LUNDS UNIVERSITET

Benchmarking a memetic algorithm for global all-atom protein-protein docking with backbone flexibility

Vera Karlin

Bachelor's Degree Project in Biochemistry, 30 credits. Fall 2021 - Spring 2022.

Department: Department of Chemistry

Supervisors: Daniel Varela and Ingemar André

Abstract

Determining how proteins interact with each other to form complexes is very important for understanding both disease and cellular functions, but experimentally determining the structures of these complexes is both tedious and slow, which is why a great number of protein-protein docking algorithms have been developed to predict them. To this day, conformational changes in protein backbones have been one of the largest challenges when making docking predictions. The recently developed docking algorithm EvoDOCK aims to resolve this challenge by making use of a memetic algorithm that combines an evolutionary algorithm with Monte Carlo optimisations while also performing swaps of the backbone structures with conformer ensembles to simulate flexibility. In this thesis, a docking benchmark evaluating the performance of EvoDOCK against a standard Monte Carlo optimization based algorithm was constructed and performed along with evaluations of the algorithm's backbone flexibility strategy. The results showed an improvement of prediction quality for EvoDOCK as measured by iRMSD, DockQ and CAPRI for most of the benchmark complexes, with slightly better results when using a more exploratory set of evolutionary parameters. However, the predictions were more computationally costly than the standard method and only made efficient use of a small part of the backbone ensemble libraries, although showing clear room for optimisations and improvements of the methodology.

Acronyms

FFT	Fast Fourier Transform
MC	Monte Carlo
MCM	Monte Carlo + Minimisation
MDS	Motif Dock Score
EA	Evolutionary Algorithm
DE	Differential Evolution
NMA	Normal Mode Analysis
ACS	Adaptive Conformer Selection
RMSD	Root Mean Square Deviation
iRMSD	Interface Root Mean Square Deviation
CAPRI	Critical Assessment of Predicted Interactions
REU	Rosetta Energy Units

Table of contents

1 Introduction

The wide range of biochemical processes and functions carried out by cells are in most instances not performed by individual isolated proteins, but rather by multisubunit complexes made up of multiple interacting proteins (Alberts and Miake-Lye, 1992). Disturbances of the interactions forming these complexes caused by for example mutations can have disastrous effects on the functions carried out by them, giving rise to various diseases (Ideker and Sharan, 2008). This is one of the reasons why understanding how proteins interact and bind with each other becomes an objective of high priority not only for medicine, but for all of molecular biology (Lensink et al., 2020).

Charting these protein-protein interactions, referred to as the interactome, has been done experimentally but only to a limited extent due to the vast number of interactions to study and the limitations of the experimental methods (Kastritis and Bonvin, 2010). To circumvent these problems, a large number of algorithms have been developed to computationally predict how proteins would bind with each other to form complexes. However, the goal of performing accurate docking predictions is not an easily achievable one due to the massive amount of possible binding modes, each with their own unique set of electrostatic and intermolecular interactions giving rise to complex and rugged energy landscapes. On top of these interactions, docking algorithms also have to take into account the conformational changes made by the side chains and backbones of proteins during complex formation. Many docking algorithms fail to produce accurate docking models for complexes with highly flexible backbones, leaving the problem in urgent need of solutions (Lensink et al., 2017).

The recently developed protein docking algorithm EvoDOCK aims to take on the challenge of conformational flexibility through the use of side-chain packing and pre-generated backbone conformers. The algorithm combines the exploratory strengths of an evolutionary algorithm with the optimising capabilities of a Monte Carlo algorithm (Varela et al., 2022).

1.1 Aims of the study

The primary aim of this study is to benchmark the performance of EvoDOCK against one of the standard methods, RosettaDOCK, for a set of 10 flexible complexes categorised as difficult by the Docking Benchmark v5 (Vreven et al., 2015). The parameters used for the evolutionary component of the EvoDOCK algorithm are also to be tested, along with

evaluations of computational efficiency and precision, especially in terms of the flexible backbone component.

2 Background

2.1 Fundamental docking principles

Protein-protein docking is in its essence the process of algorithmically predicting the structure of a complex formed by two interacting proteins, either entirely based on their three-dimensional structure or with some additional knowledge of their binding site (Vakser I. A., 2014). These two docking categories are referred to as global and local docking, where global docking is generally more computationally expensive due to the much larger search space in comparison with the area specific local docking.

The solutions to the protein-protein docking problem consists of four essential components: a way of representing the proteins, an energy function to mathematically score the free energy of the interactions, a search algorithm making use of this energy function to make docking predictions and lastly a way of dealing with conformational flexibility. The conformational flexibility can be split into side-chain and backbone flexibility, with the latter only being used in what is referred to as unbound (or flexible) docking, in which the structural conformations of the proteins in their bound state is unknown. In contrast, bound (or rigid-body) docking uses the same conformations for the starting docking partners as the ones found in the native complex. This reduces its usability for many actual cases but still proves useful for the development and testing of methods intending to solve the docking problem.

In rigid-body docking the predictions exist as positions in a search space containing six degrees of freedom, three for describing the translational distance and three for describing the orientation of the ligand protein in relation to the statically located receptor protein. The same is true for flexible docking except with a much larger search space due to the complexity brought about by variations in side-chain and backbone conformations. Figure 1 depicts an example of rigid-body docking along with arrows representing each degree of freedom.

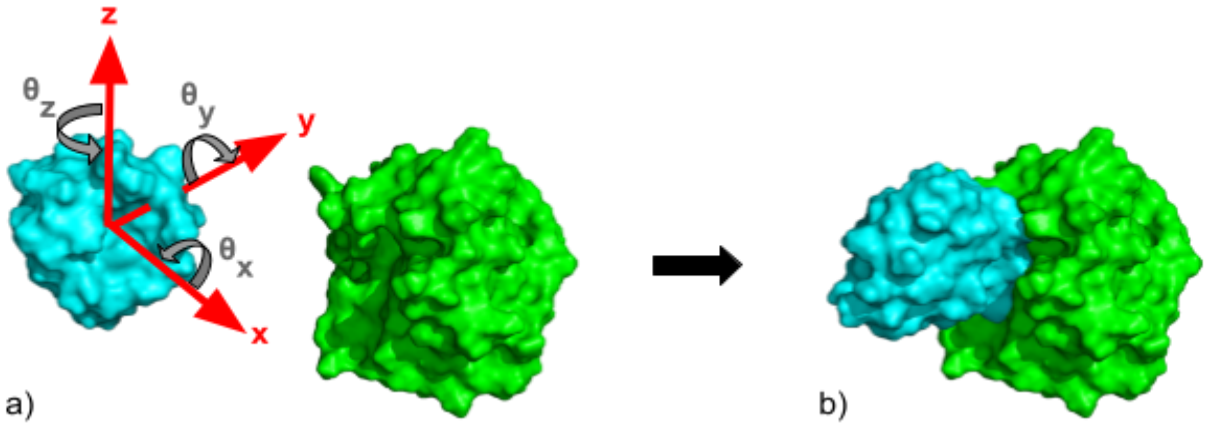

Figure 1: **Visual representation of the docking process.** Image a) depicts a possible starting position and orientation of the moving ligand protein (colored cyan) next to the static receptor protein (colored green) of the 1acb complex. The red arrows represent the three axes (x, y and z) of translational movement and the grey curved arrows represent the orientational axes (θ_x, θ_y and θ_z). Image b) depicts the experimentally determined native binding mode of the 1acb complex.

2.1.1 Protein representation

Full-atomic models are the most intuitive way of representing the proteins involved in the docking process, but simulating each individual atom comes with the drawback of often being computationally expensive, especially for large or flexible proteins. An alternative to full-atom modelling is using a low resolution coarse-grained model which represents groups of atoms with larger "pseudo-atoms" or beads (Roel-Touris et al., 2019). Such representations reduce the computational cost and also smoothen the energy landscape to avoid the search algorithm getting stuck at a local energy minima, although at the cost of reduced precision.

2.1.2 Energy functions

Docking algorithms move and reorient the protein models with the goal of reaching the lowest possible energy binding mode, which in theory should converge with the native (real-world) binding mode of the complex. The traversal of these often complex and jagged energy landscapes to find the global energy minima is guided by scoring functions; functions that estimate the free energy of a binding mode through predictions of various interactions. Such predictions can for example include force-field scoring based on non-bonded terms such as electrostatic and van der Waals potential or intermolecular factors such as hydrogen bonds, hydrophobicity and hydrophilicity of residues (Sunny & Jayaraj, 2022). Scoring can also make use of statistical knowledge of protein interactions, machine learning or combinations

of multiple different factors. Due to the nature of these energy functions, their applicability varies based on the model of representation and choice of search algorithm.

2.1.3 Search algorithms

There are many different search algorithms making use of energy functions to explore the search space, each with their own advantages and disadvantages. A notable approach to rigid-body docking that gained a lot of popularity after its introduction in the early 90s makes use of FFTs (fast Fourier transformations) to produce very computationally fast results through the use of correlation functions upon geometric molecule representations (Katchalski-Katzir et al. 1992).

Monte Carlo (MC) algorithms are another class of search algorithm used for protein-protein docking. These algorithms rely on repeated instances of random sampling to approach a solution, often making use of a Metropolis criteria. In those cases changes brought upon by random perturbations are kept if they lead to a decrease in energy, but even if the energy was not decreased there is still a chance that the change might still be kept based on a certain criteria referred to as a Metropolis criteria. The probability of that happening however gets exponentially smaller the higher the new energy is (Metropolis et al. 1953). In protein docking the randomness introduced consists of changes in orientation and relative position of the docking partners. MC search algorithms are by themselves fairly inefficient at exploring a very large search space such as during global docking and can often be considered more suited for local search. Their utility for more exploratory tasks can however be increased by for example performing a large number of searches using a coarse-grained protein representation and larger perturbations, as seen in the RosettaDOCK algorithm (Gray et al., 2003).

