement in i-RMSD value.

Ideally, a refined complex should not structurally deviate too much from its unrefined counterpart. If this is not the case, the refined interface might significantly differ from the predicted one and therefore lose a relevant predicted state. We have investigated whether our refined models differ from their starting conformations in terms of their interface RMSD of the backbone (i-RMSD). For this, we selected all LightDock models with an i-RMSD $\leq 6\text{\AA}$ from the top 100 predictions for all cases (183 models in total) and compared them to their counterpart after refinement with HADDOCK. As shown in Fig. 5C, the vast majority of points are along the diagonal, which indicates that the backbone of the refined complexes has not significantly moved during the refinement. It is mainly the positions of side chains at the interface that have been

optimized (See Fig. 5B and Fig. S2). Points above the diagonal, indicate models that have improved in terms of i-RMSD after refinement. The changes are however limited. Two models (from *2vpz* and *3csl*), however, show a significant improvement of 1.09Å and 1.85Å respectively. In summary, these results show that our coarse-grained refinement protocol is very efficient in removing steric clashes without compromising the quality of the backbone conformation of near-native models.

Using membrane topological information to drive the docking performs better than post-sampling filtering approaches

We have also analyzed how our membrane-driven protocol compares to other state-of-the-art docking software. To this end, we selected ZDOCK²¹ as docking algorithm of reference for several reasons. First, it is a well-established docking program whose scoring protocol is being trained and continuously tested on a large and relatively heterogeneous benchmark of protein-protein complexes⁴³. Second, it allows to mask regions not belonging to the interface. And third, its standalone version (3.0.2) is a fast and easy-to-use tool for systematic benchmarking. Despite that the current version of ZDOCK does not allow to use an explicit representation of the membrane, we have designed three different scenarios in which various levels of information are used to include information about the membrane. In order to mimic our *Membrane* scenario, we have masked all surface accessible residues below the maximum z coordinate provided by our membrane implementation (ZDOCK-*max*). Similarly, to compare with our *Average* and *Minimum* scenarios, we have masked those residues below the average (ZDOCK-*avg*) or minimum (ZDOCK-*min*) z coordinate (Fig. S1). For the *Antibodies* subcategory, we have also masked all non CDR loops residues. Finally, we run the 18

cases of the membrane docking benchmark in fully blind (default) mode to define the baseline of ZDOCK.

