A systematic analysis of scoring functions in rigid-body protein docking: the delicate balance between the predictive rate improvement and the risk of overtraining

Didier Barradas-Bautista[1], Iain H. Moal[1,2] and Juan Fernández-Recio[1]*

[1] Barcelona Supercomputing Center (BSC), Joint BSC-CRG-IRB Research Program in Computational Biology, 08034 Barcelona, Spain

[2] European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK.

Corresponding author:

Juan Fernández-Recio

Barcelona Supercomputing Center, C/ Jordi Girona 29, 08034 Barcelona, Spain

e-mail: juanf@bsc.es

Short title: Scoring functions in rigid-body docking

Keywords: protein-protein interactions, structural modeling, complex structure prediction, set-theoretic measure, cardinality analysis, union, symmetric difference

Abstract

Protein-protein interactions play fundamental roles in biological processes including signaling, metabolism and trafficking. While the structure of a protein complex reveals crucial details about the interaction, it is often difficult to acquire this information experimentally. As the number of interactions discovered increases faster than they can be characterized, protein-protein docking calculations may be able to reduce this disparity by providing models of the interacting proteins. Rigid-body docking is a widely used docking approach, and is often capable of generating a pool of models within which a near-native structure can be found. These models need to be scored in order to select the acceptable ones from the set of poses. Recently, more than 100 scoring functions from the CCharPPI server were evaluated for this task using decoy structures generated with SwarmDock. Here, we extend this analysis to identify the predictive success rates of the scoring functions on decoys from three rigid-body docking programs, ZDOCK, FTDock and SDOCK, allowing us to assess the transferability of the functions. We also apply set-theoretic measure to test whether the scoring functions are capable of identifying near-native poses within different subsets of the benchmark. This information can provide guides for the use of the most efficient scoring function for each docking method, as well as instruct future scoring functions development efforts.

Introduction

Protein-protein interactions are known to play key roles in almost all cellular and biological processes such as signaling, metabolism, and trafficking.[1] Thanks to experimental high-throughput screening techniques, the volume of annotated data on protein-protein interactions has experienced a huge increase in recent years.[2,3] The structural characterization of such protein interactions can provide molecular details on the determinants of their specificity and affinity, as well as on their mechanism of association.[4,5] However, technical difficulties in the determination of the 3D structures of protein complexes is causing the structural coverage of interactomes to increase at a much slower pace.[6] In this context, computational methods such as protein-protein docking, which aims to predict the structure of a protein-protein complex from its monomeric constituents, can be extremely useful to complement current experimental efforts. International efforts like the Critical Assessment of PRedicted Interactions (CAPRI)[7] have boosted the development of novel and more accurate predictive docking methods, by bringing new ideas into the field, establishing standard quality parameters, and providing protein models and structures for benchmarking the performance of any given docking method.

Two major technical aspects can be found in the majority of docking methods: the generation of a large variety of structural models (sampling) and the identification of the correct docking poses with a proper function (scoring)[8]. Many current techniques are successful if the interacting proteins undergo only small conformational changes upon binding. Even in these conditions, docking algorithms generate a large number of incorrect docking poses, so an important part of the success depends on the accuracy of the scoring function used to evaluate the docked conformations, as well as on their

capabilities to overcome the inaccuracies of their interacting surfaces and singling out near-native conformations.[9,10] Generally speaking, scoring aims to identify the lowest-energy state among the different possible states of a given interaction, and thus, in the case of docking, it should be ideally able to describe the energetic aspects of protein-protein association.[11] For practical predictions, the energy description of a system is estimated by approximate functions, and a large variety of scoring functions have been used. Docking algorithms often rely on the geometric complementarity of protein-protein interfaces. The essential zones for binding are often preformed in the interacting proteins,[12] and as a consequence the interface of a protein complex could be considered an inherent geometric feature of the protein structures. This has made shape complementarity a popular ranking criterion to distinguish near-native solutions. Still, many protein-protein interfaces are flat, so complementarity alone is not enough to describe the right association mode. This is one of the reasons why a sampling step based only on geometry criteria often fails to produce correct models. Indeed, the physicochemical nature of the residues has a major role in protein association. Important elements include the electrostatic forces with complementary charges helping to provide the micro environment needed for the interface formation and the correct orientation of the proteins, and the hydrophobic effect with the burial of hydrophobic patches favoring the desolvation of the interacting surfaces.[13,14] Other factors are van der Waals attraction and repulsion, and hydrogen bonding. However, scoring functions that use energy-based terms to model these effects are not yet accurate enough to reliably select near-native solutions from a pool of decoys, and thus further investigation is required to improve the quality of docking predictions.

Given that docking programs typically report decoys ranked with only one or

two scoring functions, it remains to be seen whether a given method could further benefit from the accumulated knowledge derived from the variety of currently available scoring functions that have been reported in the literature, many of which were developed for different modeling problems. In a recent study,[15] more than 100 scoring functions were used to evaluate docking models generated by SwarmDock, a semi-flexible docking method. This analysis identified different scoring functions that improved the predictive rates for this method with respect to the original algorithm, and suggested new strategies to further improve the results by combining different pairs of scoring functions. Here we have extended this analysis to evaluate the performance of 73 scoring functions selected from the CCharPPI server[16] on rigid-body docking models generated by three FFT-based docking programs, ZDOCK,[17] FTDock,[18] and SDOCK,[19] for the cases of available protein-protein docking[20,21] and scoring[22] benchmarks. The results here provide a systematic assessment of the predictive capabilities and limitations of a variety of scoring functions, bring interesting insights on the risk of overtraining when developing methods for structural modeling in general, and suggest new strategies for improvement of current docking protocols.

Materials and methods

Protein-protein docking

In the present work, we have used several well-known and freely available rigid-body protein-protein docking programs, which were run with the specifications described below (default parameters otherwise). FTDock 2.0[18] was run with electrostatics on, grid cell size of 0.7 Å, and surface thickness of 1.3 Å, with a total of 10,000 docking poses generated for each case. Missing side chains of interacting proteins were reconstructed with SCWRL 3.0.[23] ZDOCK 3.0.1[24] was used to generate 54,000 docking poses, from

which only the highest-scoring 10,000 were kept for further analysis. SDOCK[19] was run as recommended by their authors, and 1,000 clustered docking models were kept for further analysis.

Protein-protein docking benchmark sets

We computed the predictive success rates of the different scoring functions and their combinations on a well-known protein-protein docking benchmark**,** for which the structures of the unbound monomers and the bound complex are available. The docking benchmark version 4.0 (BM 4.0) contains a total of 176 targets[20] while the docking benchmark update version 5.0 (BM 5.0) includes 55 additional targets.[21]

For additional validation, we also used a recently published scoring benchmark derived from models submitted in the CAPRI experiment, which contains 15 published targets from 23 assessments. The CAPRI score set benchmark[22] contains more than 19,000 protein complex models generated by 43 different predictors groups, including web servers. Only 10% of them are of acceptable quality or better according to the CAPRI criteria, with a range of 281 to 2182 decoys per case; the number of acceptable quality decoys is 835, medium quality decoys 784, and high quality decoys 479.

Evaluation of docking predictions

In order to evaluate the predictive success rate of each docking method on the BM 4.0 and BM 5.0, CAPRI quality measurements were calculated for each of the generated structures, based on the fraction of native contacts (f_{nat}), interface RMSD (IRMSD) and ligand RMSD (LRMSD) as defined by CAPRI[25] with respect to the known reference complex structures. According to CAPRI criteria, the quality of the structures are

classified as follows: incorrect [$f_{nat} < 0.1$ or (LRMSD > 10 Å and IRMSD > 4 Å)], acceptable [[$(0.1 \leq f_{nat} < 0.3)$ and (LRMSD ≤ 10 Å or IRMSD ≤ 4 Å)] or [$(f_{nat} \geq 0.3)$ and (LRMSD > 5 Å or IRMSD > 2 Å)]], medium [[$(0.3 \leq f_{nat} < 0.5)$ and (LRMSD ≤ 5.0 Å or IRMSD ≤ 2 Å)] or [$f_{nat} \geq 0.5$ and (LRMSD > 1.0 Å and IRMSD > 1.0 Å)]], or high accuracy [$f_{nat} \geq 0.5$ and (LRMSD ≤ 1 Å or IRMSD ≤ 1 Å)]. This classification was already provided by CAPRI organizers for the models in the CAPRI score set benchmark. Success rates are defined as the percentage of cases in which a near-native solution, i.e. with acceptable quality or better (following CAPRI criteria), is found within the top *N* docking models as ranked by a given scoring function.

Protein-protein scoring functions

We selected 73 scoring functions from the CCharPPI server,[16] as shown in Table S1. These functions were already described in a previous study.[15] We did not use all the scoring functions provided in the CCharPPI server due to technical limitations of the computing platform employed to perform the calculations. For clarity purposes, the majority of contact and distance-dependent residue-level potentials were originally prefixed with 'CP_', while atomic and quasi-atomic potentials were prefixed with 'AP_'.

Cardinality analysis and combination of the normalized values for re-rank

For all scoring functions we calculated the set of complexes for which an acceptable or better solution appears in the top 10 decoys when ranked by that function. Then, for each pair of scoring functions (A, B), we calculated the size of their union (eq. 1) and symmetric difference (eq. 2) sets:

$$|A \cup B| = |A| + |B| - |A \cap B| \quad (1)$$

$$|A \Delta B| = |(A \setminus B) \cup (B \setminus A)| \quad (2)$$

These measures, which combine two scoring functions, indicate the extent to which the scoring functions are successful on different subsets of the complexes, and thus they provide an estimation of their predictive success complementarity. We also explored a strategy in which scoring functions are combined not just on the basis of their ability to find top 10 solutions in different subsets of the complexes, but also on different subsets of the decoys as delineated by the docking algorithm that was used to generate them. To do this, we combined three pairs of scoring functions, where each pair was evaluated and selected on the basis of its performance on the decoys generated by each of the three docking methods. We calculated the union cardinalities for the unified pair of scoring functions between the three docking methods (eq. 3), forming triplets of scoring functions containing one unified pair used with FTDock docking models, one unified pair with ZDOCK, and one unified pair with SDOCK, this way combining up to six different scoring functions together:

$$A \cup B \cup C = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C| \quad (3)$$

where A represents a unified pair of scoring functions that performs well in FTDock, B a unified pair of scoring functions that performs well in ZDOCK, and C a unified pair of scoring functions that performs well in SDOCK.

To calculate the success rates of these combined functions, we proceeded as follows. First, we combined different energy terms from pairs or triplets of scoring

functions, selected using the above measures. To do this, we first normalized each value using the z-score method (eq. 4):

$$Zi = \frac{x - \mu}{\sigma} \quad (4)$$

where x is the value, μ the average and σ is the standard deviation. The normalized values of the scoring function pairs for a given pose were directly added and used to re-rank the list of the poses generated by each method. For combining triplets of scoring functions, we similarly summed the three z-scores. Note that this is a naive ranking and no weight optimization was undertaken.

