MeetDockOne: team 1 scoring method for protein-protein docking.

François Gravey, Guillaume Delevoye, Paula Milán Rodríguez, Ilyes Abdelhamid, Maxime Borry

1 - Introduction

Protein-protein docking aims to predict the structure of a protein-protein complex from its unbound components. However, it remains a tricky challenge in structural bioinformatics. An important step of this process is the ranking of docked poses using a scoring function, for which many methods have been developed.

Student in Bioinformatics MSc. from three different universities (Paris 6, Paris 7 and Paris 11) aimed to develop a protein docking program by integrating both theoretical and applied knowledge in order to answer this modern biology problem. To do this, the protein docking workflow is broken up into two steps: sampling and scoring.

As a scoring team, the hypothesis of our work is that the integration of multiple metrics summarizing different informations regarding the docking (amino acid contact propensities, shape complementarity, electrostatic energy and Van der Waals interactions) will allow us to identify good poses that are relevant in terms of biology. In this perspective, the use of an unbound docking benchmark with the implementation of concepts derived from docking are particularly well adapted to this integration, in order to propose a program capable of correctly scoring different protein-protein complexes.

2 - Material and Methods

2.1 Generating and evaluating the docking decoys:

To generate docking poses in order to later score them, a Fibonacci sphere approach was employed (developed by the team 6). The Fibonacci sphere approach allows to evenly spread N points on a sphere around its center.

This technique starts with two PDB files, one for the receptor, and one for the ligand. For each molecule, its center of mass is computed, and taken as the reference point for the molecule. The center of mass of the receptor is taken as the center of the Fibonacci sphere, and the center of mass of the ligand is positioned on the N points of the Fibonacci sphere giving N_l ligand positions (Fig. 1). Each N_l is rotated along its 3 axis x, y, z. The distance between the receptor (R) and the ligand (L) is chosen so that the closest atom between R and

L overlap by a distance d. For this analysis, we chose N_l=20, 3 rotations per axis, and d = -4 Angstroms (Å), giving a total of *N_l X 3^3 poses.*

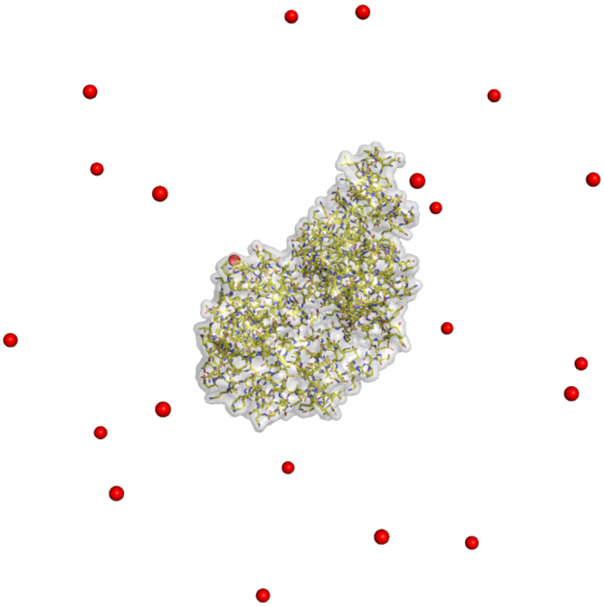


Figure 1: Fibonacci sphere around receptor of 4H03 PDB. The receptor is shown in yellow “stick” representation, while the N_l starting positions are shown as red points.

For each complex, we computed different scores.

We used both MSMS program (Sanner MF et al., 1996) and NACCESS (S. Hubbard and J. Thornton., 1993) to identify the receptor surface residues. MSMS implements a residue depth calculation whereas the accessible surface area (ASA) is computed with NACCESS using a probe radius of 1.4 Å.

We considered the surface residues defined according to the degree of exposure to the solvent; the surface was identified by the residues whose relative ASA is at least 25% of the total residue surface. As for MSMS, a residue depth cutoff was set to 4 Å in order to identify surface residues. For the rest of the analysis, only the ASA values of residues at the predicted docking interface were used.