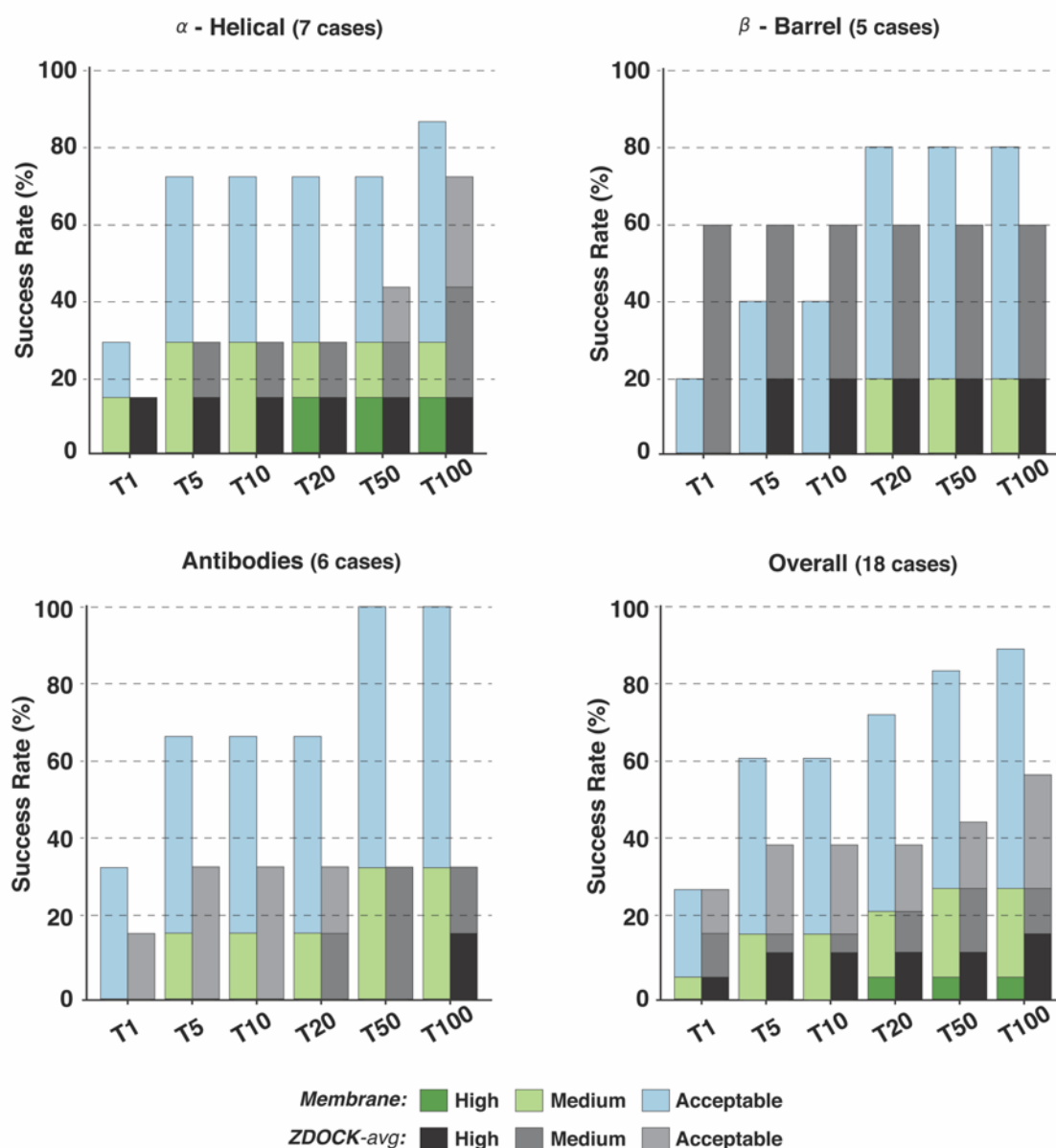


Fig. 6. Comparison of the performance of the membrane protein integrative modeling protocol (*Membrane*) with the best ZDOCK scenario (*ZDOCK-avg*) on the 18 cases of the membrane docking benchmark. Success rates are presented for each of the benchmark categories (α -Helical, β -Barrel and Antibodies) as well as *Overall* (including all three different categories). Color coding from blue to green (*Membrane*) and grayscale (*ZDOCK-avg*) indicates the model quality (from acceptable to high) according to CAPRI criteria.

The results for all those scenarios are shown in Fig. S3. The best performance is obtained for the ZDOCK-*avg* scenario. Comparing this scenario to our *Membrane* protocol shows that both protocols have an equivalent success rate of 27.7% for the top 1 model, but our protocol clearly performs best for *T5/T10* with 61.1% as compared to 38.8% for ZDOCK (Fig. 6). Our protocol reaches 88.8% of near-native models for the top 100 compared to the 55.5% for the best performing scenario in ZDOCK. These differences are more remarkable for α -*Helical* and *Antibodies* complexes with 71.4% and 66.6% for the top 5 (and top 10), respectively, compared to 28.5% and 33.3% for ZDOCK. In the case of β -*Barrel*, ZDOCK-*avg*, however, performs best with 60% in the top1 (3 out of 5) while our protocol starts at 20% for top 1 to reach a maximum of 80% in *T20*. Based on the well-established CAPRI quality criteria (See Material and Methods; *Metrics for the evaluation of model quality and success rate*), ZDOCK builds high quality models for the 16.6% of the tested cases (3 out of 18, while only one high quality model is obtained in our case) with 2 of them ranked within the top 10 predictions. These two cases correspond to the β -*Barrel* complex with the smallest conformational change (*2hdi*; i-RMSD of 0.361Å between unbound and bound structures) and a α -*Helical* bound case (*4huq*).

Discussion

In this work, we have developed and tested a new integrative modeling protocol to build membrane-associated protein assemblies. The protocol, which specifically accounts for the topological information encoded in the membrane, combines the capability of the LightDock framework to discard non-binding regions prior to the docking, with an efficient coarse-grained refinement via HADDOCK to remove clashes. As previously

demonstrated, including information during docking not only outperforms the scenario where data are only used to discard models (post-simulation approaches), but also reduces significantly the computational cost⁴², in this particular case by an average factor of 75% over the 18 complexes considered. Our *membrane*-driven protocol shows a much better performance in generating native-like structures for the vast majority of the tested cases than when the membrane is neglected (*Blind*). It achieved this by both filtering the initial *swarm* configurations and including a membrane penalty term into the scoring which helps both the optimization algorithm and the scoring of the docked models. Altogether, our findings reinforce the well-accepted notion that the integration of (experimental) data, in this case membrane topological data, into the docking calculations improves the performance of modeling approaches.