Population-based algorithms are a third type of search algorithm used in docking, one which actually excels at exploration of large search spaces (Sunny & Jayaraj, 2022). The name stems from the existence of multiple candidate solutions spread out over the search space simultaneously, leading to an expedited exploration of it. These population-based strategies, often referred to as metaheuristics are often inspired by nature. Examples of this include the particle swarm optimization used in SwarmDOCK (Li et al., 2010), and the many metaheuristics making use of Evolutionary Algorithms (EA), so called due to their inspiration from biological evolution and genetics. Through multiple generations, evolutionary algorithms iterate the evolutionary process of letting the individuals (candidate solutions) be

subject to random mutations, recombine with each other and then be selected as parents for the next generation based on their fitness (Coello Coello, 2005).

2.1.4 Conformational flexibility

Many techniques have been developed to tackle the problem of conformational flexibility during complex formation. Some of these techniques involve reducing the resolution of the docking, sacrificing precision for a smoother energy landscape and higher tolerance for structural inaccuracies and uncertainties (Vakser et al., 1999). Another approach consists of accounting for the conformational flexibility of the side-chains, for example through the use of backbone-dependent rotamer libraries (Wang et al., 2005). A backbone-dependent rotamer library contains statistically determined information about how frequencies and dihedral angles of side-chains vary based on the dihedral angles of the backbone (Gregorii et al., 2009). Optimisations of side-chain conformations, while very useful, leaves the challenging problem of backbone flexibility unresolved (Marze et al., 2018). A general approach to incorporating backbone flexibility into docking is detailed in section 2.4, along with how this method has been employed by the two docking algorithms benchmarked in this report: RosettaDOCK and EvoDOCK.

2.2 RosettaDOCK

The protein-protein docking algorithm RosettaDock was introduced in 2003 as a MC based method that expanded the Rosetta protein modelling software to the field of protein docking (Gray et al., 2003). The method uses a coarse-grained centroid stage followed by an all atom refinement stage which also optimises side chain conformations. Since then, multiple new versions and benchmarks have been made for the sake of improving the algorithm, most notably in terms of improvements to the scoring function (Chaudhury et al., 2011; Marze et al., 2017) and ability to accommodate for backbone flexibility (Chaudhury and Gray, 2008; Marze et al., 2018).

In spite of these changes, the underlying algorithmic framework remains mostly the same even up to the most recent version, RosettaDock 4.0 (Marze et al., 2018). Each of the trials start with the ligand protein at a random relative orientation and position around the larger receptor protein to create a glancing contact between them. As is shown in Figure 2, the algorithm follows with a 500-step low-resolution MC search using a coarse-grained representation and a scoring function referred to as Motif Dock Score (MDS). The MDS

function makes use of a pre-generated table of the lowest full-atom energies for all possible residue pair interactions and combines them with the translational and orientational distances from these ideal positions for each of the interacting side chains to create a score. The lowest energy structure is then subjected to 50-steps of high-resolution, full atom MCM (Monte Carlo + Minimization), which performs small random perturbations followed by orientational rigid-body energy minimizations, side-chain conformation optimizations using rotamer libraries and Metropolis criteria checks. The scoring function used during the high-resolution stage primarily takes into account Van der Waals attraction/repulsion, solvation, hydrogen bonding, statistical interactions of residue pairs, conformational energy of internal side chains and electrostatic interactions (Chaudhury et al., 2011).

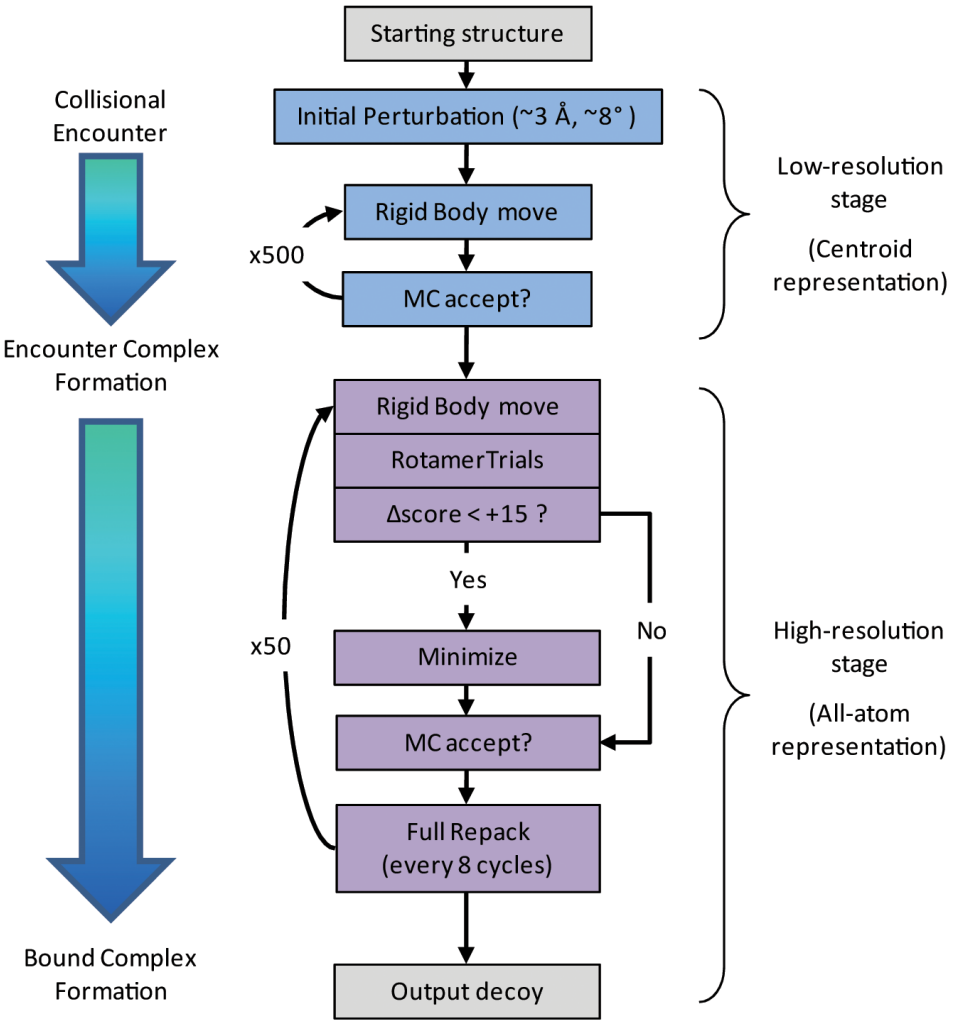

Figure 2: **Flowchart of the RosettaDock 3.2 algorithm's two stages** (Chaudhury et al., 2011).

2.3 EvoDOCK: A memetic docking algorithm

The memetic docking algorithm EvoDOCK also makes use of RosettaDOCKs full atom MCM steps to perform local searches, although in this case interspersed between global searches performed by an EA. Memetic algorithms are a class of algorithms which combine a global search using the explorative population-based approach of EAs with a local search algorithm that can exploit, i.e. optimise the solutions created during the global search (Moscato, 1989). The computational time saved by making use of an efficient EA allows for both of the global and local searches to be performed using full-atomic protein representations for an increased accuracy and level of detail.

The evolutionary component of EvoDOCK consists of a Differential Evolution (DE) algorithm; a population-based method which has gained a lot of popularity thanks to its ability to solve a wide range of complicated optimization problems. A defining feature of a DE algorithm is the population of solution candidates which are all represented as D-dimensional, real-valued vectors where each dimension of the vector corresponds to a dimension of the search space (Storn and Price, 1997). This representation is a great fit for a docking algorithm since each solution can be represented as a point in a 6-dimensional search space and thus as a 6-dimensional vector.

Figure 3 shows the docking methodology carried out by EvoDOCK. The initial population consists of 100 individuals, each existing as a vector representing a random position and orientation of the ligand protein around its static docking partner (Varela et al., 2022). For each of these individuals, three other candidate solutions: two random and the lowest scoring one, are selected as parents and are combined with the target individual at a magnitude controlled by the scaling factor F (usually $0 \leq F \leq 2$) in order to create an offspring. For each of the 6 docking degrees of freedom, a parameter of the target individual is replaced by the corresponding parameter of the mutated offspring at a rate determined by the crossover rate CR (usually $0 \leq CR \leq 1$). The following local search stage starts by sliding the moving molecule towards the static one until they come into contact. What follows is two cycles of MCM (as described in the 2.2 RosettaDOCK section) for the sake of resolving clashes and reaching a more energy optimised state in terms of rigid-body position, orientation and side chain conformation. The final stage of each generation involves comparing the fitness, i.e. the energy of the offspring solution with the original target solution using the full-atom scoring function used by RosettaDOCK. If the energy is lower, the offspring replaces the target

individual as a part of the population in the following generation. The best scoring model of the population is selected as an output and the process is repeated for 100 generations. The 100 independent docking trajectories performed during the benchmark detailed in this report therefore create a total of 10 000 predictions.

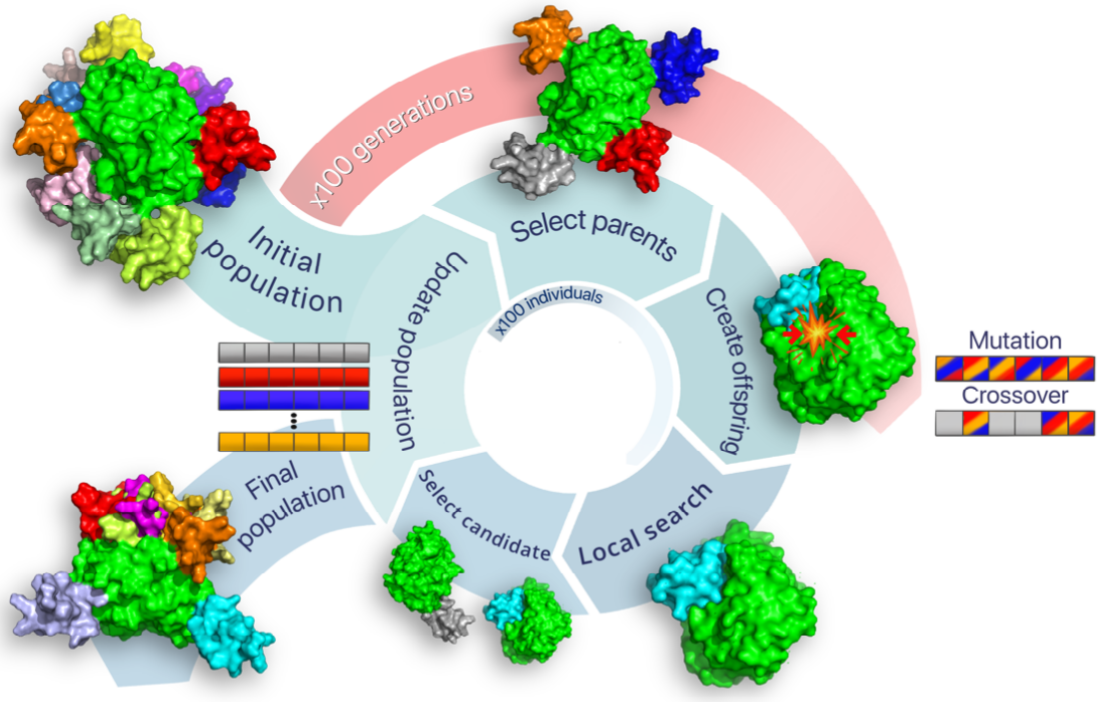

Figure 3: **Schematic representation of the EvoDOCK protein-protein docking methodology** (Varela et al., 2022). The figure shows the steps taken in each generation by the EvoDOCK algorithm along with the 3-dimensional structures of the participating protein models. The static receptor protein is colored green, the target candidate solution is grey, the other parents orange, blue and red, and the offspring is colored cyan.