2.2 Interface prediction:

If a ligand’s residue and a receptor’s residue were both exposed at the surface and the distance between them was lower than 8.6 Å, we considered that they interact. The coordinates taken as reference were the ones of the beta carbon with one exception: if the residue was a glycine we took the coordinates of the alpha carbon. The set of interacting residues was considered as the interface of the complex.

After having identified this interface, we implemented the following metrics described bellow: statistic potential, shape complementarity, electrostatic energy and Van der Waals forces.

2.3 Statistic potential:

The interface propensity of a contact between amino acids i and j is a measure of how often the contact between i and j occured at known protein interfaces compared to the expected frequency. This final score for each complex was calculated by means of a matrix that described the pair potentials fitting the characteristics of i and j. We have used the score matrix that stems from the Glaser method (Pons *et al*., 2011).

2.4 Shape Complementarity:

The shape complementarity scoring follows the approach described by Chen and Weng, 2002. Briefly, we start by discretizing separately each member of the complex on a grid of a 2 Å mesh size. Each point of the grid was attributed a score of either:

- 0 if outside of a molecule
- 1 if at the surface of a molecule
- $-9i$ if inside a molecule

Afterwards, we multiplied the receptor grid, with the ligand grid, element by element, for each point of the grid. A point with surface-surface contact equals to 1, a point with surface-outside contact equals to 0, a point with surface-inside contact equals to -9i, and a point with inside-inside contact equals to $-81i$.

We summed over all the points of the grid, and took the real part to obtain the shape complementarity score. A very negative scores translates in a lot of steric clashes, a very positive score translates in a lot of surface contacts between receptor and ligand, while a score around 0 translates by the lack of contact between the receptor and the ligand.

2.5 Electrostatic energy:

The electrostatic interactions were computed for residues (TYR, HIS, CYS, ASP, GLN, LYS, ARG) in the predicted interface with Equation 1.

$$V elec = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{qi\, qj}{4\, \pi\, \varepsilon 0\, rij}$$

Equation 1: Electrostatic interactions

With i and j bein atoms involved in the interaction, ε_0 being the vacuum permittivity, and q_i q_j being the charge of i and j, at pH 7.

2.5 Lennard-Jones/VdW Potential:

Lennard-Jones potential was computed for the interface's residues with Equation 2:

$$V_{ij} = 4\varepsilon \left[\left(\frac{\theta}{r_{ij}} \right)^{12} - \left(\frac{\theta}{r_{ij}} \right)^{6} \right]$$

Equation 2: Lennard-Jones Potential, Clementi et al, 1999

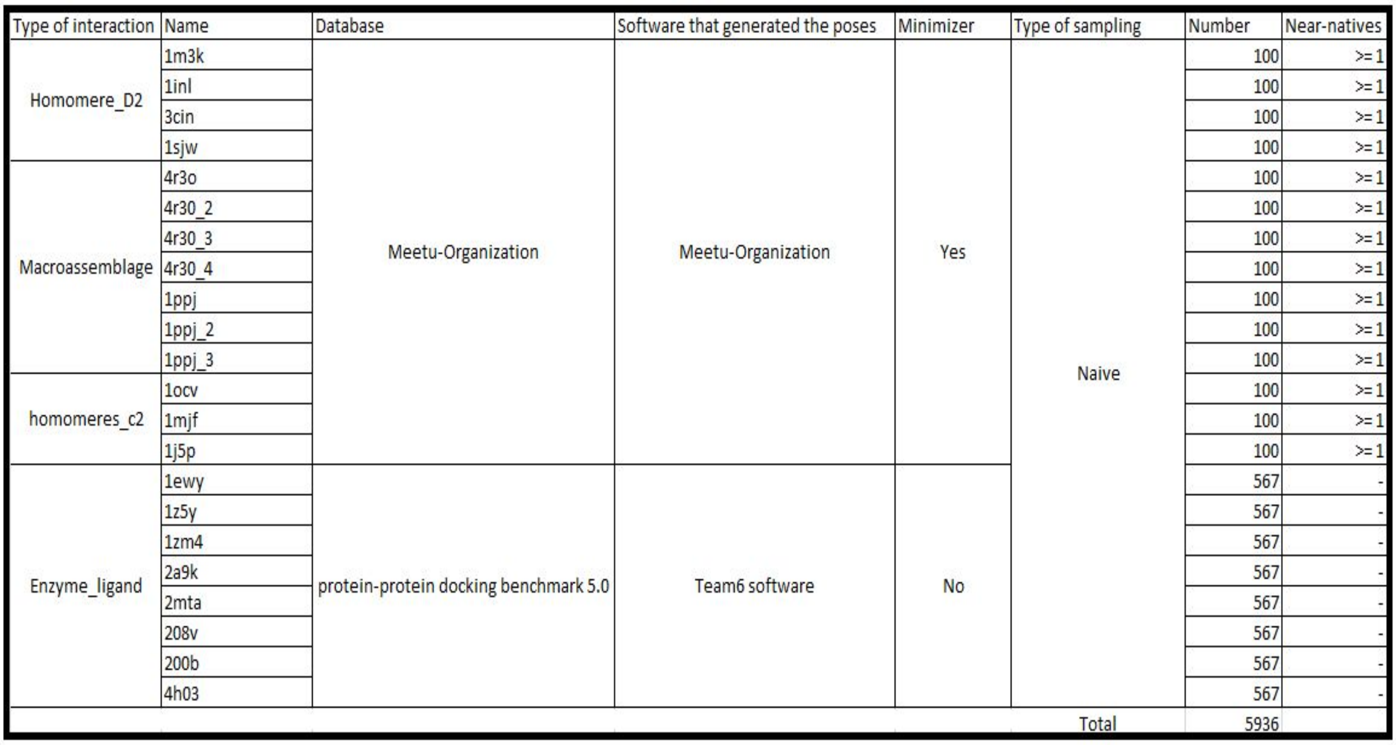
Where r was the distance between the residues i (receptor) and j (ligand). The default value was 10 for Ɛ and 3.9 for Ɵ. Van der Waals forces were considered in the second term r^{-6}, which describes the attraction at long ranges.