We have also investigated how different ways of defining the membrane topological information across the *z-axis* affect the sampling of the conformational space. Our protocol performs best when an equilibrated and simulated bilayer is incorporated into the sampling. This limits the number of *swarms* (and therefore *glowworms*), which, in turns, allows the optimization algorithm to identify more biological relevant states compared to less restricted scenarios such as *Average* and *Minimum*. This behavior is explained by the fact that in LightDock, sampling and scoring are closely interconnected since the optimization of the ligand poses (in rotational and translational spaces) is driven towards better scoring conformations. In other words, the reduction of potential false positives leads to an increase in the performance of the search algorithm.

We have analyzed the quality improvement of our docking predictions after refinement using a simple definition of steric clashes. We have shown that our

refinement protocol leads to the removal of (almost) all clashes while keeping the backbone conformation almost unaltered, with no more than 0.25Å i-RMSD for the most altered conformations (rare cases). As a consequence of the refinement, the side chains might suffer from bad conformations introduced by the removal of clashes and move away from the native conformation in the complex. To check this, we have analyzed the impact of the coarse-grained refinement on the side chain i-RMSD and the fraction of native intermolecular contacts they form. As shown in Fig. S2, the refined models do not significantly lose native intermolecular contacts as estimated by the Fnat metric and their side-chain i-RMSDs even slightly improve.

In this work, we have only focused on the modeling of membrane-associated protein assemblies. In cellular environments, however, some soluble proteins might associate with membranes in order to stabilize and/or carry out their function. These types of interactions have only been studied in a handful of systems such as signaling factors or nuclear receptors, due to the lack of more generic approaches that can be used to characterize a broader range of lipid-protein interactions⁴⁴. Our work could be extended to build realistic models of membrane-associated protein complexes. This would require extra effort to develop a scoring function that accounts for protein-lipid interactions. Such membrane-specific scoring functions have been already shown appropriate for membrane protein structure prediction and design purposes⁴⁵ and might also represent a significant advance for membrane-associated protein docking protocols. Looking ahead, a larger benchmark set will enable broader energy function development and optimization, which should eventually cover protein-lipid interactions too. In terms of software integration, LightDock, as a sampling algorithm, could be included within the future modular version of the HADDOCK software and eventually

offer an alternative to its default rigid-body sampling step. This would further extend HADDOCK modeling capabilities to account for the use of membrane-based bilayers. Note that HADDOCK has already been used with explicit membranes (nanodisks or micelles) to study the binding and orientation of proteins onto the lipid surface^{46–48}. These are however isolated cases and no systematic testing as performed here has yet been done.

In summary, we have developed an integrative modeling protocol for membrane-associated protein assemblies that accounts for the topological information provided by the membrane in the modeling process. It makes use of a membrane-derived bead bilayer during the sampling step with LightDock. Clashes resulting from the rigid-body docking are successfully removed by refinement with HADDOCK while preserving the quality of both backbone and side chains conformations at the interface. Importantly, while the present protocol only makes uses of the membrane to drive the modeling, it is fully compatible with the use of other sources of information such as mutagenesis and/or bioinformatic predictions in the form of residue restraints to further guide the docking.

Materials and Methods

Membrane docking dataset

We selected all complexes from the MemCplxDB database³⁸ (<https://github.com/haddocking/MemCplxDB>) whose interface lies between the membrane and either cytosolic or periplasmic regions. This selection yielded a dataset of 18 cases (See Fig. 2) which were further classified into:

- α -Helical: complexes whose receptor assemblies as a α -helical bundle.
- β -Barrel: complexes whose receptor forms an antiparallel β -sheet composed tandem of repeats.
- Antibodies: complexes whose soluble ligand is an antibody or nanobody.

Pre-processing of input structures

We make use of an equilibrated coarse-grained representation of the membrane to include topological information in our modeling procedure. For this, for each benchmark case, we obtain a representative coarse-grained snapshot of the transmembrane protein inserted into a simulated lipid bilayer (MARTINI representation⁴⁹) from the MemProtMD database (Fig.1 – *Step 1*)³⁹ (<http://memprotmd.bioch.ox.ac.uk/>). For the sake of simplicity and for saving computational resources, we remove all lipid beads except those representing the phosphate groups, which, to some extent, represent the most external layers. Then, we replace the coarse-grained TM receptors by their corresponding atomistic structure (Fig.1 – *Step 2*). When needed, we remove beads overlapping or clashing ($< 2.5\text{\AA}$ distance) with any heavy atom of the transmembrane protein once inserted into the membrane (*1ots*, *2gsk*, *2hi7*, *4m48*, and *3wxw*).