2.4 Flexible backbones

The previously described RosettaDOCK and EvoDOCK algorithms both have modes allowing for backbone flexibility through the use of variations of a method which was originally developed for RosettaDOCK in 2008 (Chaudhury and Gray, 2008). The method makes use of a conformer library, referred to as a backbone ensemble, which aims to mimic the possible conformational changes made during complex formation. These ensembles are pre-generated from the unbound structures of the ligand and receptor and are then “swapped” in during the docking process in the hopes of scoring a lower energy and providing a better fit (Marze et al., 2018).

2.4.1 Backbone ensemble generation methods

The backbone ensembles used by both EvoDOCK and RosettaDOCK contain 100 conformers for each of the two docking partners, each generated using three different methods for the sake of providing diversity (Marze et al., 2018). Rosetta FastRelax protocol packs side-chains and minimises energy through many iterations with gradually increasing van der Waals repulsion, Normal Mode Analysis (NMA) combines the FastRelax protocol with movements along normal modes and Rosetta Backrub protocol packs side-chains while performing backrub moves that involve rotating segments of the protein backbone around certain pivot points. Figure 4 shows examples of what the structures of these backbone ensembles look like for two of the benchmark complexes.

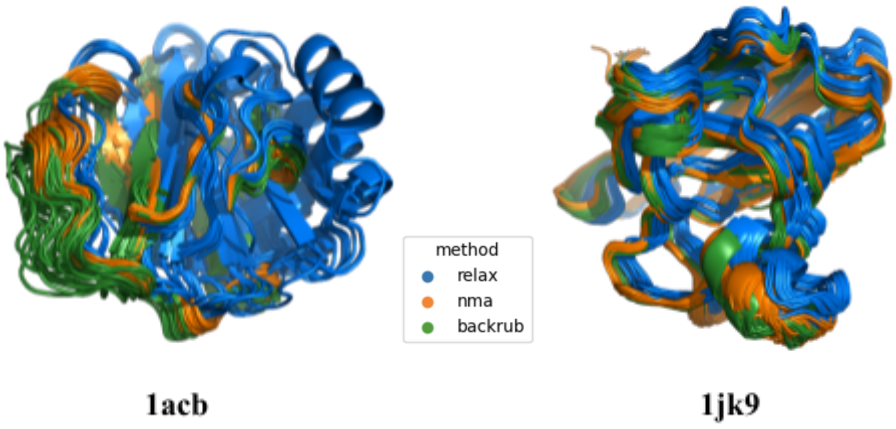

Figure 4: **Example of ligand protein structures used in the backbone ensembles.** The figure displays the structures of the 100 ligand ensemble conformers created for each of the two complexes with PDB (Protein Data Bank) ids of 1acb and 1jk9. The conformers are colored based on their method of generation: blue for the 30 Relax conformers, orange for the 40 NMA conformers and green for the 30 Backrub conformers.

2.4.2 RosettaDOCK with backbone flexibility

The backbone conformation swaps in RosettaDock are performed repeatedly during the coarse-grained centroid stage using a method referred to as Adaptive Conformer Selection (ACS). ACS aims to save computational cost by increasing the probability for conformer swaps when the acceptance rate is lower than 30% and raising it when it is above (Marze et al., 2018).

2.4.3 EvoDOCK with backbone flexibility

EvoDOCK performs backbone swaps for the ligand and receptor in a similar way as RosettaDOCK, but does it at the end of each generation as determined by a static "swap probability" (Varela et al., 2022). The swap with the ensemble is followed by a local search and a selection step where the old backbone is replaced by the new one if the energy has been improved.

2.5 Quality measures of docking predictions

Prediction quality measurements are an absolutely crucial part of the development of docking algorithms. There would not be much of a point in developing a method blindly without any knowledge of how its predictions compare to the experimentally determined real-world results, which is the main objective of benchmarking. Thankfully, many different measures exist for this exact purpose. Root-mean-square deviation (RMSD) is a common way to measure the average distance between the coordinates of each, or often a subset, of the atoms in two molecular structures (Zemla et al. 1999). In protein docking these structures consist of the predicted complex and the native one, with the distances often measured in angstrom (Å). Interface RMSD (iRMSD) is another popular alternative that only compares the atoms located in the proteins' interface residues, i.e. the interacting residues. RMSD can also be useful for measuring the magnitude of conformational changes, such as the ones made during backbone ensemble generation.

The Critical Assessment of Predicted Interactions (CAPRI) community has played an important role in the development of docking algorithms and methods for measuring their quality (Lensink and Wodak, 2013). Through the use of three different quality measures they rate the quality of docking models in four categories: Incorrect, Acceptable, Medium or High quality. The quality measure DockQ has later on expanded this to a score in the range [0,1], allowing for better comparisons between docking algorithms (Basu and Wallner, 2016).

Deciding which predictions to assess the quality of is guided by their energies as determined by the scoring function since that is the only known metric after performing a docking simulation. A simple way of assessment can be to check the iRMSD of the solution with the lowest energy; another to compare the best CAPRI or DockQ scores out of the 100 lowest energy solutions.

3 Results

3.1 Setup of experiment

The benchmark experiment as summarised in Figure 5 consisted of the following steps: Selection and preparation of protein complex structure files in the PDB (Protein Data Bank) format, generation of conformer ensembles, packing of side chains (prepacking) and running the EvoDOCK and RosettaDOCK docking protocols. A more technical description of the methods along with the command lines and scripts used can be found in the method section (section 5).

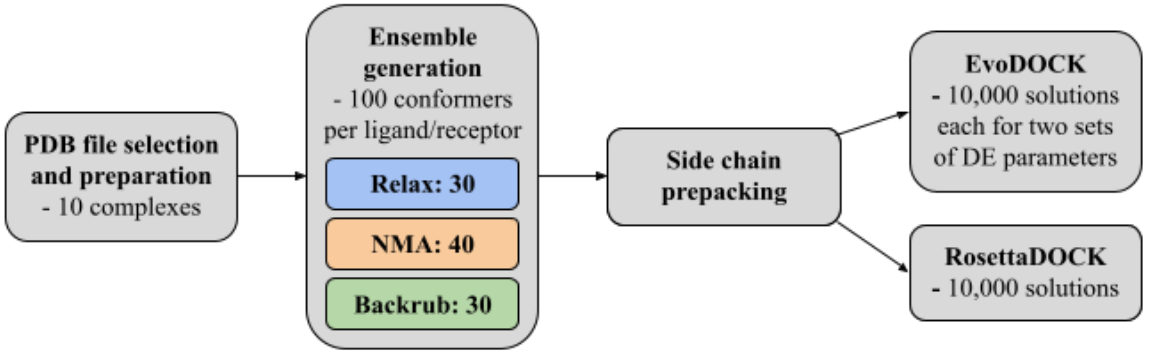

Figure 5: **Flowchart of the methodology used in the benchmark experiment.**

3.1.1 Selection of protein complexes

The 10 protein complexes chosen for the experiment were selected from the complexes used in the RosettaDOCK 4.0 benchmark (Marze et al., 2018) that were classified as difficult by the Docking Benchmark v5 (Vreven et al., 2015). They were selected based on length, lowest estimated computational time for ensemble generation and minimal difference in residues between their unbound and bound state PDB files (the PDB structure files did in many cases contain missing residues and even whole missing segments due to imperfections of experimental methods for structure determination). The complexes used in the experiment (shown in Figure 6) had the following PDB names in the protein data bank: 3f1p, 1r8s, 1eer, 2ido, 1pxv, 1fq1, 1acb, 1jk9, 1rke and 1f6m.

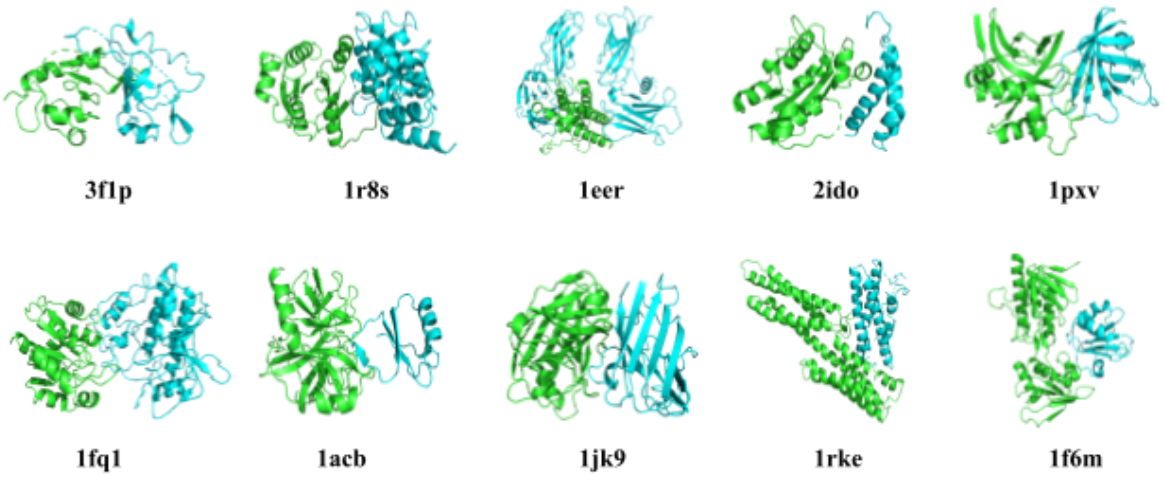

Figure 6: **Benchmark protein complex structures.** The structures of the 10 protein complexes used in the benchmark labelled by their PDB id. The receptor proteins are colored green and the ligand proteins cyan.