2. 6 Machine learning development:

2.6.1 Data selection:

We decided to use a machine learning algorithm with scikit-learn (Sklearn) (Pedregosa et al., 2012) on our previously described scores to classify complexes into 4 categories: 'Excellent', 'Good', 'Bad', 'Poor'. Indeed we wanted to build a generalist model for us and the other teams we tried to include diverse type of protein complexes.

First of all, we recovered sampling data made available on GitHub by the meet-u teaching team. Then we added other complexes from protein-protein benchmark v5.0 (Vreven et al, 2015) which were sampled using the sampling software of Team 6. Because we chose to dock enzyme-substrate interaction, we focused on enzyme-substrate data. Altogether, we analysed 5936 decoys of 17 different native complexes. The composition of our learning dataset is summarized below in table I.



Type of interaction	Name	Database	Software that generated the poses	Minimizer	Type of sampling	Number	Near-natives
Homomere_D2	1m3k	Meetu-Organization	Meetu-Organization	Yes	Naive	100	>=1
	1inl					100	>=1
	3cin					100	>=1
	1sjw					100	>=1
Macroassemblage	4r3o					100	>=1
	4r30_2					100	>=1
	4r30_3					100	>=1
	4r30_4					100	>=1
	1ppj					100	>=1
	1ppj_2					100	>=1
	1ppj_3					100	>=1
homomeres_c2	1ocv					100	>=1
	1mjf					100	>=1
	1j5p					100	>=1
Enzyme_ligand	1ewy	protein-protein docking benchmark 5.0	Team6 software	No		567	-
	1z5y					567	-
	1zm4					567	-
	2a9k					567	-
	2mta					567	-
	208v					567	-
	200b					567	-
	4h03					567	-
					Total	5936	

Table I: Summary of the learning dataset. Near-natives structures have been found in every sample, but the criteria differs between samples.

2.6.2 Data learning preparation:

Thanks to the script '*data_handler.py*', all the selected pdb were automatically evaluated through all our scoring functions and the Tmscore.f software (Zhang 2004, Protein).

The learning data were composed of four features: the contact propensity ('statpot'), the shape complementarity (shape), the electrostatic energy (elector) and the Van der Waals interactions (VdW). All the features were compared to our label which was the calculated Tm score (after maximum of alignment). All the data was organized into one file in .csv format.