Results and Discussion

Performance of scoring functions with different docking methods on the protein docking benchmark 4.0

We evaluated the performance of the 73 functions for the scoring of rigid-body docking poses generated for the cases in BM4. Fig. 1A shows the performance of the ten most successful functions ordered by top 10 success rate for FTDock, ZDOCK and SDOCK, respectively. In general, scoring functions provided better predictive rates when evaluating ZDOCK and SDOCK models. Interestingly, for each docking method, there were always other scoring functions that performed better than its own in-built scoring procedure. The three scoring functions that were found among the ten most successful ones for all docking methods were AP_PISA,[26] CP_TSC[27] and CP_HLPL.[28,29] The function CP_HLPL was originally developed for describing intramolecular contacts in protein structure modeling. The functions CP_TSC and AP_PISA were specifically designed for protein-protein docking using linear programing to train both functions.

CP_TSC is a coarse-grain potential with three interaction sites per residue (side-chain centroid and N and O backbone atoms), which calculates the energy of interacting pairs with a two-step potential well. AP_PISA is an atomic potential which has a three-step potential between atom pairs, and was trained using side chain refined interfaces. These two potentials showed the best performance for the three docking methods, with AP_PISA being particularly successful in evaluating docking models generated by ZDOCK and SDOCK methods, when considering both the top 1 and the top 10 success rates. One of the possible reasons for the difference in performance of the three docking methods is the high variability in the total number of near-native solutions generated by each method in all BM 4.0 cases (FTDock: 1,653; SDOCK: 18,700; ZDOCK: 37,709). This is an important factor that clearly can affect the capabilities of the scoring functions for discriminating near-native solutions from false positives. The lower enrichment in near-native docking solutions in the FTDock docking sets could in principle explain the worse performance of the scoring functions for this docking method. However, as we will discuss a few sub-sections later, this difference in performance cannot be fully explained on the basis of near-native enrichment. Actually, an alternative explanation is that some functions could have been overtrained in cases of BM 4.0, which advices to take with caution all the results above described.

[INSERT HERE FIGURE 1]

Performance of scoring functions according to protein flexibility

The predictive success of rigid-body docking is known to strongly depend on the degree of conformational change that interacting proteins undergo upon binding.[30] We evaluated here whether this is true for all scoring functions. For that, we classified the

BM 4.0 cases according to the extent of unbound-to-bound conformational changes, based on the average interface RMSD (avgeIRMSD) for unbound receptor and ligand when superimposed onto the corresponding molecules in the complex structure, defining five categories: "rigid" (avgeIRMSD $\leq$ 0.5 Å), "low-flexible" (0.5 Å < avgeIRMSD $\leq$ 1 Å) , "medium-flexible" (1 Å < avgeIRMSD $\leq$ 2 Å), "flexible" (2 Å < avgeIRMSD $\leq$ 3 Å), and "highly-flexible" (avgeIRMSD >3 Å).

Fig. 1A shows the top 10 success rates for the above analyzed scoring functions on models generated by each docking method, for cases classified according to unbound-to-bound conformational changes. In general, for each combination of scoring function and docking method, the best success rates are obtained for the rigid cases, as expected, and the performance decreases for the most flexible cases. However, there are interesting exceptions. For instance, AP_PISA on ZDOCK models provided better performance on the medium and flexible cases than on the low-flexible ones, and almost as good as on the rigid ones. Similarly, the performance of CP_HLPL on ZDOCK was independent on the flexibility category. Interestingly, a few scoring functions with a specific docking method identified acceptable docking models within the top 10 decoys for the highly-flexibly cases. These are extremely challenging cases for rigid-body docking prediction, so the fact that selected scoring functions are able to predict some of these cases is quite encouraging. However, due to the smaller number of flexible cases in the benchmark these differences are not statistically significant (Wilcoxon signed rank test: FTDock *p*-value 0.333, ZDOCK *p*-value 0.667, SDOCK *p*-value 0.333). Only 6% (11 cases) of the BM 4.0 correspond to the highly-flexible category, containing the monomers that undergo the biggest conformational changes upon binding. In general, the performance of the different scoring functions on the rigid cases shows more

consistency, while that on the most flexible cases shows more variability, which suggests possible random effects on the latter due to lower signal-to-noise ratios. However, as above discussed, all these results should be taken with caution because of the possibility of overtraining in cases of BM 4.0, as discussed a few sub-sections later.

Performance of scoring functions according to binding affinity

The predictive performance of rigid-body docking also strongly depends on the binding affinity of the complex.[31] High-affinity complexes are in general predicted with higher accuracy. We explore here to what extent the performance of different scoring functions depend on the affinity of the complexes. We gathered the experimental binding affinity data from the Structural Affinity Benchmark[32] for 125 cases of the protein-protein docking BM 4.0, classified according their binding ΔG value as "Strong" (ΔG ≤ -12 kcal/mol) or "Weak" (ΔG > -12 kcal/mol).

Fig. 1A provides the success rates for the top 10 predictions of the previously analyzed scoring functions for the different docking methods on the benchmark BM 4.0 cases as classified by binding affinity. In general, predictive performance on the strong affinity cases is better than on the weak affinity cases. However, there are some exceptions, being the most notable ones the SIPPER[33] and PROPNSTS[34] functions when evaluating FTDock models, which yielded much better predictions for the weak affinity cases. Interestingly, these two scoring functions are based on the same residue potentials derived from protein-protein complex structures. It seems that they are able to capture the binding energy determinants of weak complexes better than other atomistic potentials. Again, all these results should be taken with caution because of the possibility of overtraining in cases of BM 4.0, as discussed a few sub-sections later.

Performance of scoring functions on the CAPRI score set benchmark

We evaluated the performance of the 73 scoring functions on the CAPRI score set benchmark, which is formed by 15 targets from the CAPRI experiment,[22] for which a range of docking models were blindly generated by a variety of docking methods (see Methods). Fig. 2A shows the predictive rates for the best 30 scoring functions in this benchmark according to the top 10 success rate.

Many of these scoring functions overlap with those that perform well on the BM 4.0, such as AP_T1,[35] AP_T2,[35] CP_DECK,[36] CP_TB,[37] CP_TSC and AP_PISA. Interestingly, the best success rates for the top 100 predictions were obtained by coarse-grain potentials, in general. Perhaps coarse-grained potentials are providing a more balanced score that is more adequate to the heterogeneity of docking models generated by the large variety of docking methods in the CAPRI score set benchmark. The most successful function for the top 100 predictions is CP_TB, a scoring function designed for docking, which was among the most successful ones with FTDock on the BM 4.0.

[INSERT HERE FIGURE 2]

Performance of scoring functions with different docking methods on the protein docking benchmark 5.0 update

In order to confirm the previous findings regarding the good success rates observed for some of the scoring functions on specific docking methods, such as AP_PISA on ZDOCK and SDOCK, we also evaluated the performance of the different scoring functions and docking methods on the recently available BM 5.0 update,[21] formed by

cases that were not present in BM 4.0. This analysis provided unexpected and interesting results. Fig. 1B shows the success rates of the ten most successful functions (when considering the top 10 predictions) for FTDock, ZDOCK and SDOCK, on the BM 5.0 update. We can observe that the best scoring functions now are different from the best scoring functions of BM 4.0 (Fig. 1A), especially for ZDOCK and SDOCK, in which their performance in general is much lower. One of the most striking differences is AP_PISA, which shows much lower performance than on BM 4.0, and may be indicative of overfitting to the BM 4.0 complexes during the development of such functions. This may also explain some of the other differences observed**.**

Fig. S1 shows the performance on the BM 5.0 update of the best-performing functions resulting of the previous BM 4.0 analysis. The predictive rates of these functions for the BM 5.0 update are much lower than those observed for the BM 4.0 cases. In addition, there are now fewer differences in the best predictive rates for the different docking methods. Indeed, now the evaluation of ZDOCK and SDOCK docking models does not show better success rates than FTDock as is the case for the BM4.0. The performance for the scoring functions on the FTDock models, with similar success rates on both BM 4.0 and 5.0, is more consistent than that of ZDOCK and SDOCK. Interestingly, according to the total number of near-native solutions generated by each method in all BM 5.0 cases (FTDock: 454; SDOCK: 5,597; ZDOCK: 7,726) the enrichment of BM 5.0 in near-native solutions is similar to that of BM 4.0, which suggests that the lower performance of FTDock in BM 4.0 could not be fully explained on the basis of near-native enrichment. In fact, it seems that the performance obtained for some scoring functions on BM 4.0 with ZDOCK and SDOCK were excessively high. One reason could be that these scoring functions might have been overtrained on

cases from BM 4.0 during their development, using ZDOCK and SDOCK methods to generate docking decoys. Another explanation could be that SDOCK is technically similar to ZDOCK, and overtraining in any of these two methods could also affect the other one.

The BM 5.0 update provides a set of cases that were not used for training, since it does not include complexes from previous benchmark sets. A key question is whether the best-performing scoring functions for BM 5.0 update represent bona fide success rates for docking in general or they appear good only for this particular set of cases for another reason. The fact that CP_HLPL and CP_TB are found among the best-performing scoring functions with the three docking methods on BM 5.0 update, suggests that their good performance on BM 4.0 was not due to overtraining, and therefore they could be of more general applicability for new cases. Indeed, CP_HLPL, which used with SDOCK provided the best top 10 success rate among all functions (25%), was originally developed from intramolecular contacts for modeling protein monomers. On the other side, CP_TB was developed for docking but trained in a composite set of representative transient complexes. This knowledge-based potential was designed to tolerate small changes in side chain orientations, which may contribute to its avoidance of overtraining.

Performance of scoring functions according to binding affinity and flexibility on BM 5.0 update cases

We have analyzed the results on the cases in BM 5.0 update when classified according to unbound-to-bound conformational flexibility (Fig. 1B). Several functions (CP_TB with FTDock and SDOCK models; CP_BT, CP_BFKV and CP_SKOa with FTDock,

etc.) can provide similarly good performance for rigid and low-flexible cases.

We also analyzed the results for the 35 cases of the BM 5.0 update for which there is experimental binding affinity available.[21] These cases were classified as strong or weak, according to their experimental binding affinity (Fig. 1B). The affinity-dependent performance of some of the scoring functions varies according to the docking method. For instance CP_HLPL shows no dependence on affinity with SDOCK, but strong dependence with FTDock. The performance of some of the functions for the strong binders in the BM 5.0 update is better than those in the BM 4.0, perhaps due to the fact that the BM 5.0 has fewer cases with affinity information.