Implementation of a coarse-grained membrane in LightDock

To allow for the use of a coarse-grained membrane within the LightDock framework, we added new logic for the two different stages namely: The internal preparation of the molecules (at the *setup* level) and the actual simulation (at the scoring level). In the first stage, *setup*, we have added a new flag (*-membrane*) to activate the filtering of initial *swarms* (independent centers of simulation) according to the topological information of the membrane (no *swarms* will be generated below it). The protocol will detect the number of bead membrane layers provided by the user and select the upper one. For that purpose, it is expected that the user will provide the structure in PDB format and by a central plane point of view (the Z-axis is perpendicular to the membrane plane, that is the default view when saved by PyMol⁵⁰ for example). In case the lower layer is the target of interest, the system should be rotated by 180° around the X or Y-axis. During the simulation, we have included a term into the scoring scheme so that docking models in which the ligand penetrates the membrane are penalized and will be forced to optimize towards more favorable poses. In our case, we have defined a very unfavorable potential value for the membrane beads in the DFIRE scoring function used by LightDock (-999.0 - the more negative the value is, the worse becomes the score), in order to penalize models whose ligand's position is incompatible with the provided membrane model.

Running LightDock in membrane mode

LightDock execution consists of two steps: *setup* and *simulation*. In the first step, *setup*, the user provides to the *lightdock3_setup.py* command line tool the receptor and ligand structures in PDB file format, together with the number of *swarms*, *glowworms* per

swarm and other options such as removing hydrogen atoms and/or enabling ANM. In this new version of LightDock, a *-membrane* flag has been implemented in order to filter out *swarms* not compatible with the simulated coarse-grained membrane. For each of the filtered *swarms*, if residue restraints information is provided (as it is the case for the CDR loops for antibody-antigen complexes), this is used for pre-orienting the ligand poses as previously described⁴². In this work, the number of initial *swarms* used is 400 (default - many of them will be filtered by the membrane protocol) and the number of *glowworms* 200 (default). Hydrogen atoms are also removed as they are not supported by the DFIRE scoring function. When the setup step finishes, the docking simulation is ready to be started. A second command line tool, *lightdock3.py*, performs the simulation for the number of steps provided by the user (100 in this work) using the DFIRE scoring function and running in parallel depending on the number of cores specified. Once the simulation finishes successfully, predicted poses are generated (*lgd_generate_conformations.py*) and clustered (*lgd_cluster_bsas.py*) according to the default LightDock protocol. Finally, the *lgd_rank.py* command line tool generates a ranking of the top clustered predictions according to LightDock. An exhaustive tutorial of the different steps of the protocol can be accessed online at <https://lightdock.org/tutorials/membrane/>.

Coarse-grained refinement in HADDOCK

For the local installation, models must be converted into their coarse-grained representation. This is done via an *in-home* script included in the *CGtools* directory of the HADDOCK2.4 distribution as: “*python aa2cg.py model.pdb*”. As output, the script generates the MARTINI-based CG model (*model_cg.pdb*) as well as a restraints file in the

form of *model_cg_to_aa.tbl*, which includes the mapping of the generated coarse-grained beads to their corresponding atoms. The *atom-to-bead* restraints files of the different CG models must be combined into a single file (e.g. *cg-to-aa.tbl*) that will be used by HADDOCK to restore the atomistic resolution. In order to perform the refinement, a handful of parameters within the HADDOCK parameter file (*run.cns*) must be adapted as follows assuming that 100 models will be refined:

- *rotate180_it0=false* (to skip sampling 180° complementary interfaces)
- *crossdock=false* (to refine receptor – ligand from the structures provided)
- *rigidmini=false* (to skip *it0* stage)
- *randorien=false* (to skip *it0* stage)
- *rigidtrans=false* (to skip *it0* stage)
- *ntrials=1* (to skip *it0* stage)
- *structures_0=100* (for *it0* stage)
- *structures_1=100* (for *it1* stage; must always be \leq than *structures_0*)
- *anastruc_1=100* (for analysis purposes at *it1* stage)
- *waterrefine=100* (for *itw* stage; this is the number of final output models)
- *initiosteps=0* (to skip *it1* stage)
- *cool1_steps=0* (to skip *it1* stage)
- *cool2_steps=0* (to skip *it1* stage)
- *cool3_steps=0* (to skip *it1* stage)
- *dielec_0=cdie* (to switch a constant dielectric constant when CG is used)
- *dielec_1=cdie* (to switch a constant dielectric constant when CG is used)

For setting up the refinement on the HADDOCK2.4 webserver version, a tutorial can be found at http://www.bonvinlab.org/software/haddock2.4/tips/advanced_refinement/.

Note that on the server coarse-graining should be enabled under the *Input data* tab.