3.1.2 Preparation of PDB structure files

A great number of crashes occurred when attempting to run the ensemble generation, prepack and docking protocols. These crashes were attributed to a number of factors in the PDB files such as missing residues between the bound and unbound proteins, small molecule ligands, multiple chain names and inconsistent residue numbering. These issues were all resolved through the use of a pipeline script detailed in method section 5.1.

3.1.3 Ensemble generation and prepacking

For each ligand and receptor of the 10 protein complexes, 100 backbone conformers were generated to form an ensemble: 30 using the relax protocol, 40 using NMA method and 30 using the backrub protocol. The side-chains on the generated conformers and the unbound and bound versions of the complexes were packed to reduce energy in preparation of docking.

3.1.4 Running the docking protocols

The EvoDOCK algorithm performed unbound, global docking using two sets of DE parameters: a mutation rate (F) of 0.3 and crossover probability (CR) of 0.9 for a more exploitative search, and a F of 0.9 and CR of 0.3 for a more explorative search. Each of these two runs created 100 trajectories with a population size of 100 and ran for 100 generations, creating a total of 10 000 docking predictions per run. The same number of unbound global docking predictions were also created using the RosettaDOCK algorithm.

3.2 Main benchmark results

The results of the docking experiment are visually summarised in the scatter plots of Figure 8, where the interface RMSD difference between the native and predicted models are plotted against their score in Rosetta energy units (REU). It can be seen in the zoomed-out scatter plots that the models produced by EvoDOCK often have a significantly smaller range of both iRMSD and especially energy in comparison with RosettaDOCK. The RosettaDOCK results are spread out along the y-axis for most of the benchmark complexes with energies stretching into the positives, signifying a struggle in finding energy minima for many of the independent runs. EvoDOCK on the other hand can be seen exploiting the search space around the best scoring result and with the exception of the 3f1p complex, always scoring under -500 REU and in some cases under -1000 REU. It can also be noted that the plots for both of the docking algorithms show a "wall"-like shape, with the vast majority of solutions having an iRMSD over a certain value, which often lies between 2 Å and 6 Å, although regularly higher for the RosettaDOCK results. In the zoomed in plots of Figure 8, EvoDOCK can on many occasions be observed finding and honing in on energy minima with improved iRMSD values that RosettaDOCK was unable to find. It can however struggle with discerning between these minima, giving rise to the multiminima shapes apparent in the 1f6m, 1pxv and 2ido plots. In contrast, the plots for the complexes 1eer, 1rke and especially 1jk9 display a more desirable funnel-like shape, meaning a repeated decrease in energy correlating with a lowered iRMSD. The 1jk9 complex is by far the most successful one of the benchmarks with its two very clean funnel shapes, each with a low iRMSD solution at the bottom. The success of the predictions is reflected in the striking similarities that can be observed when aligning the structure with the native one (Figure 7). Furthermore, the distribution of the final lowest energy individuals of each EvoDOCK trajectory (shown as blue dots in Figure 8) show an often high concentration of predictions around certain energy minima, showcasing the algorithm's ability to make predictions with a consistent quality.

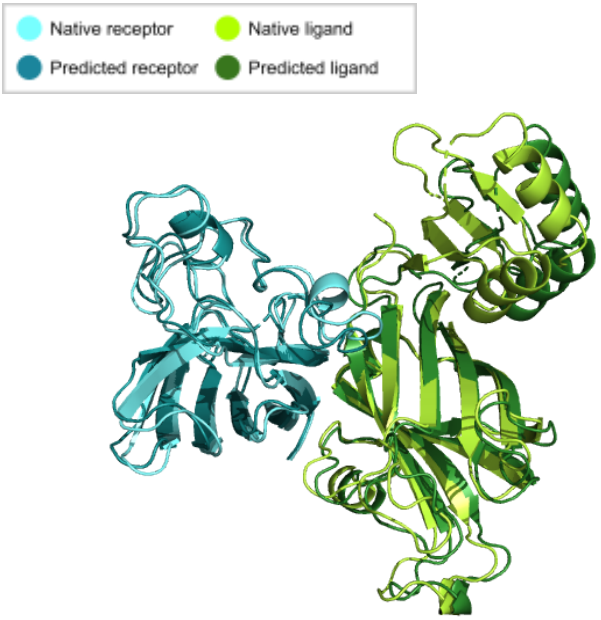

Figure 7: **Best docking prediction made by EvoDOCK for the 1jk9 complex**. The lowest energy model obtained by EvoDOCK for the 1jk9 complex (dark cyan for ligand and dark green for receptor) aligned with its native structure (light cyan for ligand and light green for receptor).

The DockQ, iRMSD and CAPRI prediction quality data displayed in Table 1 confirm the previous observations, with EvoDOCK giving a better DockQ score than RosettaDOCK for 8 out of 10 benchmark complexes. Despite these improvements, the CAPRI score of the benchmark shows that the scores produced are not very high for either of the methods, with most complexes only having an acceptable quality, two (or one for RosettaDOCK) with medium quality and one classified as being incorrect. This is however to be expected due to the very large search space of flexible docking.

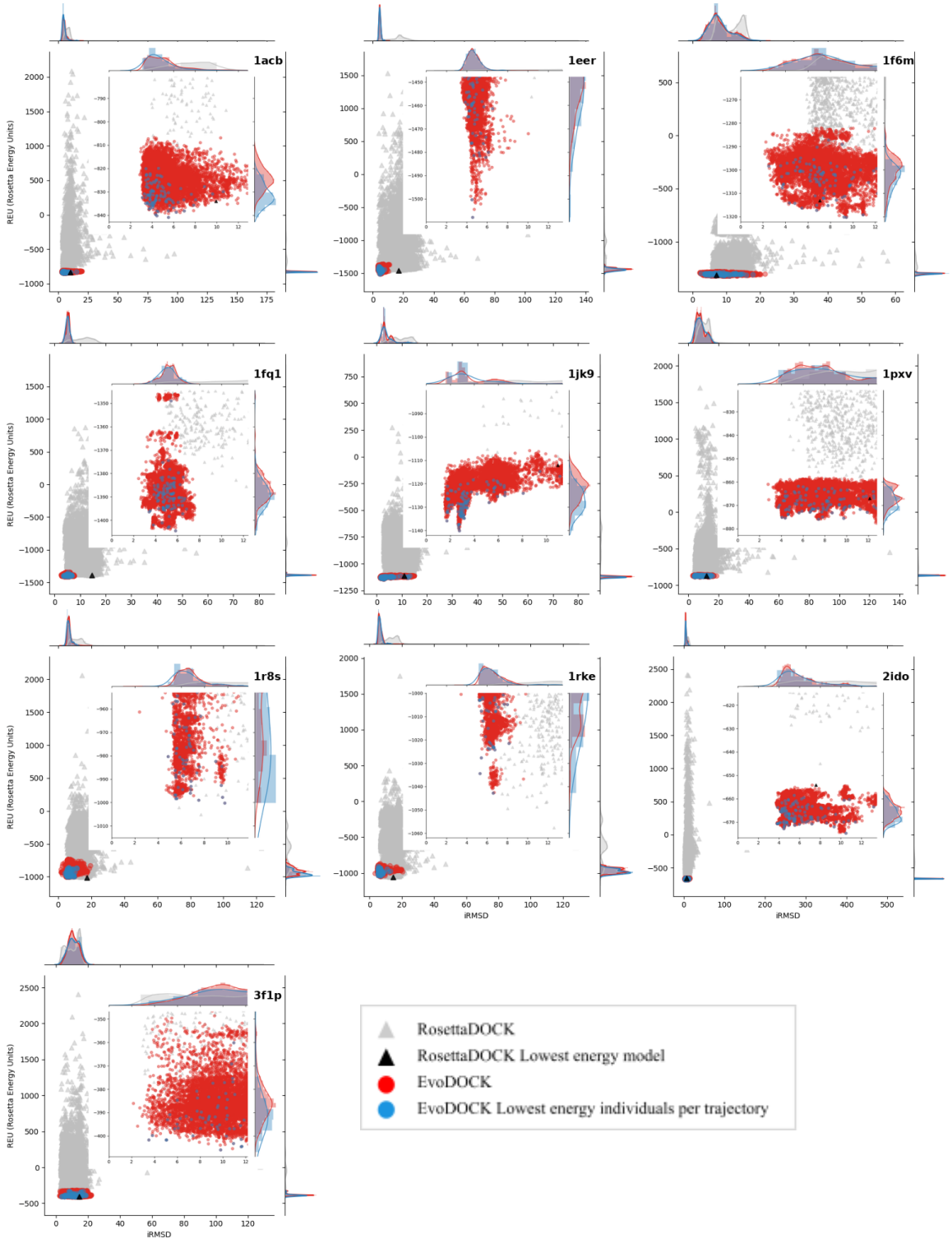

Figure 8: **Benchmark results using exploitative differential evolution parameters for EvoDOCK.** For each of the 10 benchmark complexes, the 10,000 predicted models created by EvoDOCK (shown as red dots) and RosettaDOCK (shown as grey triangles) had their iRMSD in Å (x-axis) plotted against their score in Rosetta energy units (REU, y-axis) with a zoomed in version of the plot in the upper left corner. The lowest energy individual of each EvoDOCK trajectory is colored blue and the lowest energy model produced by RosettaDOCK is colored black. The parameters used for the DE component of EvoDOCK during the benchmark were F = 0.3 and CR = 0.9.