2.6.3 Data preparation and normalization:

The script used to developed the machine learning algorithm was called *'analyse_donnes_sans_sampling.py'*. First of all, the data was put in a Pandas dataframe object in order to analyze it, and check for missing values and to perform normalization.

A Sklearn pipeline was developed in order to prepare the data for training the machine learning. First of all, the presence of missing values was tracked using the Imputer of Sklearn with the median strategy. In this approach, any missing data was replaced by the median value of the column. Then, all the data was normalized using the sklearn algorithm StandardScaler.

We compared three different machine learning algorithms. They were, linear regression, decision tree regressor, and random forest regressor.

In order to determine the best model, we performed for all the algorithms a cross validation based on the mean squared error (mse) comparing the predictions to the labels with a 'K fold' cross validation (n = 10). Then, when our best model was found, we employed the GridSearchCV algorithm in order to determine the best hyper-parameters values. We evaluated three different hyper-parameters: the number of trees (n=10, 50 or 100), the number of features (n=1, 2, 3, or 4), and the Bootstrap (True or False).

We added interpretation of the predict TM score using this scale: TM score < 0.4 'Poor', 0.4 < TM score < 0.6 'Passable', 0.6 < TM score < 0.8 'Good', TM score > 0.8 'Excellent'.

Finally, we used the Python Library Pickle in order to use our machine learning algorithm through our scoring function.

We released for our users a machine learning algorithm trained with all the data previously described. Nevertheless, as our biology question was to analyze the enzyme-substrate interactions, we used a specific algorithm which contained only the 4536 enzyme-substrate poses.

2.7 Application of our program to an enzyme-substrate docking:

In order to illustrate how our program works, we decided to present the results obtained with the analyze of the 2PCC which is the pdb of the interaction between the cytochrome C and and the cytochrome C peroxidase.

We realised 567 differents poses using the sampling script proposed by the team 6 and sorted all the complexes using MeetDockOne. For the purpose of evaluating the performance of our predicted Tm score, we also calculated separately the 'real' Tm score with the Tmscore.f software.

All the complexes were sorted according to the 'real' Tm score to check for MeetDockOne ability to predict near-native complexes.

3 - Results

3.1 Machine learning algorithm:

In order to find the best algorithm, we performed a cross validation process based on the mse. Regarding to our data, the random forest regressor appeared to be the best model to use. Indeed, this model had the lowest mse and the lowest standard deviation. Based on ten tests, the mean mse was 0,064 with a standard deviation at 0,0156. In the same time, the performances of the two other models were poorest, 0,089/0,011 and 0,081/0,23 for both linear regression and the decision tree regressor respectively.

We next, tried to tune our random forest model using the GridSearchCV function and looked for the best value for three hyper-parameters. The best random forest regressor model appeared with these following parameters: bootstrap=True, criterion='mse', max_depth=None, max_features=3, max_leaf_nodes=None, min_impurity_decrease=0.0, min_samples_split = 2, min_weight_fraction_leaf = 0, n_estimators=100, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False.

We extracted the weight of each feature in our model. We used the model.feature_importance_ to do so. Our features, ‘electro’, ‘shape’, ‘statpot’ and ‘vdw’ harbored an importance evaluated at 0,09, 0,57, 0,23 and 0,11 respectively.

3.2 2PCC results:

First of all, we were be able to obtain a score and a classification for all the different complexes generated by the sampling algorithm.

3.2.1 Tm score distribution:

The figure 2 illustrates the distribution of the predicted Tm score in comparison to the ‘real’ Tm scores. The values of predicted Tm score ranged from 0,48 to 0,82 whereas the ‘real’ Tm score harbored values between 0,74 to 0,82. As the consequence the distribution of the predicted Tm score was larger than the ‘real’ Tm score.

The analysis of the distribution of the two scores highlighted two interesting facts.
First of all, some high ‘real’ Tm scores were miss evaluated by our algorithm (red circle on the figure) whereas some high ‘real’ Tm score were better evaluated by our algorithm (green circle on the figure). It appeared like two completely different complexes among the high ‘real’ Tm score. The miss evaluated high ‘real’ Tm score will be discussed during the

discussion part. On the other hand, for average values, our predicted Tm score was quite well associated to the 'real' one.