The refined models are scored and ranked according to the default HADDOCK score, which is a linear weighted combination of terms as:

$$HADDOCK_{score} = 1.0 * E_{vdw} + 0.2 * E_{elec} + 0.1 * E_{AIR} + 1.0 * E_{desolv}$$

where E_{vdw} and E_{elec} are the van der Waals and electrostatic energies terms calculated using a 12-6 Lennard-Jones and Coulomb potential, respectively, with OPLS nonbonded parameters, E_{AIR} is the ambiguous interaction restraints energy, E_{desolv} is an empirical desolvation score⁵¹. Note that in this protocol, since we are only refining the model and not providing any restraints, the E_{AIR} term is not contributing to the final score. An example of the HADDOCK parameter files to run the refinement (*run.param* and *run.cns*) can be found at:

https://github.com/lightdock/membrane_docking/tree/master/refinement/example

Metrics for the evaluation of model quality and success rate

The quality of the models is assessed according to the well-accepted CAPRI criteria⁵². Docking models are classified as high (***), medium (**) or low (*) quality according to their similarities with the native structure by calculating the interface and ligand root mean square deviations (i-RMSD and l-RMSD) and the fraction of native contacts (Fnat) as:

- High: $F_{nat} \geq 0.5$ and $i\text{-RMSD} \leq 1\text{\AA}$ or $l\text{-RMSD} \leq 1\text{\AA}$,
- Medium: $F_{nat} \geq 0.3$ and $1\text{\AA} < i\text{-RMSD} \leq 2$ or $1\text{\AA} < l\text{-RMSD} \leq 5\text{\AA}$,
- Acceptable: $F_{nat} \geq 0.1$ and $2\text{\AA} < i\text{-RMSD} \leq 4$ or $5\text{\AA} < l\text{-RMSD} \leq 10\text{\AA}$ and
- Incorrect: $F_{nat} < 0.1$ or $i\text{-RMSD} > 6\text{\AA}$ or $l\text{-RMSD} > 10\text{\AA}$.

The overall success rate is defined as the percentage of benchmark cases with at least one acceptable or better model within a given Top N ($N = 1, 5, 10, 20, 50, 100$).

Metrics for the determination of steric clashes

We define a steric clash as any heavy atom-heavy atom intermolecular contact shorter than 2.5Å (i.e. hydrogens excluded). Using this definition of clashes, we sought to investigate whether our coarse-grained refinement protocol in HADDOCK leads to higher quality structures (i.e. less absolute number of clashes) as compared to those generated from the docking step with LightDock. To do so, for each of the benchmarked cases we quantified and compared, on a per model basis, the number of clashes present on the top 100 docked models before and after refinement.

References

1. Allen, K. N., Entova, S., Ray, L. C. & Imperiali, B. Monotopic Membrane Proteins Join the Fold. *Trends in Biochemical Sciences* vol. 44 7–20 (2019).
2. Overington, J. P., Al-Lazikani, B. & Hopkins, A. L. How many drug targets are there? *Nat. Rev. Drug Discov.* **5**, 993–996 (2006).
3. Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schiöth, H. B. & Gloriam, D. E. Trends in GPCR drug discovery: New agents, targets and indications. *Nat. Rev. Drug Discov.* **16**, 829–842 (2017).
4. Munk, C. *et al.* An online resource for GPCR structure determination and analysis. *Nat. Methods* **16**, 151–162 (2019).
5. Shimizu, K., Cao, W., Saad, G., Shoji, M. & Terada, T. Comparative analysis of membrane protein structure databases. *Biochimica et Biophysica Acta - Biomembranes* vol. 1860 1077–1091 (2018).
6. Pinto, C. *et al.* Formation of the β -barrel assembly machinery complex in lipid bilayers as seen by solid-state NMR. *Nat. Commun.* **9**, (2018).

7. Birch, J. *et al.* The fine art of integral membrane protein crystallisation. *Methods* vol. 147 150–162 (2018).
8. Sirdeshmukh, R. Indian proteomics efforts and human proteome project. *Journal of Proteomics* vol. 127 147–151 (2015).
9. Berman, H. M. The Protein Data Bank / Biopython. *Presentation* **28**, 235–242 (2000).
10. Koehler Leman, J., Ulmschneider, M. B. & Gray, J. J. Computational modeling of membrane proteins. *Proteins Struct. Funct. Bioinforma.* **83**, 1–24 (2015).
11. Almeida, J. G., Preto, A. J., Koukos, P. I., Bonvin, A. M. J. J. & Moreira, I. S. Membrane proteins structures: A review on computational modeling tools. *Biochimica et Biophysica Acta - Biomembranes* vol. 1859 2021–2039 (2017).
12. Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292**, 195–202 (1999).
13. Viklund, H. & Elofsson, A. OCTOPUS: Improving topology prediction by two-track ANN-based preference scores and an extended topological grammar. *Bioinformatics* **24**, 1662–1668 (2008).
14. Kelm, S., Shi, J. & Deane, C. M. MEDELLER: Homology-based coordinate generation for membrane proteins. *Bioinformatics* **26**, 2833–2840 (2010).
15. Ebejer, J. P., Hill, J. R., Kelm, S., Shi, J. & Deane, C. M. Memoir: template-based structure prediction for membrane proteins. *Nucleic Acids Res.* **41**, (2013).
16. Šali, A. & Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779–815 (1993).