Scoring performance on models merged from different docking methods

We merged all docking models generated by the three docking methods into a single decoy set, and evaluated the performance of each scoring function on this heterogeneous pool of docking solutions. Fig. 2B shows the performance for the best 30 scoring functions on this set ordered by top 10 success rates. In general, the success rates for the best performing functions were lower than those obtained with the individual methods. For instance, the best performing scoring function on the merged pool of docking models is CP_TB, with 24% success rate for the top 10 predictions, while for the individual methods, CP_HLPL with SDOCK, and CP_BFKV[38] with FTDock yielded higher success rates (over 25%). Surprisingly, these scoring functions yielded much lower success rates on the large docking set (20% and 12%, respectively).

This shows that some scoring functions are particularly efficient for a specific docking method, which suggests that it would be more reasonable to use each docking

method only with the scoring functions that have shown the best performance on such method. A different question is choosing the most adequate scoring function when we do not know which docking method was used to build each docking model. In this case, a good scoring function that could work for a particular method (i.e. CP_BFKV on SDOCK and FTDock) might give worse predictive rates in other docking method (i.e. CP_BFKV on ZDOCK). In this situation, it would be better to choose a more general scoring function that could provide good success rates in all methods according to our tests here (i.e. CP_TB or PYDOCK_TOT). This could be relevant in the CAPRI scorers experiment, for instance, in which a variety of docking models need to be scored, but no information is provided on how they were generated. However, as above described, currently available scoring functions show worse performance in heterogeneous sets. This could be due to the fact that most of them have been developed in homogeneous data sets. Therefore, it would be important to use heterogeneous data sets, such as the CAPRI score set benchmark, for developing and testing new scoring functions that could be of general applicability and thus not so dependent on a particular docking method.

Performance of combined scoring functions

We next explored whether the combination of scoring functions might improve the predictive rates. First, we identified pairs of scoring functions that provided successful results in complementary subsets of complexes. The first metric we used to do this is the size of the combined set of complexes for which an acceptable or better solution was found in the top 10 by either of the scoring functions (union cardinality). The second metric was similar, but excluding the complexes that are identified by both functions (symmetric difference cardinality). These measures were chosen to give an indication of

how both scoring functions bolster each other, and therefore, this could be used as an estimation of the potential synergistic effect of the two functions when combined. Fig. 3 shows the cardinality values (for top 10 predictions) for the combinations of the ten functions with the greatest union values when paired, for each of the docking methods on the BM 5.0 update. Fig. S2 shows these values for all pairs of scoring functions. We can observe that some pairs of scoring functions are highly complementary, since they are able to capture near-native solutions on non-overlapping sets of complexes (e.g. PYDOCK_TOT/CP_BFKV with FTDock; PYDOCK_TOT/AP_T2 with ZDOCK; AP_MPS/SDOCK or AP_MPS/CP_RMFCEN1 with SDOCK).

[INSERT HERE FIGURE 3]

From the above analysis, one could estimate the most favorable pairs of scoring functions, i.e. those ones that when combined would be expected to yield improved success rates. Therefore, we tested the predictive power of the cardinality analysis. For this, we normalized the energy values obtained from each pair of functions and converted them into z-scores. Then we added these values without weighting and used them to re-rank all the generated decoys for each case. Fig. 4 shows the predictive rates (on BM 5.0 update) for the combinations of the ten scoring functions that provided the largest union values (for top 10 predictions) on the BM 5.0 update. Some combinations yielded >30% success rates for FTDock models (as compared with 20-25% for the individual scoring functions). However, in the case of ZDOCK and SDOCK docking methods, success rates of the best combined scoring functions did not improve the individual ones. This small improvement in the success rates for a few combinations of scoring functions is not sufficient to guarantee that this strategy could be of general

applicability to a new set of cases, and requires further investigation.

[INSERT HERE FIGURE 4]

For some pairs of scoring functions, the cardinality analysis did not reflect well the success rate values after rescoring with the combined functions. For instance, with FTDock, the best union was found for PYDOCK_TOT/CP_BFKV, providing near-native models for 42% of the cases within the top 10 predictions, but the combined functions have a top 10 success rate of 27%. Individually, PYDOCK_TOT has a top 10 success rate of 20% and CP_BFKV of 26%. For some reason PYDOCK_TOT seems to contribute little at the combined success rate in the top 10 predictions in spite of the observed high cardinality. On the other hand, the best top 10 success rates after rescoring with the combined functions is provided by CP_SJKG/AP_dDFIRE (33%), while individually, CP_SJKG and AP_dDFIRE have much lower success rates (16% and 20%, respectively). For this pair, the union was not among the best values of all cases, so cardinality analysis was not able to foresee the strong synergy shown by the combination of these two scoring functions.

So far, we selected the top scoring functions for each docking method in BM 5.0 update and evaluated its performance in the BM 5.0 itself. To make a blind test, we selected the ten scoring functions with the best top 10 success rates from BM 4.0, and computed their cardinalities on BM 5.0 update (Fig. S3). With FTDock the best cardinalities are found for combined pairs involving PYDOCK_TOT, being the highest ones the combinations with CP_HLPL and CP_TB. With ZDOCK there are many combinations that give a high cardinality, such as the combination of PYDOCK_TOT

with AP_T1/2 or AP_PISA. Fig. S4 shows the top 10 success rates on BM 5.0 for the best scoring function pairs formed by unweighted combination based on z-scores, using the ten scoring functions that showed better performance from BM 4.0. We found two pairs of combinations that reached a success rate above 30% within the top 10 predictions: the pair PYDOCK_TOT/CP_HLPL with FTDock (31%), and the pair AP_PISA/CP_HLPL with SDOCK (31%). Thus, combined pairs formed by scoring functions selected on the basis of BM 4.0 yielded success rates on the external test set BM 5.0 as high as those obtained when the combined pairs were formed by functions selected among the best ones in BM 5.0 update. The fact that the pairs of functions selected either using the BM 4.0 or the BM 5.0 provided similar success rates on the BM 5.0, suggested that the selection of pairs on the basis of BM 5.0 had little or no overfitting.

Overall, this is not a considerable increase in the success rate. To extend the number of existing near-native solutions and possibly improve the scoring performance, a heterogeneous pool of decoys could be created from the three docking methods and the best scoring functions for each docking method. In fact, a researcher is not limited to use only one docking method, e.g. the complementarity of ZDOCK and FTDock both using the PYDOCK_TOT scoring was used to help to model yeast interactome.[39] In this line, we aimed to combine the pairs of scoring functions that performed well on each set of docking decoys generated by FTDock, ZDOCK and SDOCK, and tried to evaluate whether they would improve the predictive results. For this, we built scoring function pairs formed by unweighed combinations based on z-scores, using the ten scoring functions that provided the best top 10 success rates for each docking method in BM 4.0. With them, we built triplets of combinations formed by one pair of scoring

functions from each docking method, and computed the union cardinality (for the top 10 predictions) for each triplet on BM 5.0. Table S2 shows the combined triplets with the 50 best union cardinalities. The best triplet combinations generated by this strategy captured 30 cases (55%), considerably more than the 18 cases (33%) predicted by the best-performing pairs of scoring functions (CP_SJKG and AP_dDFIRE with FTDock) from the cardinality analysis carried out with the individual docking methods. According to these results, the use of triplet combinations of the best pairs of functions for each method seemed to anticipate a large improvement in success rates. To confirm this, we used the best scoring function pairs for each method (according to BM 4.0), and computed the success rates of the triplet combinations of function pairs / docking methods on BM 5.0 (Fig. S5). The best triplet combination is formed by PYDOCK_TOT and CP_HLPL with FTDock, AP_T2 and AP_PISA with ZDOCK, and AP_calRWp and SDOCK scoring function with SDOCK (38% success rate). However, despite the expectances, this is not much better than the best performance we found for a pair of scoring functions (CP_SJKG/AP_dDFIRE with FTDOCK; top 10 success rate 33% on BM 5.0).

The combination of scoring functions performed here was based on a direct addition of the normalized functions. There was no attempt to improve the combination of values, by optimization of parameters, multi-parametric fitting, etc. However, due to the process of selection of scoring functions, there could be a possible bias towards the best-performing functions on the BM 5.0. The use of more sophisticated approaches to combine the scoring functions could yield better predictive rates, but such analysis should be done with caution, to minimize the risk of overfitting, for instance by putting feature selection within an outer cross-validation wrapper.

Conclusion

We performed here a systematic analysis of the performance of 73 known functions for the scoring of rigid-body docking poses generated with different docking methods on a standard protein-protein docking benchmark. From a first analysis on an existing protein-protein docking benchmark (BM 4.0), we initially found that some of the functions provided much better predictive rates than those from the original functions used in each method. However, when they were evaluated in a new, independent set of protein-protein docking cases (BM 5.0 update), success rates for these functions were significantly lower, which suggested that much of the observed improvement in the first analysis could have been due to overtraining. In this external set, the performance of some scoring functions was highly dependent on each type of docking method, so the most logical approach would be to use the most appropriate scoring function for a given docking method. However, a few scoring functions were sufficiently robust to different types of docking methods, which can be of interest when evaluating a heterogeneous pool of docking models generated by a variety of methods. Finally, the combination of different scoring functions looks promising to obtain better predictive rates, but this should be carefully done in order to avoid overtraining.

REFERENCES

1. Jones S, Thornton JM. Principles of protein-protein interactions. Proc Natl Acad Sci USA 1996;93:13–20.
2. Stumpf MPH, Thorne T, Silva E de, Stewart R, An HJ, Lappe M, Wiuf C. Estimating the size of the human interactome. Proc Natl Acad Sci USA 2008;105:6959–6964.
3. Rolland T, Taşan M, Charloteaux B, Pevzner SJ, Zhong Q, Sahni N, Yi S, Lemmens I, Fontanillo C, Mosca R, Kamburov A, Ghiassian SD, Yang X, Ghamsari L, Balcha D, Begg BE, Braun P, Brehme M, Broly MP, Carvunis A-R, Convery-Zupan D, Corominas R, Coulombe-Huntington J, Dann E, Dreze M, Dricot A, Fan C, Franzosa E, Gebreab F, Gutierrez BJ, Hardy MF, Jin M, Kang S, Kiros R, Lin GN, Luck K, MacWilliams A, Menche J, Murray RR, Palagi A, Poulin MM, Rambout X, Rasla J, Reichert P, Romero V, Ruyssinck E, Sahalie JM, Scholz A, Shah AA, Sharma A, Shen Y, Spirohn K, Tam S, Tejeda AO, Trigg SA, Twizere J-C, Vega K, Walsh J, Cusick ME, Xia Y, Barabási A-L, Iakoucheva LM, Aloy P, Rivas J De Las, Tavernier J, Calderwood MA, Hill DE, Hao T, Roth FP, Vidal M. A proteome-scale map of the human interactome network. Cell 2014;159:1212–1226.
4. Janin J. Principles of protein-protein recognition from structure to thermodynamics. Biochimie 1995;77:497–505.
5. De S, Krishnadev O, Srinivasan N, Rekha N. Interaction preferences across protein-protein interfaces of obligatory and non-obligatory components are different. BMC Struct Biol 2005;5:15-18.
6. Mosca R, Céol A, Aloy P. Interactome3D: adding structural details to protein

networks. Nat Methods 2013;10:47–53.