PDB id	EvoDOCK (F = 0.3, CR = 0.9)			RosettaDOCK		
	DockQ	iRMSD	CAPRI	DockQ	iRMSD	CAPRI
3f1p	0.55	2.76	Medium	0.48	2.61	Acceptable
1r8s	0.19	5.41	Incorrect	0.05	7.93	Incorrect
1eer	0.43	2.67	Acceptable	0.33	3.37	Acceptable
2ido	0.29	4.75	Acceptable	0.39	3.63	Acceptable
1pxv	0.24	4.48	Acceptable	0.36	3.69	Acceptable
1fq1	0.48	3.32	Acceptable	0.43	3.54	Acceptable
1acb	0.45	2.66	Acceptable	0.40	2.98	Acceptable
1jk9	0.58	3.04	Medium	0.54	3.05	Medium
1rke	0.36	5.41	Acceptable	0.31	5.51	Acceptable
1f6m	0.27	5.99	Acceptable	0.24	6.12	Acceptable

Table 1: **Quality assessment of EvoDOCK and RosettaDOCK results.** Column 2-4 shows the DockQ score, lowest iRMS and CAPRI grading for the lowest energy models produced by the 100 EvoDOCK (with F = 0.3 and CR = 0.9) trajectories for each of the benchmark complexes in Column 1. Column 5-7 shows the same metrics for the 100 lowest energy models produced by RosettaDOCK. The highest DockQ score per complex is colored red.

3.3 Evaluation of differential evolution parameters

The scatter plots displaying the results of using a more exploratory set of DE parameters (Figure 9) show an increased scattering compared to the plots in Figure 8, most likely due to the more significant changes brought about by mutations. This can especially be seen in the 1jk9 complex having a more diffuse and blurry funnel shape when compared with the results when using the previous parameters. There have however been some noticeable improvements for the 1rke and 1r8s complexes as seen in their plots, showing even more funnel-like shapes. Apart from those cases, the plots tell a similar story as the previous ones, with similar looking funnels and multiminima.

This is further reinforced by the data in Table 2 showing very similar energy and DockQ scores across all complexes, with a difference in DockQ score of only ± 0.03 for all complexes except for 1jk9 having a 0.23 points higher score (40% increase) when using the exploratory method. The table also shows a better ability to achieve a lower iRMSD of its lowest energy solution in 8 out of 10 complexes, in some cases even significantly lower such as for the 1pxv complex with a change of almost 10 Å.

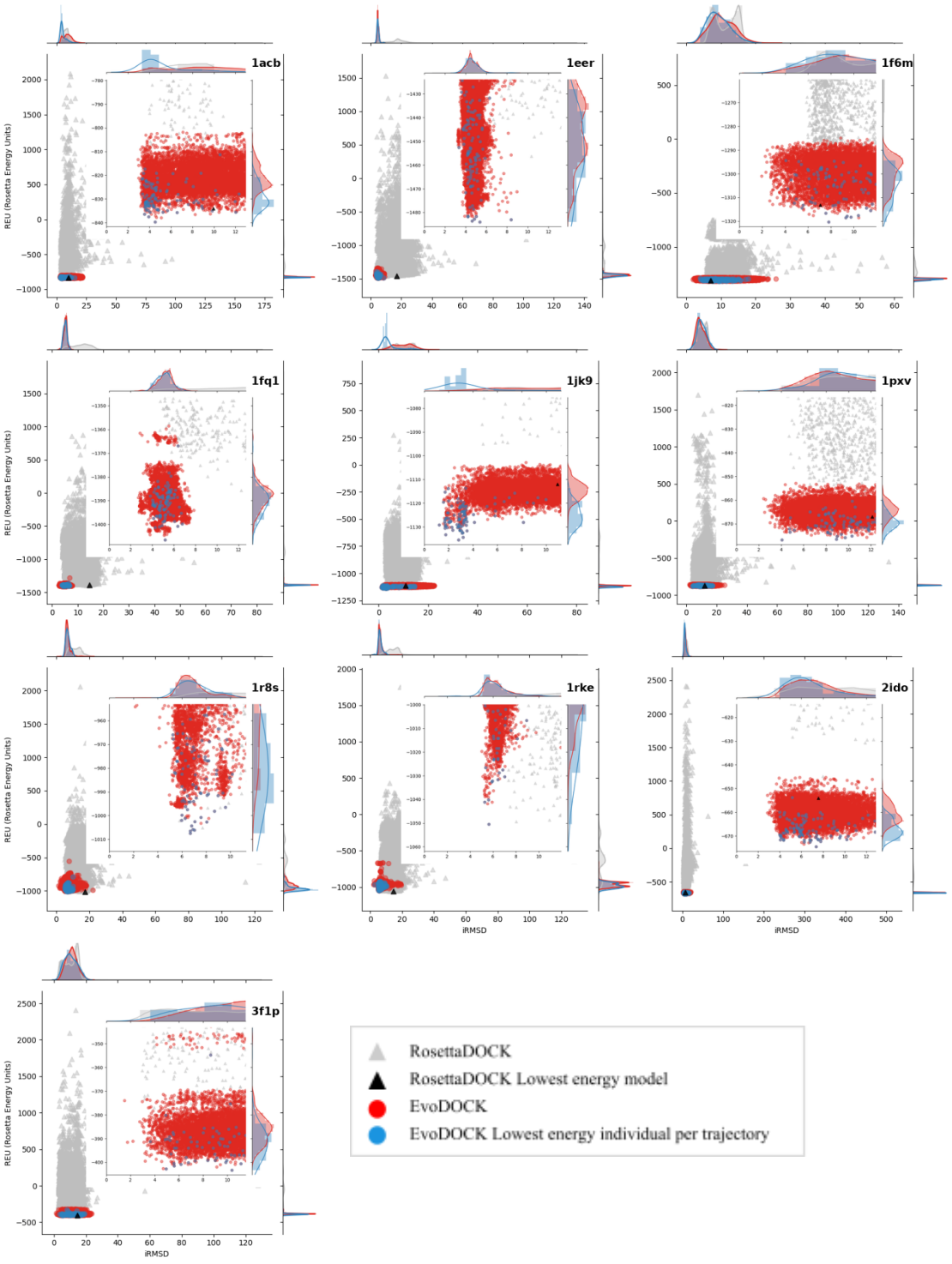

Figure 9: **Benchmark results using more explorative differential evolution parameters for EvoDOCK.** For each of the 10 benchmark complexes, the 10,000 predicted models created by EvoDOCK (shown as red dots) and RosettaDOCK (shown as grey triangles) had their iRMSD in Å (x-axis) plotted against their score in Rosetta energy units (REU, y-axis) with a zoomed in version of the plot in the upper left corner. The lowest energy individual of each EvoDOCK trajectory is colored blue and the lowest energy model produced by RosettaDOCK is colored black. The parameters used for the DE component of EvoDOCK during the benchmark were F = 0.9 and CR = 0.3.

PDB id	**EvoDOCK** (F = 0.3, CR = 0.9)		**EvoDOCK** (F = 0.9, CR = 0.3)	
	DockQ	**min score (iRMSD)**	**DockQ**	**min score (iRMSD)**
3f1p	**0.55**	**-406.8** (12.8)	**0.52**	-403.3 (**10.4**)
1r8s	**0.19**	-1000 (9.76)	**0.18**	**-1008 (6.7)**
1eer	**0.43**	**-1508 (4.58)**	**0.45**	-1484 (5.36)
2ido	**0.29**	**-674.7** (10.5)	**0.27**	-674.5 (**7.35**)
1pxv	**0.24**	**-880.8** (14)	**0.21**	-876.7 (**4.07**)
1fq1	**0.48**	-1404 (5.82)	**0.47**	**-1406 (5.18)**
1acb	**0.45**	**-840.8** (5.53)	**0.46**	-839.7 (**4.54**)
1jk9	**0.58**	**-1140 (2.75)**	**0.81**	-1135 (2.82)
1rke	**0.36**	-1043 (6.6)	**0.35**	**-1050 (5.63)**
1f6m	**0.27**	-1320 (9.4)	**0.28**	-1320 (**7.8**)

Table 2: **Quality assessment of using the two sets of differential evolution parameters.** Table comparing the results of the two sets of DE parameters for each protein complex in Column 1. Column 2 and 4 show the DockQ score, while Column 3 and 5 show the lowest energy solution achieved by each method together with the iRMSD of that solution in parenthesis. The lowest DockQ score per complex is colored red.

3.4 Computational time observations

The total computational time taken by each of the two algorithms to create 10,000 docking models for each of the benchmark complexes is shown in Table 3, where it becomes apparent that EvoDOCK in most cases takes about 2-3 times the computational time RosettaDOCK does. This is not a massive difference but it is still reason enough to try optimising the algorithm. The difference in time was hypothesised as being a result of EvoDOCK performing more frequent backbone sampling and scoring than RosettaDOCK, motivating a small change in the EvoDOCK code to reduce the probability of sampling new backbones to 30% from 100%. The 1eer complex had the highest computational time of all complexes at 2309 hours and was therefore selected for improvement. The impact in precision of using the modified algorithm on the 1eer complex (Figure 10) was not very large, with lowest energy solutions having about the same iRMSD as before and the largest discernable difference being a higher upper limit for the energy range. The computational time showed a 20,4% improvement as a result of the change, going from 2309 to 1837 hours for 10,000 models.

PDB id	RosettaDOCK	EvoDOCK (F = 0.3, CR = 0.9)		EvoDOCK (F = 0.9, CR = 0.3)	
	time (hours)	time (hours)	ratio	time (hours)	ratio
3F1P	251.39	770.78	3.1 : 1	611.28	2.4 : 1
1R8S	566.67	1081.5	1.9 : 1	1061	1.9 : 1
1EER	724.24	2309.1	3.2 : 1	1924.8	2.7 : 1
2IDO	173.33	491.65	2.8 : 1	531.66	3.1 : 1
1PXV	276.35	628.29	2.3 : 1	639.28	2.3 : 1
1FQ1	616.47	1205.3	2 : 1	1209.1	2 : 1
1ACB	354.22	686.77	1.9 : 1	700.49	2 : 1
1JK9	360.2	1059.4	2.9 : 1	866.7	2.4 : 1
1RKE	421.64	1325.6	3.1 : 1	1326.9	3.1 : 1
1F6M	503.16	1020.7	2 : 1	938.67	1.9 : 1

Table 3: **Computational time comparisons between EvoDOCK and RosettaDOCK.** Column 2 shows the total amount of hours required to create 10,000 docking predictions for the complexes in Column 1 using the RosettaDOCK algorithm. Column 3 and 5 show the corresponding time taken by EvoDOCK to create the same number of predictions using the different DE parameters, while Column 4 and 6 show the ratio between the computational time of Rosetta and EvoDOCK.