17. Katchalski-Katzir, E. *et al.* 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 (1992).
18. Tovchigrechko, A. & Vakser, I. A. Development and testing of an automated approach to protein docking. in *Proteins: Structure, Function and Genetics* vol. 60 296–301 (2005).
19. Comeau, S. R., Gatchell, D. W., Vajda, S. & Camacho, C. J. ClusPro: A fully automated algorithm for protein-protein docking. *Nucleic Acids Res.* **32**, (2004).
20. Cheng, T. M. K., Blundell, T. L. & Fernandez-Recio, J. PyDock: Electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins Struct. Funct. Genet.* **68**, 503–515 (2007).
21. Mintseris, J. *et al.* Integrating statistical pair potentials into protein complex prediction. *Proteins Struct. Funct. Genet.* **69**, 511–520 (2007).
22. Dominguez, C., Boelens, R. & Bonvin, A. M. J. J. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* **125**, 1731–1737 (2003).
23. Zacharias, M. Protein-protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Sci.* **12**, 1271–1282 (2003).
24. Gray, J. J. *et al.* Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J. Mol. Biol.* **331**, 281–299 (2003).
25. Moal, I. H. & Bates, P. A. SwarmDock and the use of normal modes in protein-protein Docking. *Int. J. Mol. Sci.* **11**, 3623–3648 (2010).

26. Jiménez-García, B. *et al.* LightDock: A new multi-scale approach to protein-protein docking. *Bioinformatics* **34**, 49–55 (2018).
27. Geng, C., Xue, L. C., Roel-Touris, J. & Bonvin, A. M. J. J. Finding the $\Delta\Delta G$ spot: Are predictors of binding affinity changes upon mutations in protein–protein interactions ready for it? *Wiley Interdisciplinary Reviews: Computational Molecular Science* vol. 9 (2019).
28. Lensink, M. F. & Wodak, S. J. Score_set: A CAPRI benchmark for scoring protein complexes. *Proteins Struct. Funct. Bioinforma.* **82**, 3163–3169 (2014).
29. Yu, J. & Guerois, R. PPI4Dock: Large scale assessment of the use of homology models in free docking over more than 1000 realistic targets. *Bioinformatics* **32**, 3760–3767 (2016).
30. Kundrotas, P. J. *et al.* Dockground: A comprehensive data resource for modeling of protein complexes. *Protein Sci.* **27**, 172–181 (2018).
31. Geng, C. *et al.* IScore: A novel graph kernel-based function for scoring protein-protein docking models. *Bioinformatics* **36**, 112–121 (2020).
32. Chaudhury, S. *et al.* Benchmarking and analysis of protein docking performance in Rosetta v3.2. *PLoS One* **6**, (2011).
33. Viswanath, S., Dominguez, L., Foster, L. S., Straub, J. E. & Elber, R. Extension of a protein docking algorithm to membranes and applications to amyloid precursor protein dimerization. *Proteins Struct. Funct. Bioinforma.* **83**, 2170–2185 (2015).
34. Hurwitz, N., Schneidman-Duhovny, Di. & Wolfson, H. J. Memdock: An α -helical membrane protein docking algorithm. *Bioinformatics* **32**, 2444–2450 (2016).
35. Alford, R. F. *et al.* An Integrated Framework Advancing Membrane Protein Modeling and Design. *PLoS Comput. Biol.* **11**, (2015).