7. Janin J. Assessing predictions of protein-protein interaction: the CAPRI experiment. Protein Sci 2005;14:278–283.
8. Huang S-Y. Search strategies and evaluation in protein-protein docking: principles, advances and challenges. Drug Discov Today 2014;19:1081–1096.
9. Halperin I, Ma B, Wolfson H, Nussinov R. Principles of docking: An overview of search algorithms and a guide to scoring functions. Proteins 2002;47:409–443.
10. Vajda S, Kozakov D. Convergence and combination of methods in protein-protein docking. Curr Opin Struct Biol 2009;19:164–170.
11. Moal IH, Fernandez-Recio J. SKEMPI: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. Bioinformatics 2012;28:2600–2607.
12. Levy ED. A Simple Definition of Structural Regions in Proteins and Its Use in Analyzing Interface Evolution. J Mol Biol 2010;403:660–670.
13. Camacho CJ, Vajda S. Protein docking along smooth association pathways. Proc Natl Acad Sci 2001;98:10636–10641.
14. Camacho CJ, Weng Z, Vajda S, DeLisi C. Free Energy Landscapes of Encounter Complexes in Protein-Protein Association. Biophys J 1999;76:1166–1178.
15. Moal IH, Torchala M, Bates P, Fernandez-Recio J. The scoring of poses in protein-protein docking: current capabilities and future directions. BMC Bioinformatics 2013;14:286.
16. Moal IH, Jiménez-García B, Fernández-Recio J. CCharPPI web server: computational characterization of protein-protein interactions from structure. Bioinformatics 2015;31:123–125.
17. Chen R, Li L, Weng Z. ZDOCK: an initial-stage protein-docking algorithm.

Proteins 2003;52:80–87.

18. Gabb HA, Jackson RM, Sternberg MJ. Modelling protein docking using shape complementarity, electrostatics and biochemical information. J Mol Biol 1997;272:106–120.
19. Zhang C, Lai L. SDOCK: A global protein-protein docking program using stepwise force-field potentials. J Comput Chem 2011;32:2598–2612.
20. Hwang H, Vreven T, Janin J, Weng Z. Protein-protein docking benchmark version 4.0. Proteins 2010;78:3111–3114.
21. Vreven T, Moal IH, Vangone A, Pierce BG, Kastritis PL, Torchala M, Chaleil R, Jiménez-García B, Bates PA, Fernandez-Recio J, Bonvin AMJJ, Weng Z. Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. J Mol Biol 2015;427:3031-3041.
22. Lensink MF, Wodak SJ. Score_set: a CAPRI benchmark for scoring protein complexes. Proteins 2014;82:3163–3169.
23. Canutescu AA, Shelenkov AA, Dunbrack RL. A graph-theory algorithm for rapid protein side-chain prediction. Protein Sci 2003;12:2001–2014.
24. Pierce BG, Hourai Y, Weng Z. Accelerating Protein Docking in ZDOCK Using an Advanced 3D Convolution Library. PLoS ONE 2011;6:e24657.
25. Lensink MF, Méndez R, Wodak SJ. Docking and scoring protein complexes: CAPRI 3rd Edition. Proteins 2007;69:704–718.
26. Viswanath S, Ravikant DV, Elber R. Improving ranking of models for protein complexes with side chain modeling and atomic potentials. Proteins 2012;81:592–606.
27. Tobi D, Bahar I. Optimal design of protein docking potentials: efficiency and limitations. Proteins 2006;62:970–981.

28. Park B, Levitt M. Energy functions that discriminate X-ray and near native folds from well-constructed decoys. J Mol Biol 1996;258:367–392.
29. Pokarowski P, Kloczkowski A, Jernigan RL, Kothari NS, Pokarowska M, Kolinski A. Inferring ideal amino acid interaction forms from statistical protein contact potentials. Proteins 2005;59:49–57.
30. Pons C, Grosdidier S, Solernou A, Pérez-Cano L, Fernández-Recio J. Present and future challenges and limitations in protein-protein docking. Proteins 2010;78:95–108.
31. Vajda, S. Classification of protein complexes based on docking difficulty. Proteins 2005;60:176–180.
32. Kastritis PL, Moal IH, Hwang H, Weng Z, Bates PA, Bonvin AM, Janin J. A structure-based benchmark for protein-protein binding affinity. Protein Sci 2011;20:482–491.
33. Pons C, Talavera D, Cruz X de la, Orozco M, Fernandez-Recio J. Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein-protein docking. J Chem Inf Model 2011;51:370–377.
34. Liu S, Zhang C, Zhou H, Zhou Yi. A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. Proteins 2004;56:93–101
35. Tobi D. Designing coarse grained-and atom based-potentials for protein-protein docking. BMC Struct Biol 2010;10:40.
36. Liu S, Vakser IA. DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking. BMC Bioinformatics 2011;12:280.
37. Tobi D. Designing coarse grained-and atom based-potentials for protein-protein

docking. BMC Struct Biol 2010;10:40.

38. Feng Y, Kloczkowski A, Jernigan RL. Potentials “R” Us web-server for protein energy estimations with coarse-grained knowledge-based potentials. BMC Bioinformatics 2010;11:92.

39. Mosca R, Pons C, Fernández-Recio J, Aloy P. Pushing Structural Information into the Yeast Interactome by High-Throughput Protein Docking Experiments. PloS Comput Biol. 2009 Aug 28;5:e1000490.

FIGURE LEGENDS

Figure 1. Performance of scoring functions on (A) BM 4.0, and (B) BM 5.0. The first three columns show the success rates for the ten best performing scoring functions for each docking method, for the top 1, 10 and 100 predictions. Only the ten best performing scoring functions according to top 10 success rates are shown. Columns F1-F5 show success rates for the top 10 predictions according to conformational changes upon binding (F1: rigid; F2: low-flexible; F3: medium-flexible; F4: flexible, and F5: highly-flexible; see Methods). The two last columns show the success rates for the top 10 predictions according to binding affinity (see Methods).

Figure 2. Performance of scoring functions on heterogeneous docking sets. (A) Success rates on the CAPRI score set benchmark. (B) Success rates on the merged docking sets from the three docking methods in BM 5.0 update. Only the 30 best performing scoring functions are shown.

Figure 3. Cardinality analysis of pairs of scoring functions on BM 5.0. The heat-maps show the union (left panels) and symmetric difference (right panels) values for pair combinations of the ten scoring functions that provided the highest union values (top 10 predictions) for each docking method on BM 5.0 update, with functions grouped using single linkage clustering.

Figure 4. Success rates on BM 5.0 for pair combinations of scoring functions using z-scores. Performance on BM 5.0 of scoring function pairs formed by unweighted combination based on z-scores, using the ten scoring functions that provided the best

union values (for top 10 predictions) on BM 5.0 for each docking method. The ten pairs of scoring functions with the best top 10 success rates are shown for each docking method: A) FTDock, B) ZDOCK, and C) SDOCK.

SUPPLEMENTARY FIGURE LEGENDS

Figure S1. Performance on the BM 5.0 update for the ten best-performing functions from BM 4.0. For each docking method, the ten scoring functions with the best top 10 success rates on BM 4.0 were selected, and their performance on BM 5.0 update is shown.

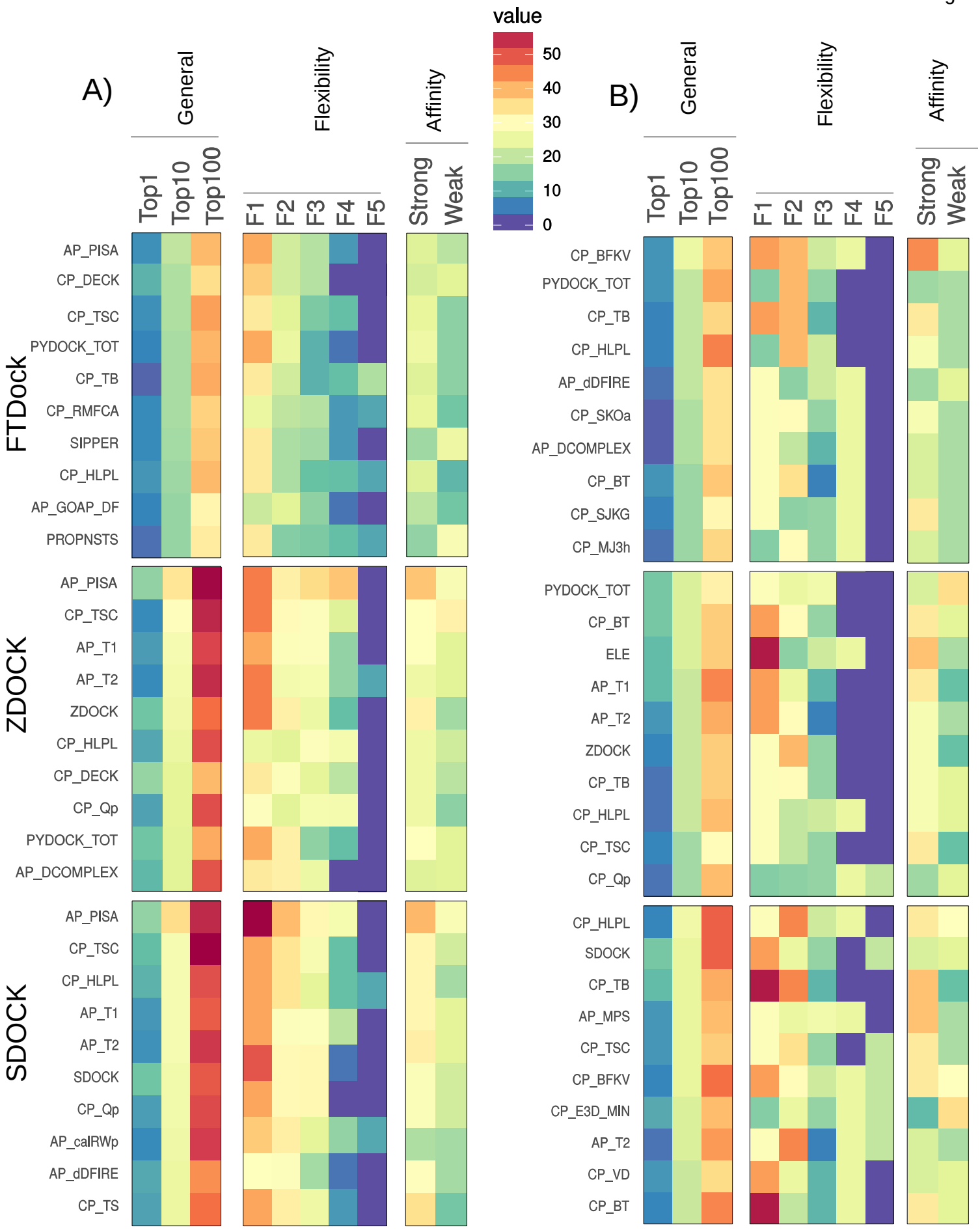
Figure S2. Cardinality analysis for all pairs of scoring functions on BM 5.0. The heat-maps show the cardinality values (union in left panels; symmetric difference in right panels) for all scoring function pairs using the top 10 predictions, with functions grouped using single linkage clustering, for FTDock, ZDOCK and SDOCK.