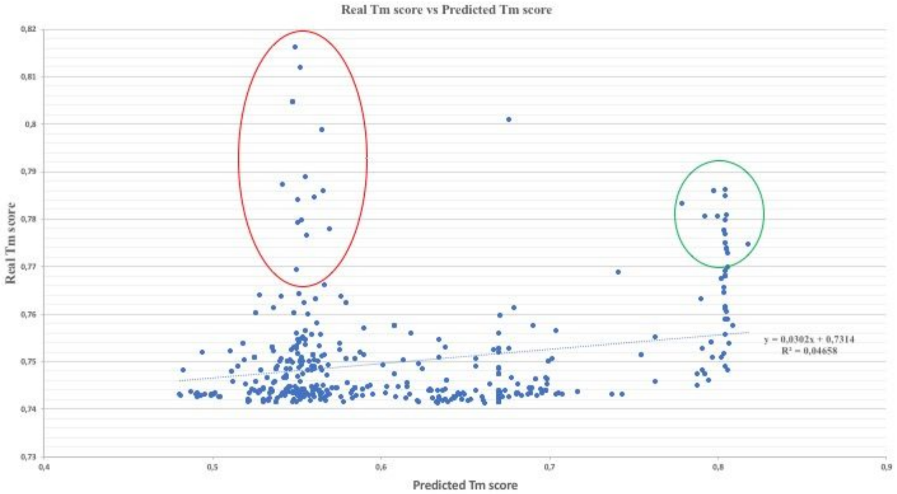


Figure 2: Distribution of the predicted Tm score regarding to the 'real' Tm score.
Red circle highlighted the wrong distribution of the predicted Tm score whereas the green circle illustrated the correct distribution of the predicted Tm score.

3.2.2 Ability to find the correct and the incorrect poses:

For the purpose of this analysis, we sorted all the generated dockings poses according to the 'real' Tm score. We then checked the ability of MeetDockOne to correctly identify the 'best' and the 'worst' poses. Results are summarized in Table II.

Among the top 50 poses, MeetDockOne was able to identify 23 as excellent. Moreover, 38 were classed as 'excellent' poses, including 33 were in the Top 100. We were pleased to see that no pose that belongs to the worse poses was classified as excellent.

Finally, among the 467 worst poses, 272 were classified into the 'Passable' category. Interestingly, no pose was classified in the 'poor' category which is in agreement with the distribution of the 'real' Tm score. This can be explained by the fact that we calculated the Tm score on the whole protein complex, with a fixed receptor. Therefore, because of the receptor contribution to the Tm score, it won't be lower than 0.48 .

Nevertheless, we did not succeed to correctly classify all the best poses: among the top 10 no one was in the 'Excellent' category, one belonged to the 'Good' category and nine belonged to the 'Passable' category. However, the top poses display a quite similar conformation to the native pose (fig 3).

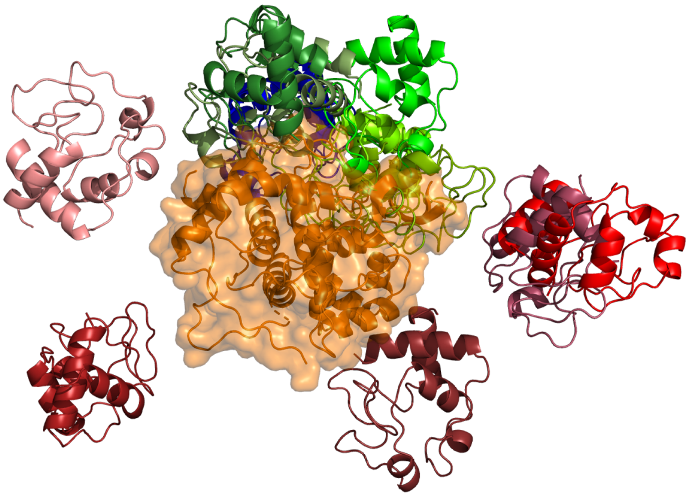


Figure 3: 2PCC native complex (receptor in orange, ligand in blue). Top 5 poses (shades of green), worst 5 poses (shades of red), according to MeetDockOne predicted Tm score

MeetDockOne predictions	Ranking regarding the 'real' Tm score values						
	Top 1 - 10	Top 11 - 50	Top 51 -100	Top 101 - 200	Top 201 - 300	Top 301 - 567	Total
Excellent	0	23	10	4	1	0	38
Good	1	6	14	24	38	128	211
Passable	9	11	26	72	61	139	318
Poor	0	0	0	0	0	0	0
Total	10	40	50	100	100	267	567

Table II: Comparison of the interpretation of MeetDockOne regarding to the ranking made on the 'real' Tm score values

4 - Discussion

We decided not to include the nature of the ppi (Protein Protein interaction) into our machine learning algorithm. We compared the performances of our program with or without the presence of this information. To not include the nature of the ppi costed only 1% accuracy to our predictions.