36. Roel-Touris, J., Bonvin, A. M. J. J. & Jiménez-García, B. LightDock goes information-driven. *Bioinformatics* **36**, 950–952 (2020).
37. Roel-Touris, J., Don, C. G., Honorato, R. R., Rodrigues, J. P. G. L. M. & Bonvin, A. M. J. J. Less Is More: Coarse-Grained Integrative Modeling of Large Biomolecular Assemblies with HADDOCK. *J. Chem. Theory Comput.* **15**, 6358–6367 (2019).
38. Koukos, P. I., Faro, I., van Noort, C. W. & Bonvin, A. M. J. J. A Membrane Protein Complex Docking Benchmark. *J. Mol. Biol.* **430**, 5246–5256 (2018).
39. Stansfeld, P. J. *et al.* MemProtMD: Automated Insertion of Membrane Protein Structures into Explicit Lipid Membranes. *Structure* **7**, 1350–1361 (2015).
40. Zhou, H. & Zhou, Y. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.* **11**, 2714–2726 (2009).
41. Monticelli, L. *et al.* The MARTINI coarse-grained force field: Extension to proteins. *J. Chem. Theory Comput.* **4**, 819–834 (2008).
42. Roel-Touris, J., Bonvin, A. M. J. J. & Jiménez-García, B. LightDock goes information-driven. *Bioinformatics* **36**, 950–952 (2020).
43. Vreven, T. *et al.* Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. *J. Mol. Biol.* **427**, 3031–3041 (2015).
44. Saliba, A. E., Vonkova, I. & Gavin, A. C. The systematic analysis of protein-lipid interactions comes of age. *Nature Reviews Molecular Cell Biology* vol. 16 753–761 (2015).

45. Alford, R. F., Fleming, P. J., Fleming, K. G. & Gray, J. J. Protein Structure Prediction and Design in a Biologically Realistic Implicit Membrane. *Biophys. J.* **118**, 2042–2055 (2020).
46. Dancea, F., Kami, K. & Overduin, M. Lipid interaction networks of peripheral membrane proteins revealed by data-driven micelle docking. *Biophys. J.* **94**, 515–524 (2008).
47. Koppiseti, R. K. *et al.* Ambidextrous binding of cell and membrane bilayers by soluble matrix metalloproteinase-12. *Nat. Commun.* **5**, (2014).
48. Fang, Z. *et al.* Inhibition of K-RAS4B by a Unique Mechanism of Action: Stabilizing Membrane-Dependent Occlusion of the Effector-Binding Site. *Cell Chem. Biol.* **25**, 1327-1336.e4 (2018).
49. Monticelli, L. *et al.* The MARTINI coarse-grained force field: Extension to proteins. *J. Chem. Theory Comput.* **4**, 819–834 (2008).
50. DeLano, W. L. The PyMOL Molecular Graphics System. *Schrödinger LLC* *www.pymol.org* (2002) doi:citeulike-article-id:240061.
51. Fernández-Recio, J., Totrov, M. & Abagyan, R. Identification of protein-protein interaction sites from docking energy landscapes. *J. Mol. Biol.* **335**, 843–865 (2004).
52. Lensink, M. F. & Wodak, S. J. Docking and scoring protein interactions: CAPRI 2009. *Proteins Struct. Funct. Bioinforma.* **78**, 3073–3084 (2010).

Acknowledgements

General: The authors would like to thank all the Computational Structural Biology group at Utrecht University and specially Dr. Panagiotis Koukos for all the useful insights and fruitful discussions.

Funding: This work has been done with the financial support of the Dutch Foundation for Scientific Research (NWO) (TOP-PUNT grant 718.015.001) and the European Union Horizon 2020 projects BioExcel (675728, 823830) and EOSC-hub (777536).

Author contributions: JRT has designed the protocol, performed the experiments, analyzed the data and co-written this manuscript. BJG has co-designed the protocol, developed the new features required for the LightDock software and co-written this manuscript. AMJJB has co-designed the protocol, developed the HADDOCK required modules, supervised and directed the experiments and co-written this manuscript.

Competing interests: No conflict or competing interests declared.

Data and materials availability: The full HADDOCK version 2.4 is freely accessible as a web server at: <https://wenmr.science.uu.nl/haddock2.4> and as a local installation (see bonvinlab.org/software/haddock2.4). LightDock is open source software released under GPLv3 license, available at: <https://github.com/lightdock/lightdock/>. LightDock is also available at the Python Package Index Repository (PyPI) <https://pypi.org/project/lightdock/> and can be easily installed from the command line using pip. Code to reproduce results presented in this manuscript as well as the membrane-associated protein docked/refined models can be found at https://github.com/lightdock/membrane_docking. An online tutorial concerning

structure preparation and docking with LightDock is available at

<https://lightdock.org/tutorials/membrane>. Further reference and help on how to refine

models with the new Haddock2.4 server can be found at

http://www.bonvinlab.org/software/haddock2.4/tips/advanced_refinement/.

Supplementary information

The supplementary information includes a graphical representation of the different pseudo-membrane scenarios tested with the ZDOCK software as well the corresponding results in terms of success rate. Further, a comparison of unrefined and refined models is provided in terms of side chain i-RMSD and Fn_{at}.