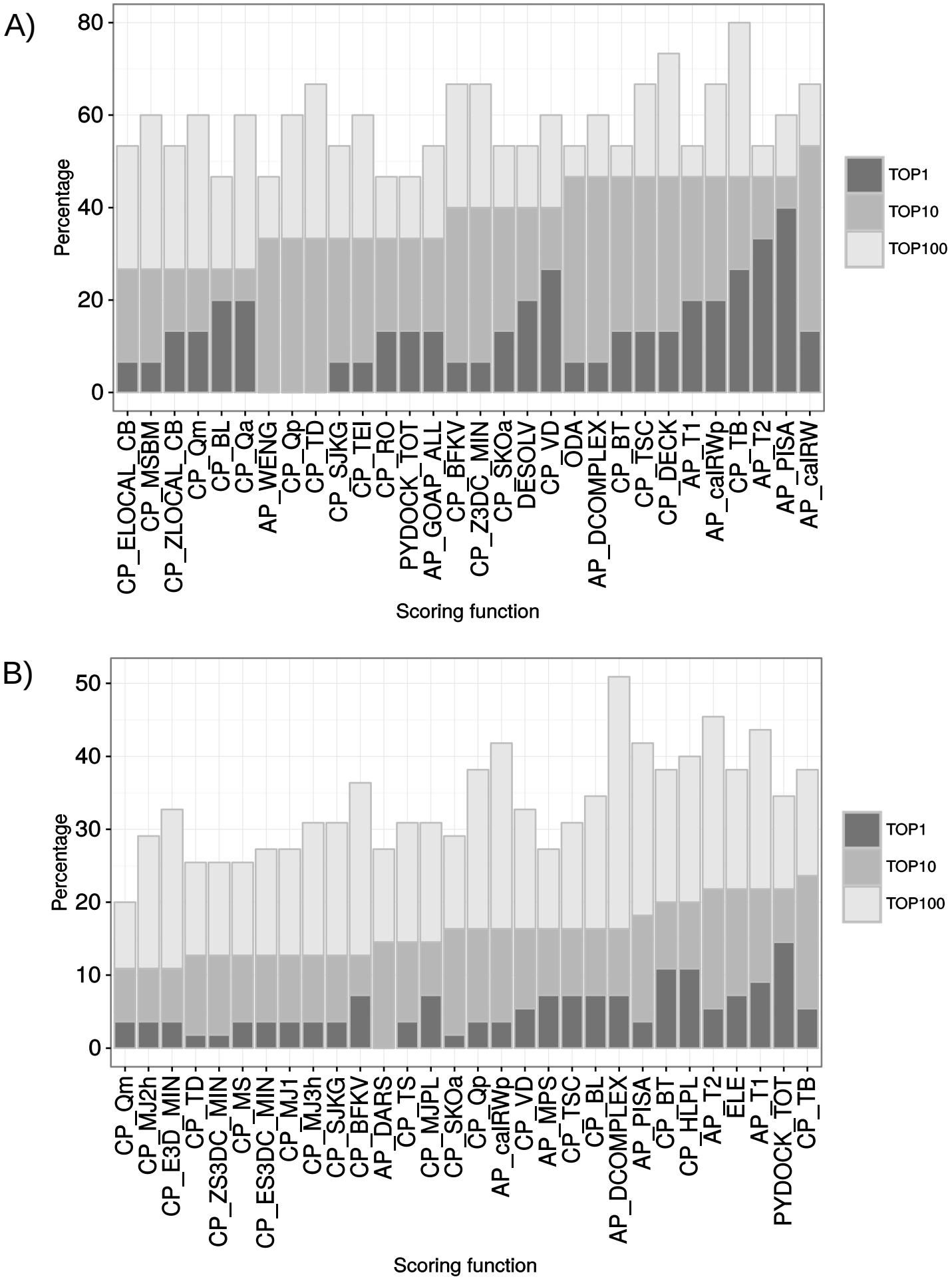
Figure S3. Cardinality analysis on the BM 5.0 update for pair combinations of the best-performing scoring functions from BM 4.0. The heat-maps show the cardinality values (union in left panels; symmetric difference in right panels) on BM 5.0 for pair combinations of the ten most successful (top 10 predictions) scoring functions from BM 4.0, with functions grouped using single linkage clustering.

Figure S4. Success rates on BM 5.0 for pair combinations of the best-performing scoring functions from BM 4.0. Performance on BM 5.0 of scoring function pairs formed by unweighted combination based on z-scores, using the ten most successful (top 10 predictions) scoring functions from BM 4.0, for each docking method*:* A) FTDock, B) ZDOCK, and C) SDOCK.

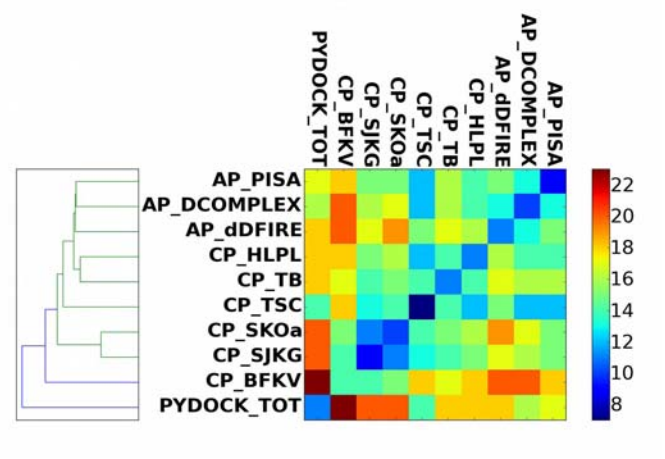
Figure S5. Success rates on BM 5.0 for triplet combinations of the best performing

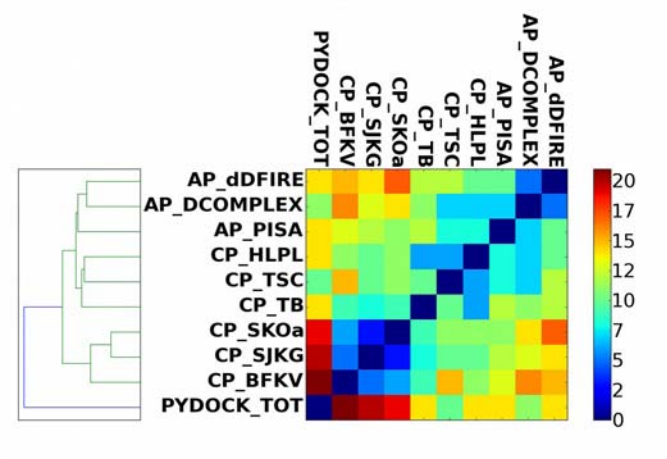
scoring functions and docking methods from BM 4.0. Performance on BM 5.0 of the triplets formed by unweighted combination of scoring functions (z-scores) with each of the docking methods, using the ten most successful (top 10 predictions) scoring functions from BM 4.0 for each docking method.


A)
B)
value
50
40
30
20
10
0
General
Flexibility
Affinity
Top1
Top10
Top100
F1
F2
F3
F4
F5
Strong
Weak
FTDock
AP_PISA
CP_DECK
CP_TSC
PYDOCK_TOT
CP_TB
CP_RMFCA
SIPPER
CP_HLPL
AP_GOAP_DF
PROPNSTS
ZDOCK
AP_PISA
CP_TSC
AP_T1
AP_T2
ZDOCK
CP_HLPL
CP_DECK
CP_Qp
PYDOCK_TOT
AP_DCOMPLEX
SDOCK
AP_PISA
CP_TSC
CP_HLPL
AP_T1
AP_T2
SDOCK
CP_Qp
AP_calRWp
AP_dDFIRE
CP_TS
CP_BFKV
PYDOCK_TOT
CP_TB
CP_HLPL
AP_dDFIRE
CP_SKOa
AP_DCOMPLEX
CP_BT
CP_SJKG
CP_MJ3h
PYDOCK_TOT
CP_BT
ELE
AP_T1
AP_T2
ZDOCK
CP_TB
CP_HLPL
CP_TSC
CP_Qp
CP_HLPL
SDOCK
CP_TB
AP_MPS
CP_TSC
CP_BFKV
CP_E3D_MIN
AP_T2
CP_VD
CP_BT

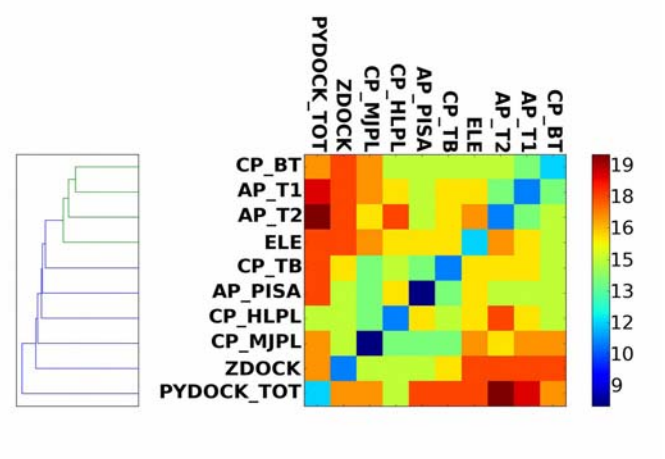

A)
Percentage
80
60
40
20
0
CP_ELOCAL_CB
CP_MSBM
CP_ZLOCAL_CB
CP_Qm
CP_BL
CP_Qa
AP_WENG
CP_Qp
CP_TD
CP_SJKG
CP_TEI
CP_RO
PYDOCK_TOT
AP_GOAP_ALL
CP_BFKV
CP_Z3DC_MIN
CP_SKOa
DESOLV
CP_VD
ODA
AP_DCOMPLEX
CP_BT
CP_TSC
CP_DECK
AP_T1
AP_calRWp
CP_TB
AP_T2
AP_PISA
AP_calRW
Scoring function
TOP1
TOP10
TOP100
B)
Percentage
50
40
30
20
10
0
CP_Qm
CP_MJ2h
CP_E3D_MIN
CP_TD
CP_ZS3DC_MIN
CP_MS
CP_ES3DC_MIN
CP_MJ1
CP_MJ3h
CP_SJKG
CP_BFKV
AP_DARS
CP_TS
CP_MJPL
CP_SKOa
CP_Qp
AP_calRWp
CP_VD
AP_MPS
CP_TSC
CP_BL
AP_DCOMPLEX
AP_PISA
CP_BT
CP_HLPL
AP_T2
ELE
AP_T1
PYDOCK_TOT
CP_TB
Scoring function
TOP1
TOP10
TOP100