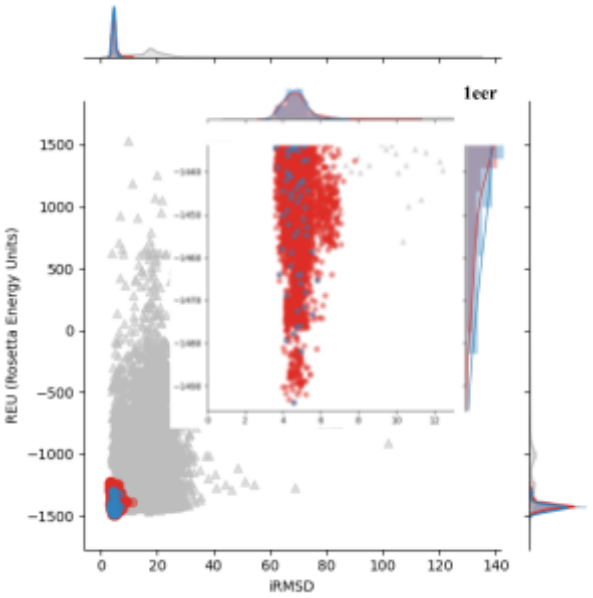

Figure 10. **Results of the 1eer complex using a modified version of EvoDOCK.** The plot shows the distribution of 10,000 predicted docking models of the 1eer complex created by RosettaDOCK (shown as grey triangles) and a modified version of EvoDOCK (shown as red dots) that reduces the frequency of swapping and scoring new backbones. The iRMSD in Å (x-axis) of each model plotted against their score in Rosetta energy units (REU, y-axis) and a zoomed-in version of the plot is placed in the upper left corner. The parameters used for the DE component of EvoDOCK during the benchmark were F = 0.3 and CR = 0.9.

3.5 RMSD difference between ensemble and native backbones

It is also important to take the backbone ensemble used during docking into consideration when evaluating the benchmark results and docking performance of EvoDOCK. Figure 11 shows that the difference in RMSD between the conformers of the larger molecule (usually but not always the receptor) and their native bound-state counterparts was generally quite large. In some cases this difference was very large, such as the receptor protein in the 1f6m complex, showing a lowest RMSD of over 6.5 Å. However, the smaller, often ligand complexes displayed a much smaller minimum RMSD closer to 1 Å. The range between the largest and smallest RMSD for the conformers was mostly between 0.5 and 1 Å, with the exception of the 1f6m and 2ido complexes displaying greater ranges.

The distributions shown in Figure 12 and 13 display a lack of correlation between the RMSD and energy of the conformers, meaning that the lowest energy backbones aren't usually the ones most similar to the native backbone. The figures also showed that the relax protocol consistently produced the lowest energy conformers, with the backrub protocol often giving much higher scores; in the case of the ligand of the 1eer complex, more than 600 REU higher. These findings are reinforced by EvoDOCK log files showing which of the backbones were kept during the conformer library swapping in EvoDOCK. The logs showed that the algorithm settled for backbone conformers created by the relax method after just a few generations. It is also worth noting that the early selected conformers were often kept throughout the rest of the generations without being successfully swapped with new ones.

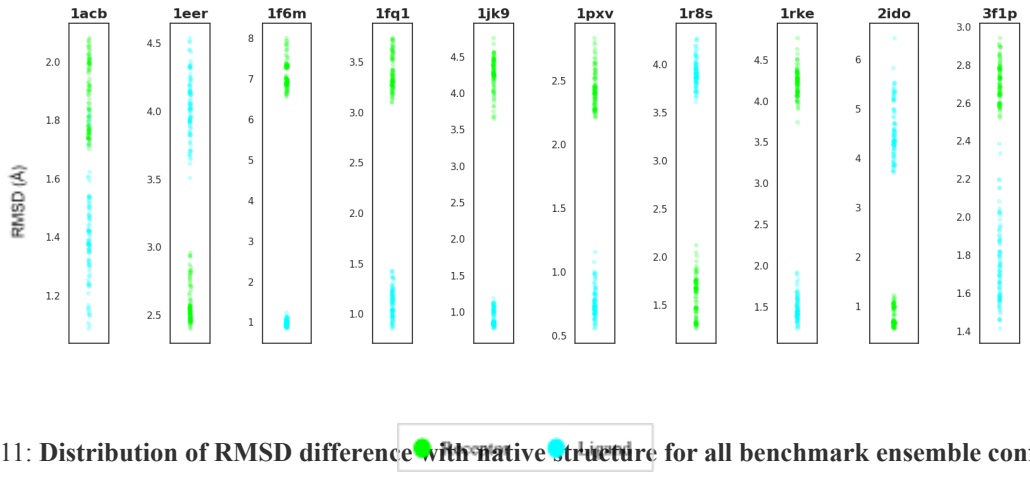

Figure 11: **Distribution of RMSD difference with native structure for all benchmark ensemble conformers.** The ensemble conformers are distributed along the y-axis according to their RMSD in Å when aligned with their native (bound state) counterpart. The receptor conformers are colored green and the ligand conformers cyan.

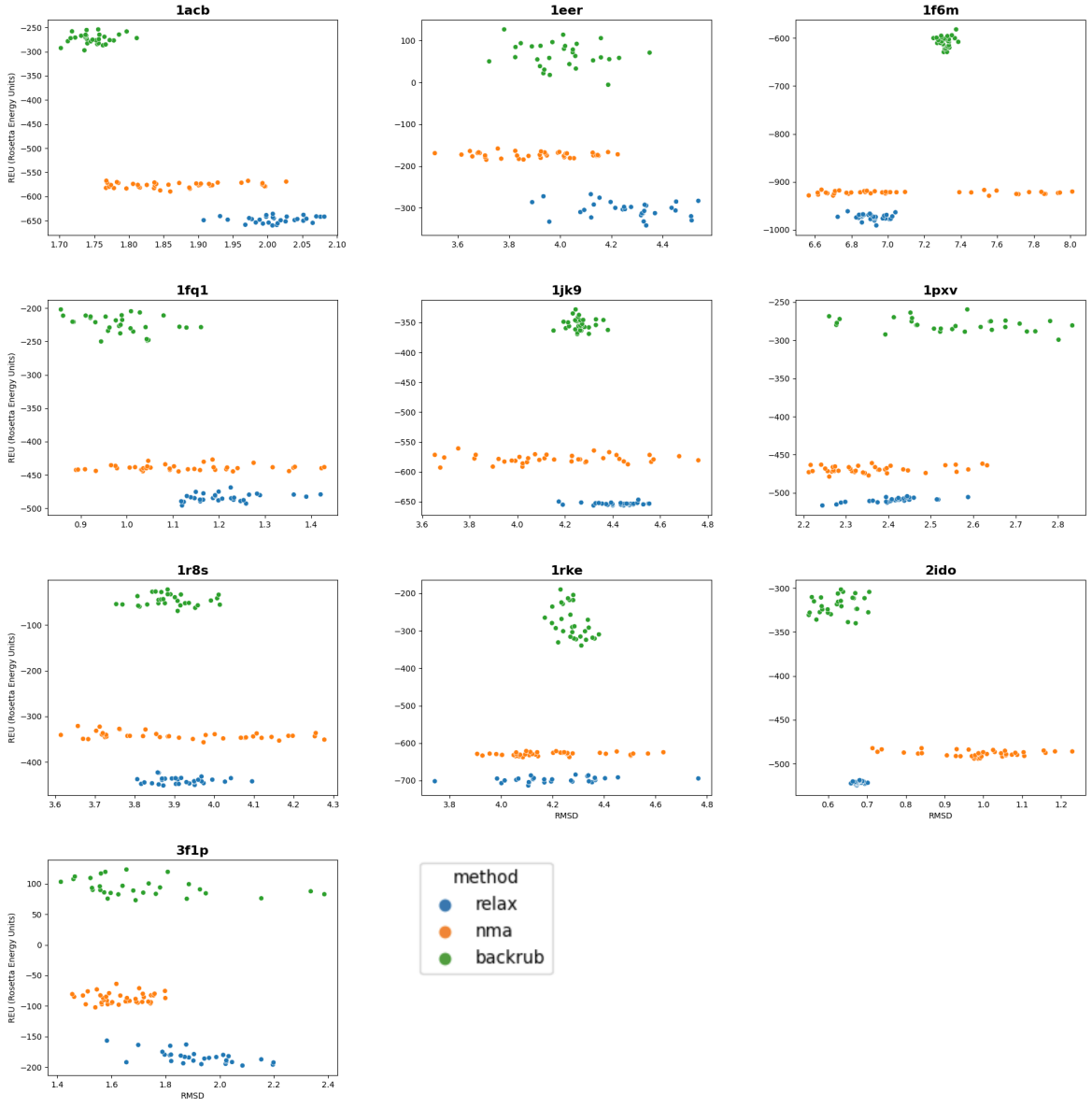

Figure 12: **RMSD and energy distribution of receptor conformers generated using the different methods.** For each of the receptor conformers in the backbone ensembles, the RMSD difference (in Å) from the native receptor (x-axis) is plotted against the conformers energy score (y-axis). The dots are colored based on the backbone generating method used: blue for relax protocol, orange for NMA and green for backrub protocol.