Fig. S1. Graphical representation of the 1k4c receptor for the different tested scenarios in ZDOCK. The residues masked according to the maximum (ZDOCK-max), average (ZDOCK-avg) or minimum (ZDOCK-min) z-axis coordinate provided by the MemProtDB database are colored in dark gray. The orange regions, thus, represent those residues still allowed to contact with their counterpart.

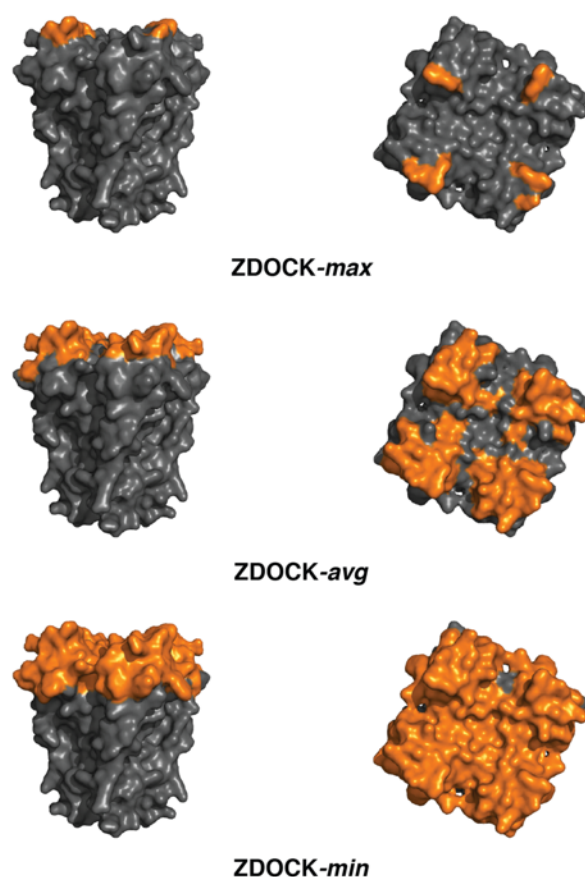


Fig. S2. Sidechain i-RMSD (**A**) and FNAT (**B**) comparison of all models (backbone) i-RMSD < 6Å before (y-axis) and after (x-axis) refinement (183 in total).

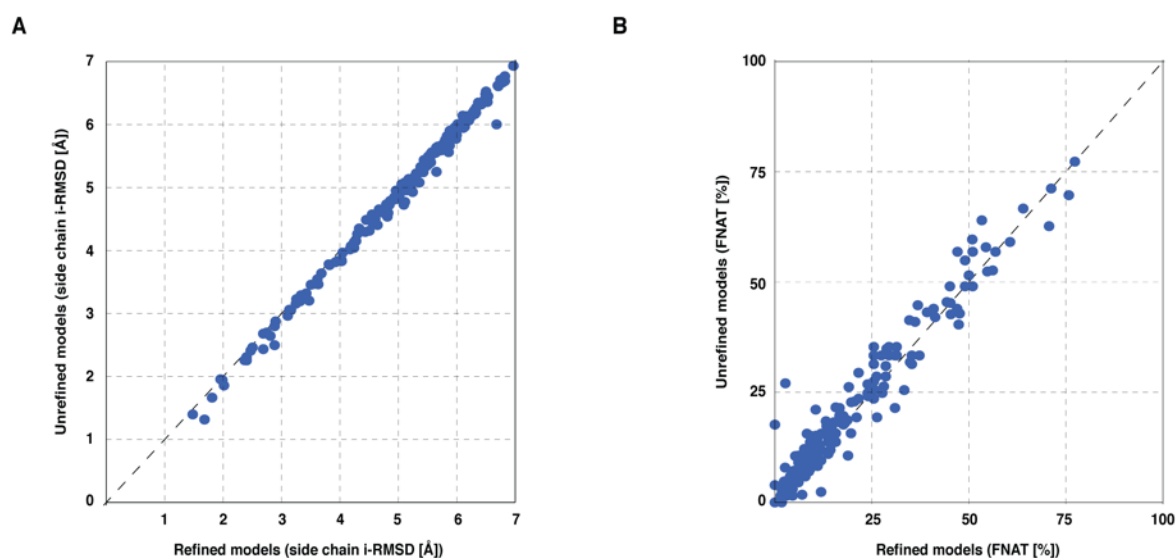


Fig. S3. ZDOCK success rate for the different tested (pseudo)membrane scenarios in the full dataset (18 cases).