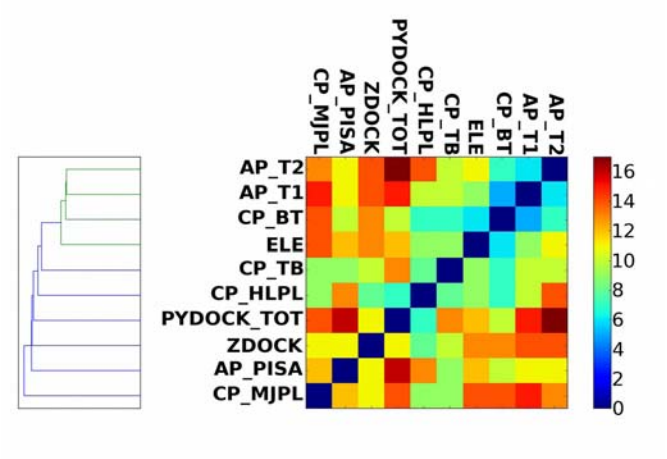
A) FTDock



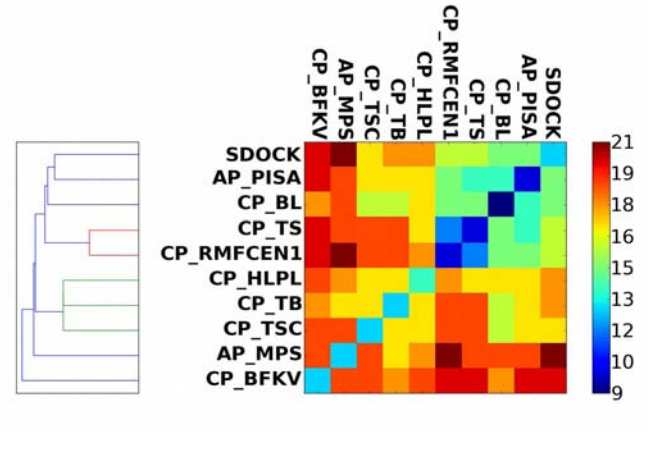


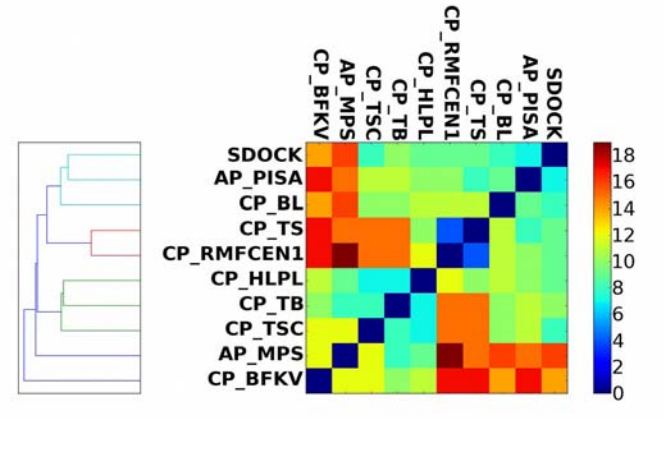
B) ZDOCK

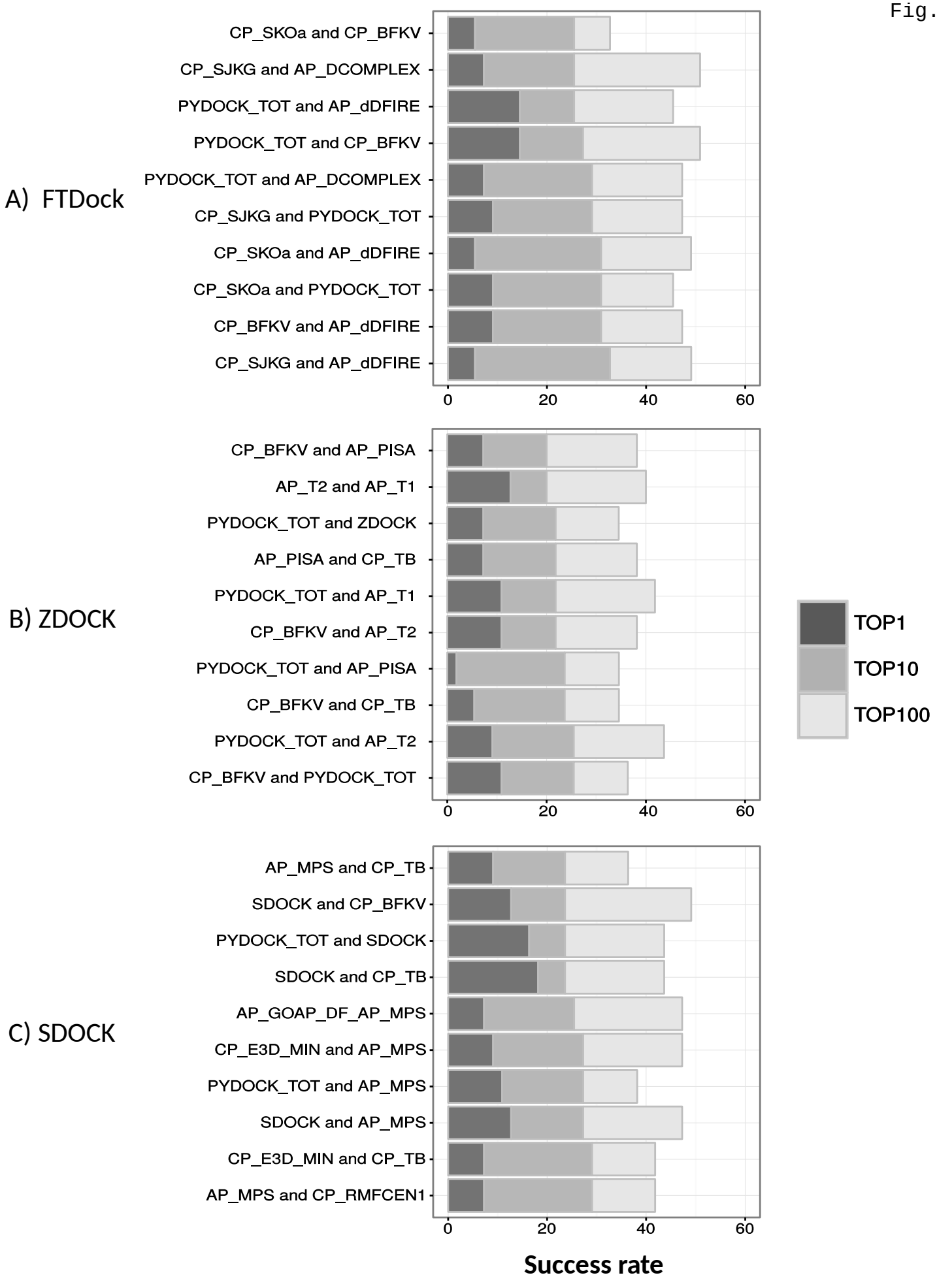


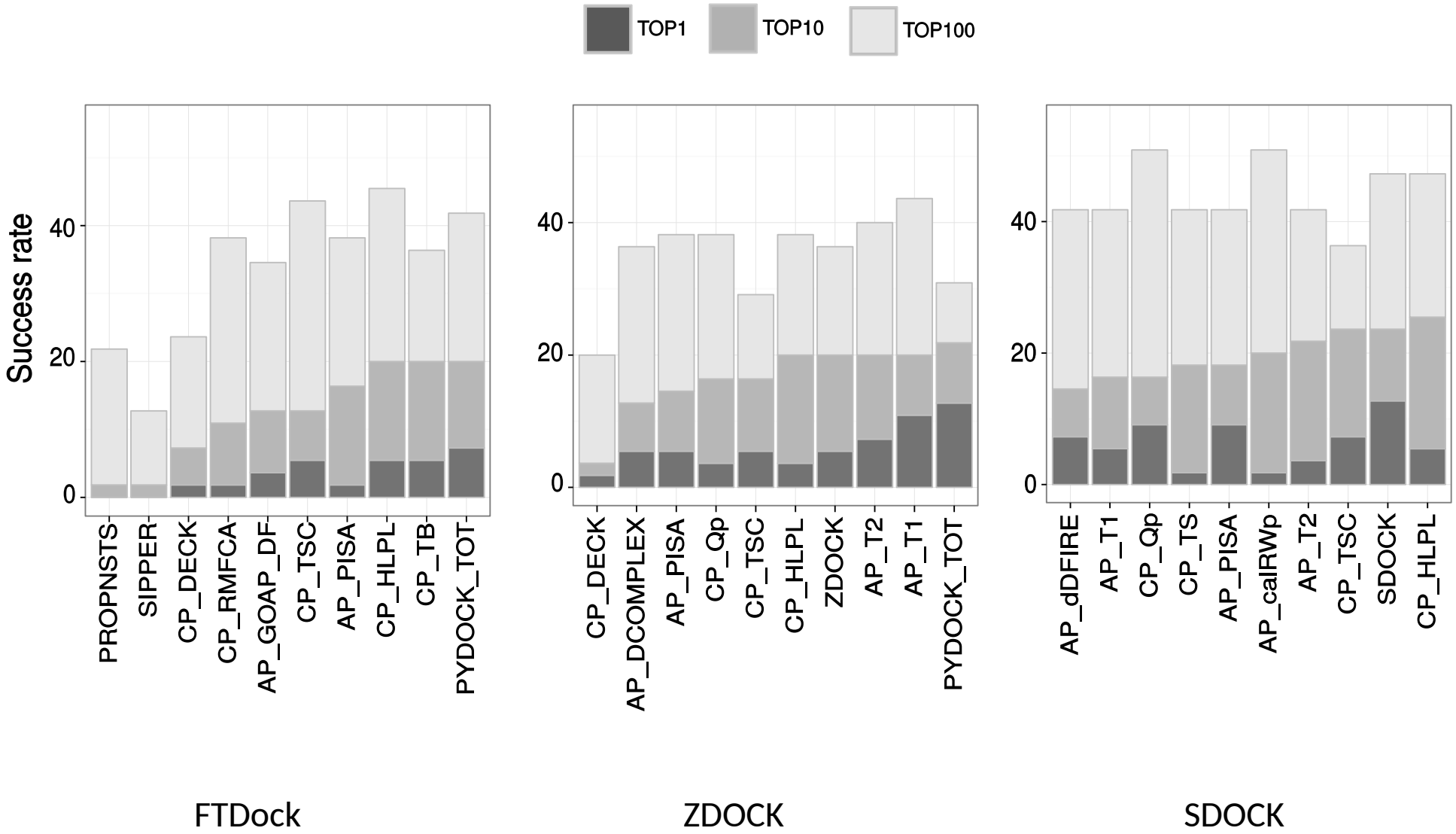


C) SDOCK

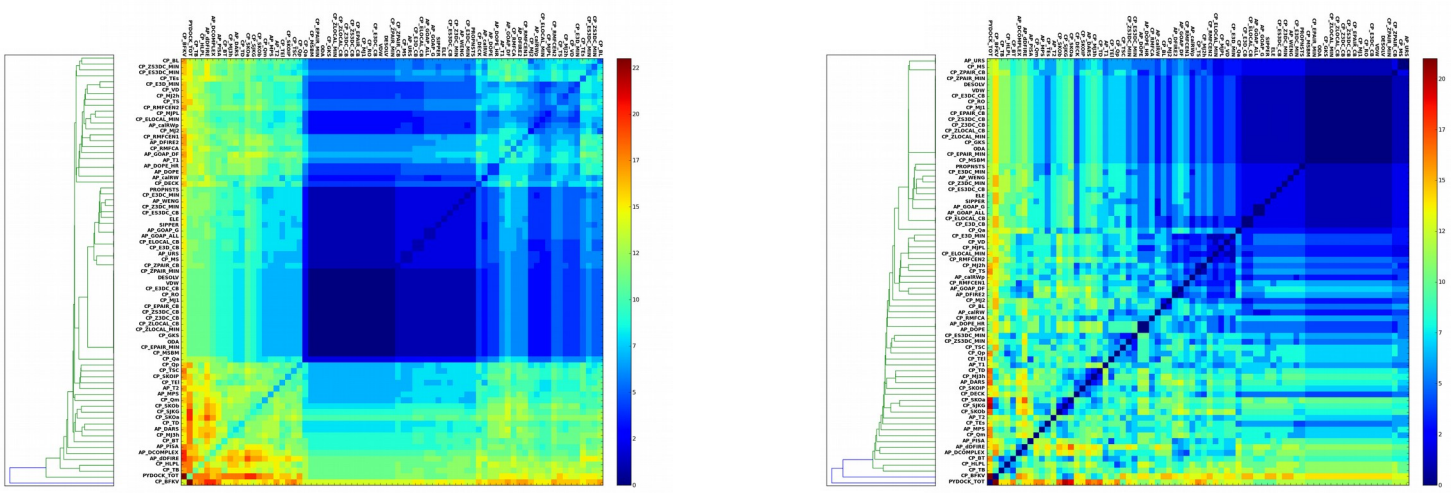




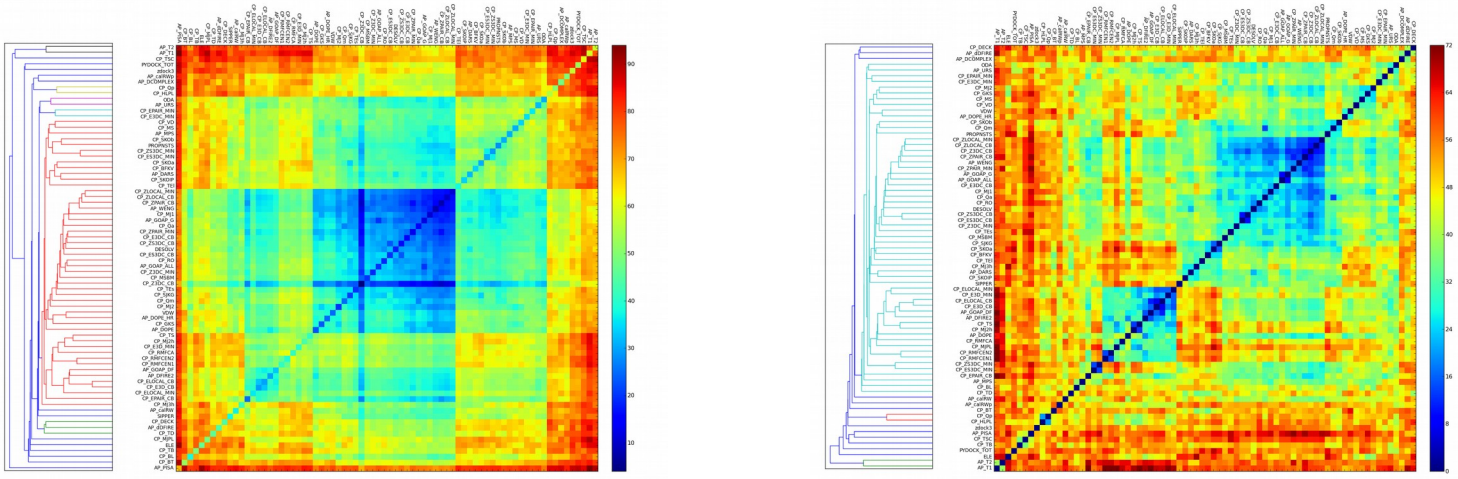

A) FTDock
CP_SKOa and CP_BFKV
CP_SJKG and AP_DCOMPLEX
PYDOCK_TOT and AP_dDFIRE
PYDOCK_TOT and CP_BFKV
PYDOCK_TOT and AP_DCOMPLEX
CP_SJKG and PYDOCK_TOT
CP_SKOa and AP_dDFIRE
CP_SKOa and PYDOCK_TOT
CP_BFKV and AP_dDFIRE
CP_SJKG and AP_dDFIRE
0
20
40
60
B) ZDOCK
CP_BFKV and AP_PISA
AP_T2 and AP_T1
PYDOCK_TOT and ZDOCK
AP_PISA and CP_TB