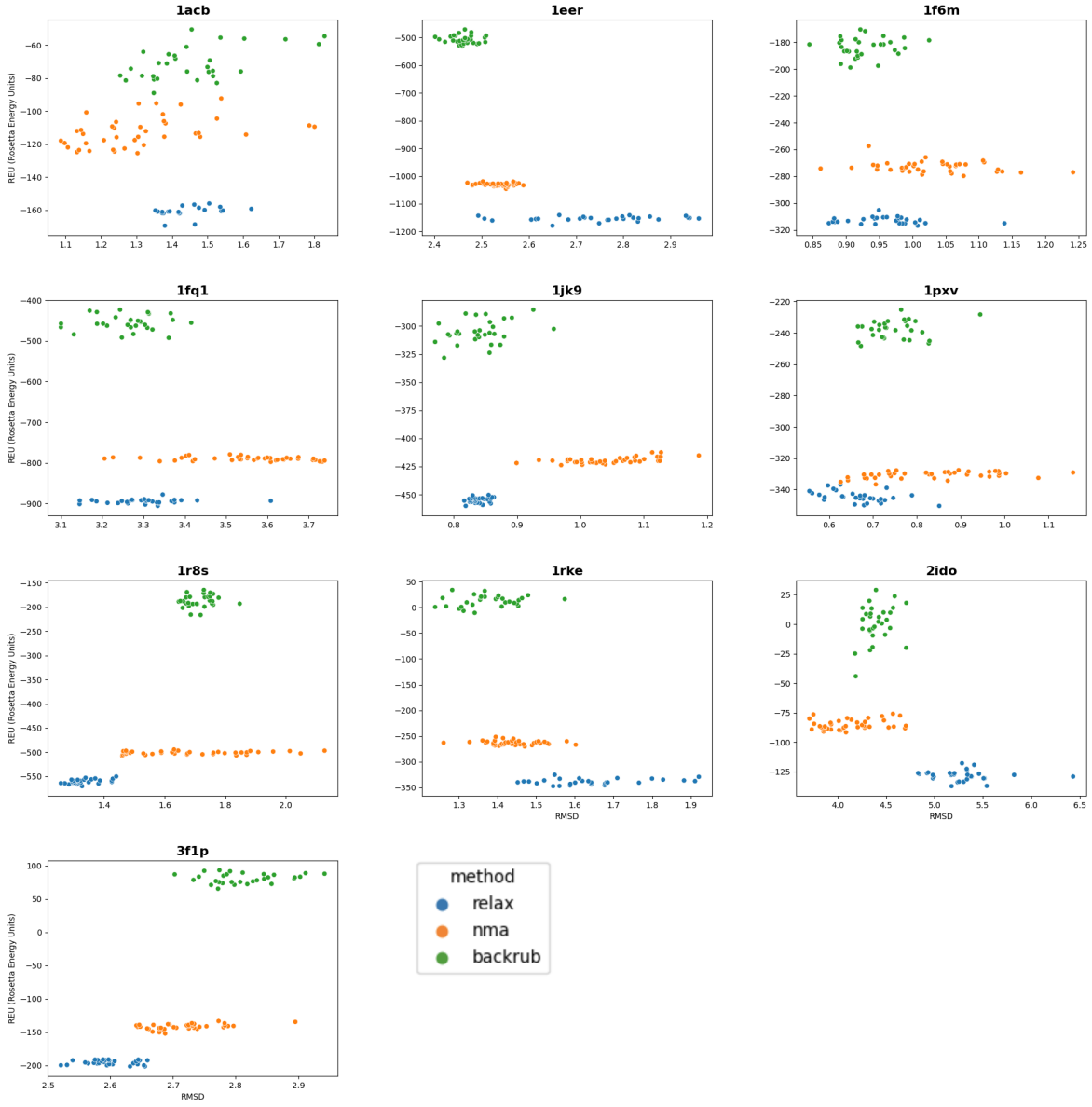

Figure 13: **RMSD and energy distribution of ligand conformers generated using the different methods.** For each of the ligand conformers in the backbone ensembles, the RMSD difference from the native ligand (x-axis) is plotted against the conformers energy score (y-axis). The dots are colored based on the backbone generating method used: blue for relax protocol, orange for NMA and green for backrub protocol.

4 Discussion

The benchmark experiment proved to be successful in comparing the quality of docking predictions made by the memetic algorithm EvoDOCK against the standard Monte-Carlo based RosettaDOCK algorithm. The result of these comparisons showed clear improvements in terms of prediction quality and precision when using EvoDOCK for most of the benchmark complexes, as shown by general improvements in DockQ scores, the lower energies and smaller iRMSD values. The plots confirm these results and additionally show EvoDOCK as having a better ability to identify and exploit energy minima as well as produce more funnel-like shapes for some of the complexes. The performances of the two sets of parameters used for the population-based differential evolution component of the EvoDOCK algorithm produced very similar results in terms of DockQ score, indicating a great robustness of the algorithm. The exception to this trend is a benchmark complex that scored much better when using the exploratory parameters. The exploratory approach did however appear to have a general advantage of not getting stuck in local energy minima as frequently due to the larger perturbations to its position and orientation brought about by its higher mutation rate. However, merely testing two sets of crossover probabilities and mutation rates is not enough to determine the optimal balance between exploration and exploitation. There is a possibility that the exploratory parameters, which were determined and tested for the original rigid-body experiments with EvoDOCK (Varela et al., 2022) does not translate as well to flexible docking, giving cause for testing a wider range of parameters. Another possibly advantageous change worth further investigation would be to change the evolutionary search strategy of mutating using the best scoring solution to only use random individuals instead. This would likely detract from the algorithm's ability to exploit around the area of a promising solution, but potentially carry with it the benefit of being able to get out of local energy minima to find ones with even lower energies.

In addition to precision, efficiency is another non-negligible factor in protein-protein docking; a docking algorithm is simply not feasible if it is too computationally expensive. While the EvoDOCK algorithm often takes two to three times the computational time for generating the same number of predictions as RosettaDOCK does, they are still in the same order of magnitude and both computationally viable, although expensive. While evolutionary algorithms are known for efficiently exploring search spaces, the sheer size of a global full-atom docking search space with both side-chain and backbone flexibility is a likely

contributor to the high computational cost. An attempted optimisation of the flexible backbone component involving a reduced frequency of backbone sampling and scoring by 70% managed to reduce the computational time by 20% without impacting the prediction quality too much for the tested complex. This optimisation shows some potential and could be developed further to perhaps provide a more dynamic approach to backbone conformer swapping like in RosettaDOCK's Adaptive Conformer Selection. Furthermore, it was observed that newly sampled conformers are only kept in the early generations and then very rarely change throughout the preceding generations. A likely explanation for this is that the candidate solutions of the later generations have honed in on low energy docking modes that different backbone conformers would have too poor energy interactions to be viable for. This renders the majority of the backbone swapping and scoring redundant and easily removable for a reduction in computational cost without negatively impacting quality. It is also very likely that the 10,000 predictions made by EvoDOCK are a bit excessive and that a fewer number of trajectories would be sufficient for generating good docking predictions. This is backed up by the high concentration of predictions made by many trajectories with good energy and iRMSD values. Reducing the number of trajectories could therefore provide very large improvements in computational time without much impact on the results.

There are a few observations to be made from the evaluation of the backbone conformers used during the docking experiment. Firstly, the methods for generating the ensembles rarely managed to produce structures with RMSDs that were that much closer to the native structure, instead only performing more small-scale conformational changes. Despite this, the 1jk9 complex which had a moderate RMSD between the conformers and native managed to find a binding mode with a low iRMSD, likely due to the interfacing parts of the conformer proteins being structurally similar to the native ones. Backbone conformers with a closer resemblance to the native proteins would most likely raise the prediction quality, but lower RMSDs does not necessarily correlate with lower energies, as evident by the results. Instead, the only strong correlation to be found is the one between energy and method of generation, with the relax protocol consistently creating the lowest energy conformers and the backrub protocol creating the highest energy ones. The energy gaps between these methods is often large, bringing about the unfortunate effect of the EvoDOCK algorithm selecting and keeping the lowest energy conformers (the ones generated by the relax protocol) after just a few generations. This effectively means discarding 70% of each ensemble and backbone swap, including some backbones with more native-like structure and better energy interactions, just

because their free energy made them unfavourable. A possible solution to allow for the use of a larger percentage of the ensemble conformers would be to normalise their energies by introducing artificial energy reductions on the higher energy conformers. This would in turn lead to a better diversity of viable conformers and likely benefit the finding of new docking modes, but could give misleading results since some of the conformations used would be too energetically unviable to be formed during real-world complex formation. The energy differences between the relax and NMA methods were often not very large, so such an approach would likely work better for them than for the higher energy backrub conformers. A simpler alternative solution could involve just using one method of generation to avoid wasting so many of the backbone swaps and save computational time. However, such a solution would not add much in terms of structural diversity and consequently not help increase the quality of the docking predictions.

5 Methods

5.1 Preparation of benchmark PDB structure files

A small pipeline for testing all the steps of the experiment in sequence was constructed in the *mock_pipeline.py* script. The script was a useful tool for finding bugs and crashes, as well as for testing changes to see if they resolve crashes.

Another pipeline script, *chain_break_pipe.py,* was written for the sake of resolving the crashes occurring during ensemble generation, prepackaging and docking. It performed the following initial operations on the initial PDB files of each protein complex (unbound/bound version of receptor/ligand):

- Removal of molecule ligands (such as FAD).
- Renaming of receptor chains to “A” and ligand chains to “B”.
- Running the Rosetta Backrub Protocol using "-backrub:ntrials 0".

The pipeline script then executes the following steps to resolve the problem of different sequence lengths of the bound and unbound versions of the PDB structure files:

- Generation of bound and unbound FASTA sequences for each of the docking partners using the *pdb2fasta.py* script.

- Pairwise alignment of these sequences using *Bio.pairwise2* to identify missing residues.
- Removal of these unmatching residues.
- Renumbering of the residues.

The bound and unbound complexes were then created from the docking partner files using the *pdb_merger.py* script.

5.2 Ensemble generation

A *command_list.txt* file was created, containing the commands required for generating the ensembles for all of the protein complexes. For each of the receptor and ligand proteins, 30 commands were created: 10 for creating three relax conformers, 10 creating four NMA conformers and 10 creating three backrub conformers. A SLURM script making use of this command list and a python script for managing multiple parallel processes was created and submitted to a cluster provided by the Swedish National Infrastructure for Computing (SNIC) at HPC2N.