Therefore we maintained our decision to not use the nature of the ppi in order to make our scoring program easier to use, which would constrain the user to put this information into our algorithm in order to obtain a predicted Tm score.

Another reason is also a limitation in our machine learning process: we included 'only' four types of ppi. Therefore, using any different ppi from our ppi list, will likely produce unpredictable docking results.

These two main reasons led us to not use the ppi into our machine learning algorithm. This is one of the axis of further development: use more data belonging of complexes having different PPI.

In order to understand why we had a 'double' population among the high 'real' Tm score, we look for all the results obtained by these complexes through our scoring function. It appeared that, all the miss classified complexes were attributed a score of 0 for electrostatic and VdW interactions, as well as amino acids interactions propensities. These complexes have a shape complementarity score equal or close to 0. This denotes a lack of interaction between the receptor and the ligand because of too great distance separating them, due to the sampling algorithm used and the absence of minimization.

5 - Conclusions and Perspectives

Although docking methods have been the subject of numerous studies, our own machine learning-related predictive tools of docking had yet to be constructed from ground up. The results of this work show that the different metrics we used and our machine learning method allow for a good degree of consistency in the construction and interpretation of scoring.

The method of pose generation and scoring discussed above, and applied to several sets of data derived from one kind of ppi (enzyme-substrate complex), allowed us to illustrate during this work how to make a relevant use of structural information to assess and predict docking of a particular kind of interaction. However, at this stage it would be desirable to consider improvements in our methodological approach.

The docking data analysis strategy of integrating several metrics to analyze them globally was particularly relevant. Nevertheless, it has a number of pitfalls. Although we obtained good poses close to the native's with our validation set (PDB 2PCC), the sampling of different types of enzyme-substrate complexes remains low.

Another lead to explore lies in the predictive aspect. To date, our program is focused on one type of binding, but it is only a matter of computational time and resources availabilities to build a more extensive machine learning model.

Single docking experiments are useful for exploring the function of the target, and virtual screening, where a large library of compounds are docked and ranked, may be used to identify new inhibitors for drug development. In this context, extensive application of the docking concepts introduced in our program on data from specific biological questions can lead to improved knowledge of molecular mechanisms at work in diseases.

6 - References

- Sanner, MF., Olson, AJ., and Spehner, JC. (1996). Reduced surface: an efficient way to compute molecular surfaces. Biopolymers, 38, 305-320.
- Hubbard, S., and Thornton, J. 1993. NACCESS, Computer Program. Department of Biochemistry Molecular Biology, University College London.
- Pons, C., Glaser, F., and Fernandez-Recio, J. (2011). Prediction of protein-binding areas by small-world residue networks and application to docking. BMC Bioinformatics 12:378.
- Chen, R., and Weng, Z. (2002). Docking unbound proteins using shape complementarity, desolvation, and electrostatics. Proteins: Structure, Function, and Genetics *47*, 281–294.
- Clementi, C., M. Vendruscolo, A. Maritan, y E. Domany.(1999). Folding Lennard-Jones Proteins by a Contact Potential. *Proteins* 37, n.º 4, 544-53.
- Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12.Oct (2011): 2825-2830.
- Vreven, Thom, et al. "Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2." *Journal of molecular biology*427.19 (2015): 3031-3041.
- Zhang, Yang, and Jeffrey Skolnick. "Scoring function for automated assessment of protein structure template quality." *Proteins: Structure, Function, and Bioinformatics* 57.4 (2004): 702-710.
- Schymkowitz, Joost, et al. "The FoldX web server: an online force field." *Nucleic acids research* 33.suppl_2 (2005): W382-W388.