PYDOCK_TOT and AP_T1
CP_BFKV and AP_T2
PYDOCK_TOT and AP_PISA
CP_BFKV and CP_TB
PYDOCK_TOT and AP_T2
CP_BFKV and PYDOCK_TOT
0
20
40
60
TOP1
TOP10
TOP100
C) SDOCK
AP_MPS and CP_TB
SDOCK and CP_BFKV
PYDOCK_TOT and SDOCK
SDOCK and CP_TB
AP_GOAP_DF_AP_MPS
CP_E3D_MIN and AP_MPS
PYDOCK_TOT and AP_MPS
SDOCK and AP_MPS
CP_E3D_MIN and CP_TB
AP_MPS and CP_RMFCEN1
0
20
40
60
Success rate


TOP1
TOP10
TOP100
Success rate
0
20
40
PROPNSTS
SIPPER
CP_DECK
CP_RMFCA
AP_GOAP_DF
CP_TSC
AP_PISA
CP_HLPL
CP_TB
PYDOCK_TOT
CP_DECK
AP_DCOMPLEX
AP_PISA
CP_Qp
CP_TSC
CP_HLPL
ZDOCK
AP_T2
AP_T1
PYDOCK_TOT
AP_dDFIRE
AP_T1
CP_Qp
CP_TS
AP_PISA
AP_calRWp
AP_T2
CP_TSC
SDOCK
CP_HLPL
FTDock
ZDOCK
SDOCK