The following three command lines were used for ensemble generation:

```
relax.static.linuxgccrelease
     -in:file:s <Unbound receptor or ligand PDB>
     -nstruct 3
     -relax:thorough

rosetta_scripts.static.linuxgccrelease
     -in:file:s <Unbound receptor or ligand PDB>
     -nstruct 4
     -parser:protocol nma.xml

backrub.static.linuxgccrelease
     -in:file:s <Unbound receptor or ligand PDB>
     -nstruct 3
     -backrub:ntrials 20000 -backrub:mc_kt 0.6
```

The nma.xml file used by the normal mode analysis method contained the following text:

```
<ROSETTASCRIPTS>
 <SCOREFXNS>
 <ScoreFunction name="bn15_cart" weights="beta_nov15_cart" />
 </SCOREFXNS>
 <RESIDUE_SELECTORS>
 </RESIDUE_SELECTORS>
 <TASKOPERATIONS>
 </TASKOPERATIONS>
```

```
 <FILTERS>
 </FILTERS>
 <MOVERS>
 <NormalModeRelax name="nma" cartesian="true" centroid="false"
scorefxn="bn15_cart" nmodes="5" mix_modes="true" pertscale="1.0"
randomselect="false" relaxmode="relax" nsample="20"
cartesian_minimize="false" />
 </MOVERS>
 <APPLY_TO_POSE>
 </APPLY_TO_POSE>
 <PROTOCOLS>
 <Add mover="nma" />
 </PROTOCOLS>
 <OUTPUT scorefxn="bn15_cart" />
</ROSETTASCRIPTS>
```

5.2.1 Prepacking

The *prepack_pipleline.py* pipeline script was created for the sake of preparing and packing the side chains on all of the backbones used during docking. It also created text files listing the paths to the pdb files in the ensembles. An additional script provided by EvoDOCK, *prepackaging.py*, was used to prepack the bound state complexes prior to docking.

The following command line was used for prepackaging of the backbone ensembles and unbound complexes:

```
docking_prepack_protocol.linuxgccrelease
      -in:file:s <Unbound complex>
      -nstruct 1
      -ensemble1 <Receptor conformer list>
      -ensemble2 <Ligand conformer list>
      -partners A_B
      -detect_disulf true
      -rebuild_disulf true
      -ex1
      -ex2aro
```

5.3 EvoDOCK docking protocol

Batch jobs executing parallel instances of EvoDOCK using the evodock.py script were created for each of the 10 complexes and submitted to the cluster provided by LUNARC, the center for scientific and technical computing at Lund University. The configuration files used for running EvoDOCK had the following structure, with a more detailed set of instructions on https://github.com/Andre-lab/evodock:

```
[Docking]
type=Unbound
bb_strategy=popul_library
```

```
[inputs]
pose_input=<Prepacked unbound complex>
native_input=<Prepacked bound complex>

path_ligands=<Paths to the ligand conformers>
path_receptors=<Paths to the receptor conformers>

[outputs]
output_file=<Output log file>

[DE]
scheme=BEST
popsize=100
mutate=<Mutation rate (F)>
recombination=<Crossover probability (CR)>
maxiter=100
local_search=mcm_rosetta
```

The mutation rate was set to 0.3 for the first round of EvoDock and 0.9 for the second, while the crossover probability was set to 0.9 for the first and 0.3 for the second.

5.3.1 Reducing the sampling of new backbone conformers

The following condition was added before the sampling and scoring of new backbone conformers during the popul_library backbone strategy in line 112 of the local_search_strategy.py script:

```
if random.uniform(0, 1) < 0.3:
```

5.4 RosettaDOCK docking protocol

Batch jobs executing the following command line on 25 separate cores were written and submitted to the SNIC cluster for each the 10 complexes:

```
docking_protocol.static.linuxgccrelease

      -database <database>
      -out:level 500
      -in:file:s <Prepacked unbound complex>
      -in:file:native <Bound complex>
      -nstruct 400
      -ensemble1 <Receptor conformer list>
      -ensemble2 <Ligand conformer list>
      -partners A_B
      -dock_pert 3 8
      -spin
      -detect_disulf true
      -rebuild_disulf true
      -ex1
      -ex2aro
```

Bibliography

Alberts, B., & Miake-Lye, R. (1992). Unscrambling the puzzle of biological machines: the importance of the details. Cell, 68(3), 415-420. https://doi.org/10.1016/0092-8674(92)90179-G

Basu, S., & Wallner, B. (2016). DockQ: a quality measure for protein-protein docking models. PloS one, 11(8), e0161879. https://doi.org/10.1371/journal.pone.0161879

Chaudhury, S., & Gray, J. J. (2008). Conformer selection and induced fit in flexible backbone protein–protein docking using computational and NMR ensembles. Journal of molecular biology, 381(4), 1068-1087. https://doi.org/10.1016/j.jmb.2008.05.042

Chaudhury, S., Berrondo, M., Weitzner, B. D., Muthu, P., Bergman, H., & Gray, J. J. (2011). Benchmarking and analysis of protein docking performance in Rosetta v3. 2. PloS one, 6(8), e22477. https://doi.org/10.1371/journal.pone.0022477

Coello Coello C.A. (2005). An Introduction to Evolutionary Algorithms and Their Applications. In: Ramos F.F., Larios Rosillo V., Unger H. (eds) Advanced Distributed Systems. ISSADS 2005. Lecture Notes in Computer Science, vol 3563. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11533962_39

I.A. Vakser, O.G. Matar, C.F. Lam. (1999). A systematic study of low-resolution recognition in protein–protein complexes. Proc. Natl Acad. Sci. USA, 96 pp. 8477-8482 https://doi.org/10.1073/pnas.96.15.8477

Ideker, T., & Sharan, R. (2008). Protein networks in disease. Genome research, 18(4), 644-652. https://doi.org/10.1101/gr.071852.107

Jeffrey J. Gray, Stewart Moughon, Chu Wang, Ora Schueler-Furman, Brian Kuhlman, Carol A. Rohl, David Baker. (2003), Protein–Protein Docking with Simultaneous Optimization of Rigid-body Displacement and Side-chain Conformations, Journal of Molecular Biology 331(1), 281-299. https://doi.org/10.1016/S0022-2836(03)00670-3

Kastritis, P. L., & Bonvin, A. M. (2010). Are scoring functions in protein− protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. Journal of proteome research, 9(5), 2216-2225. https://doi.org/10.1021/pr9009854

Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., & Vakser, I. A. (1992). Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. Proceedings of the National Academy of Sciences of the United States of America, 89(6), 2195–2199. https://doi.org/10.1073/pnas.89.6.2195

Krivov, G. G., Shapovalov, M. V., & Dunbrack Jr, R. L. (2009). Improved prediction of protein side‐chain conformations with SCWRL4. Proteins: Structure, Function, and Bioinformatics, 77(4), 778-795. https://doi.org/10.1002/prot.22488

Lensink, M. F., & Wodak, S. J. (2013). Docking, scoring, and affinity prediction in CAPRI. Proteins: Structure, Function, and Bioinformatics, 81(12), 2082-2095. https://doi.org/10.1002/prot.24428

Lensink, M. F., Nadzirin, N., Velankar, S., & Wodak, S. J. (2020). Modeling protein‐protein, protein‐peptide, and protein‐oligosaccharide complexes: CAPRI 7th edition. Proteins: Structure, Function, and Bioinformatics, 88(8), 916-938. https://doi.org/10.1002/prot.25870

Lensink, M. F., Velankar, S., & Wodak, S. J. (2017). Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition. Proteins: Structure, Function, and Bioinformatics, 85(3), 359-377. https://doi.org/10.1002/prot.25215

Li, X., Moal, I.H. and Bates, P.A. (2010). Detection and refinement of encounter complexes for protein–protein docking: Taking account of macromolecular crowding. Proteins, 78: 3189-3196. https://doi.org/10.1002/prot.22770

Marze, N. A., Jeliazkov, J. R., Roy Burman, S. S., Boyken, S. E., DiMaio, F., & Gray, J. J. (2017). Modeling oblong proteins and water‐mediated interfaces with RosettaDock in CAPRI rounds 28–35. Proteins: Structure, Function, and Bioinformatics, 85(3), 479-486. https://doi.org/10.1002/prot.25168

Marze, N. A., Roy Burman, S. S., Sheffler, W., & Gray, J. J. (2018). Efficient flexible backbone protein-protein docking for challenging targets. Bioinformatics (Oxford, England), 34(20), 3461–3469. https://doi.org/10.1093/bioinformatics/bty355

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. (1953). 'Equation of State Calculations by Fast Computing Machines', Chem. Phys. 21, 1087-1092. https://doi.org/10.1063/1.1699114

Roel-Touris, J., Don, C. G., V. Honorato, R., Rodrigues, J. P. G. L. M., & Bonvin, A. M. J. J.. (2019). Less Is More: Coarse-Grained Integrative Modeling of Large Biomolecular Assemblies with HADDOCK. Journal of Chemical Theory and Computation, 15(11), 6358–6367. https://doi.org/10.1021/acs.jctc.9b00310

Storn, R., Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359. https://doi.org/10.1023/A:1008202821328

Sunny, S., & Jayaraj, P. B.. (2022). Protein–Protein Docking: Past, Present, and Future. The Protein Journal, 41(1), 1–26. https://doi.org/10.1007/s10930-021-10031-8

Vakser I. A. (2014). Protein-protein docking: from interaction to interactome. Biophysical journal, 107(8), 1785–1793. https://doi.org/10.1016/j.bpj.2014.08.033

Varela, D., & André, I. (2021). A memetic algorithm enables global all-atom protein-protein docking with sidechain flexibility. bioRxiv. https://doi.org/10.1101/2021.04.12.437963

Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., ... & Weng, Z. (2015). Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. Journal of molecular biology, 427(19), 3031-3041. https://doi.org/10.1016/j.jmb.2015.07.016

Wang, C., Schueler-Furman, O., & Baker, D. (2005). Improved side-chain modeling for protein-protein docking. Protein science : a publication of the Protein Society, 14(5), 1328–1339. https://doi.org/10.1110/ps.041222905

Zemla, A., Venclovas, Č., Moult, J., & Fidelis, K. (1999). Processing and analysis of CASP3 protein structure predictions. Proteins: Structure, Function, and Bioinformatics, 37(S3), 22-29. https://doi.org/10.1002/(SICI)1097-0134(1999)37:3+<22::AID-PROT5>3.0.CO;2-W