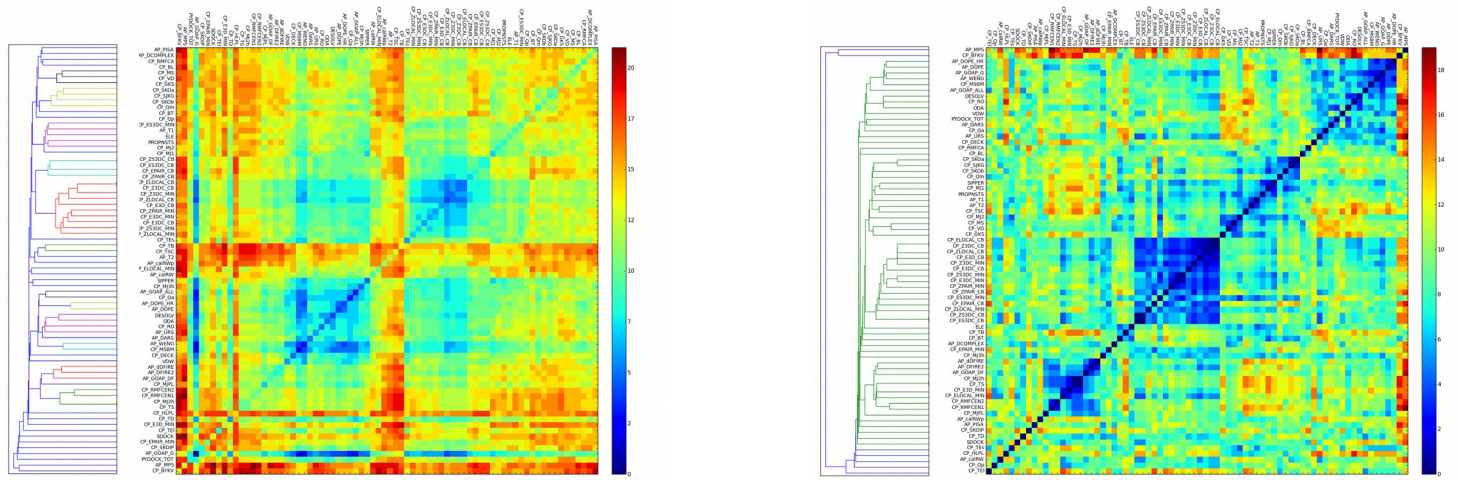
A) FTDock



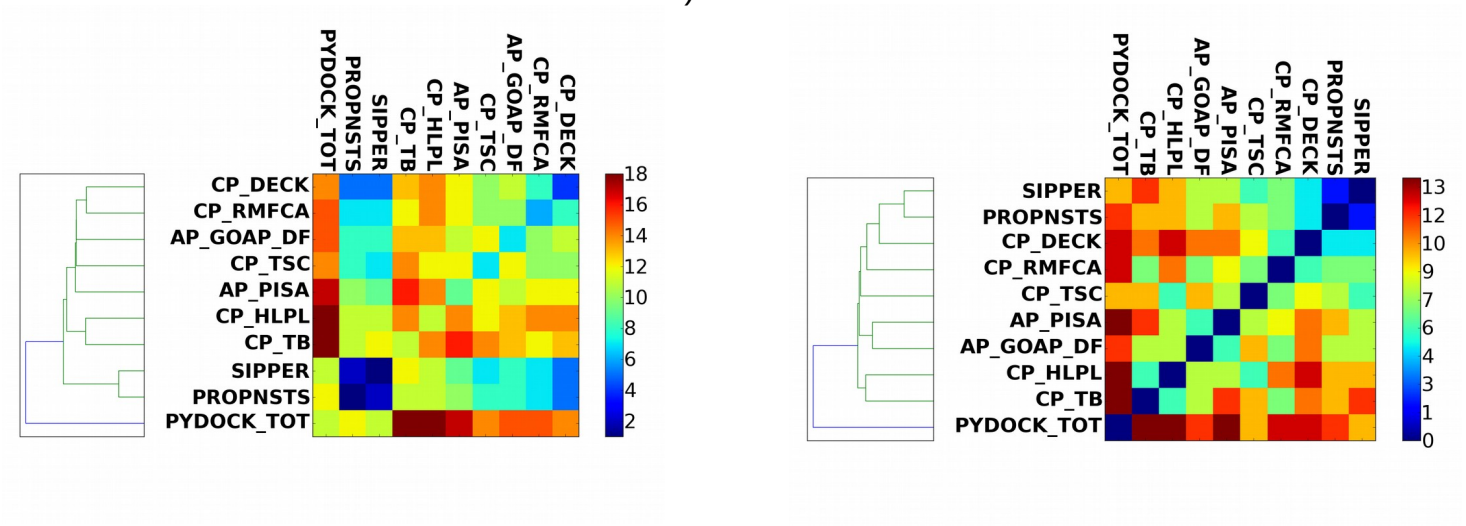
B) ZDOCK



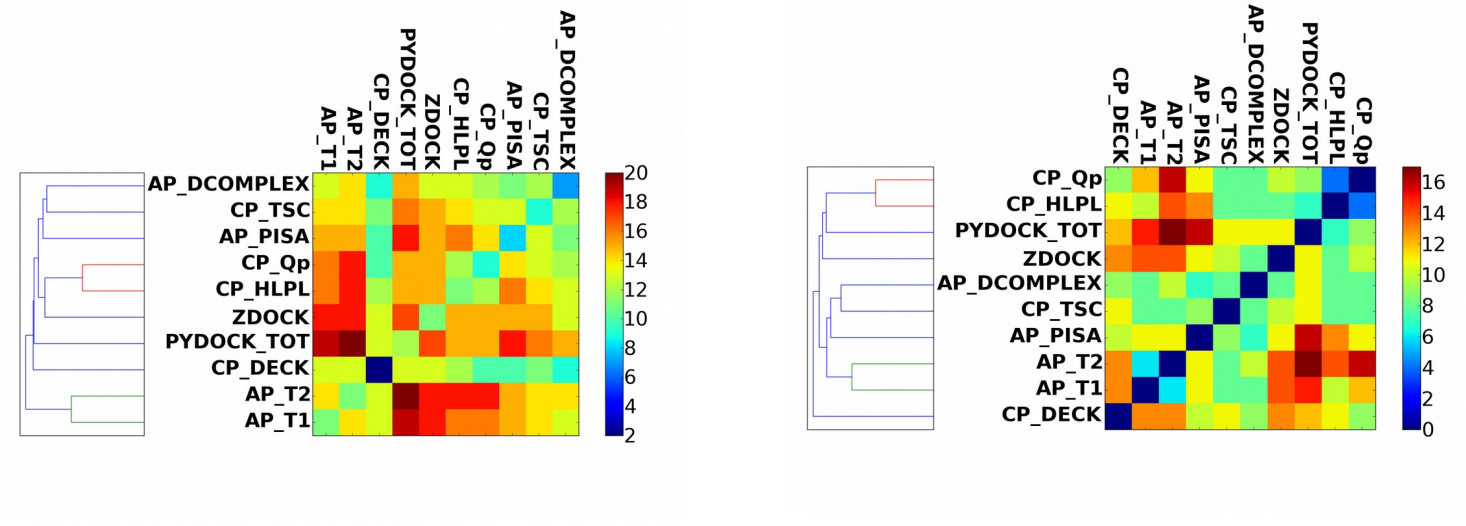
C) SDOCK



A) FTDock



B) ZDOCK



C) SDOCK

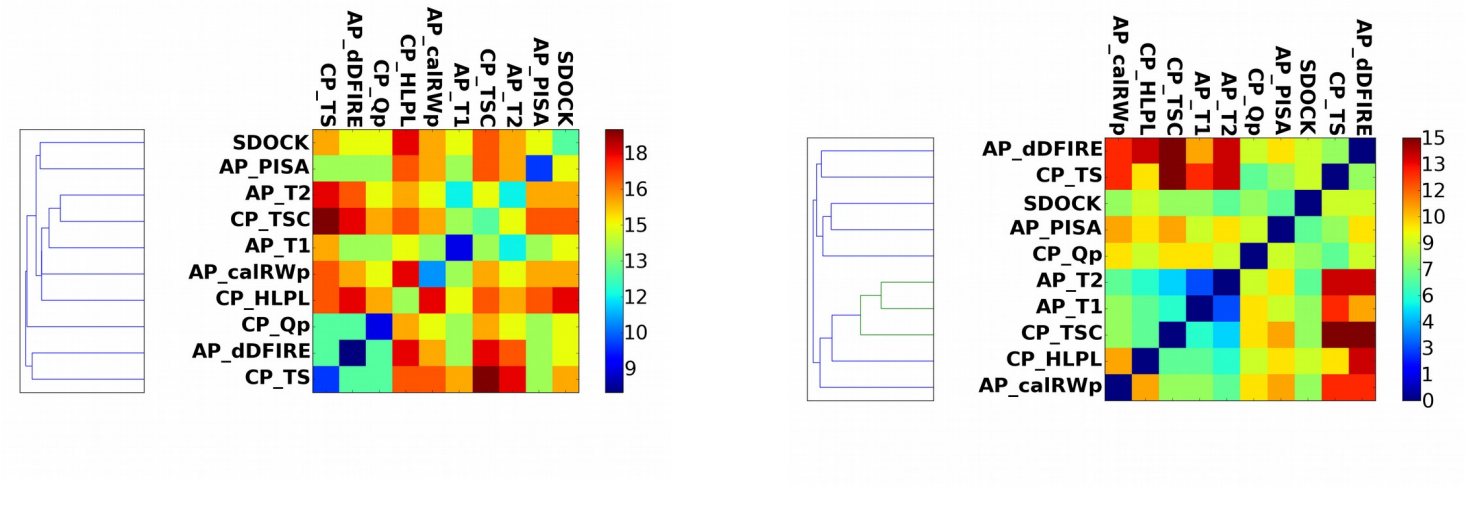


Fig. S4

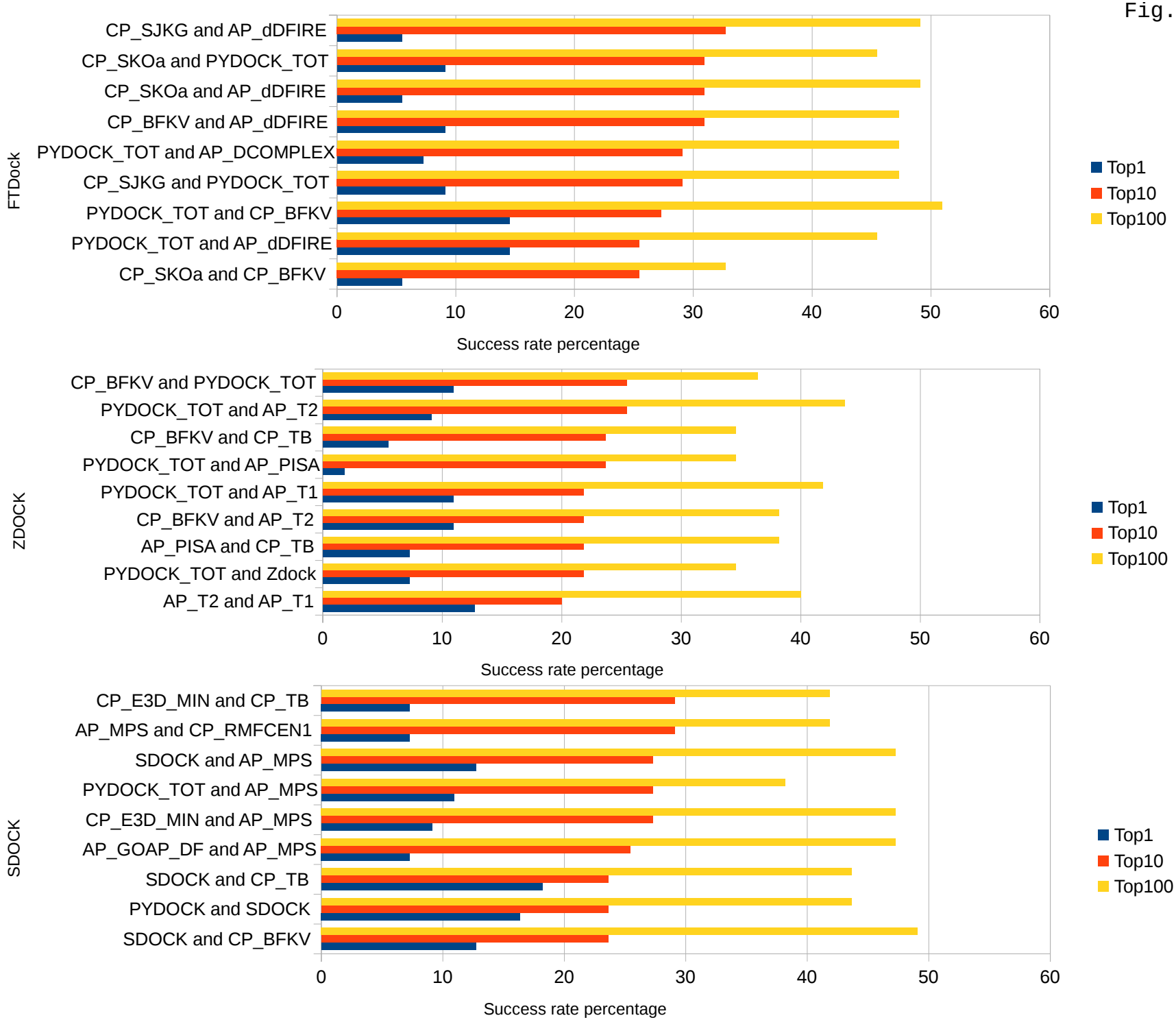

FTDock
CP_SJKG and AP_dDFIRE
CP_SKOa and PYDOCK_TOT
CP_SKOa and AP_dDFIRE
CP_BFKV and AP_dDFIRE
PYDOCK_TOT and AP_DCOMPLEX
CP_SJKG and PYDOCK_TOT
PYDOCK_TOT and CP_BFKV
PYDOCK_TOT and AP_dDFIRE
CP_SKOa and CP_BFKV
0 10 20 30 40 50 60
Success rate percentage
Top1
Top10
Top100
ZDOCK
CP_BFKV and PYDOCK_TOT
PYDOCK_TOT and AP_T2
CP_BFKV and CP_TB
PYDOCK_TOT and AP_PISA
PYDOCK_TOT and AP_T1
CP_BFKV and AP_T2
AP_PISA and CP_TB
PYDOCK_TOT and Zdock
AP_T2 and AP_T1
0 10 20 30 40 50 60
Success rate percentage
Top1
Top10
Top100
SDOCK
CP_E3D_MIN and CP_TB
AP_MPS and CP_RMFCEN1
SDOCK and AP_MPS
PYDOCK_TOT and AP_MPS
CP_E3D_MIN and AP_MPS
AP_GOAP_DF and AP_MPS
SDOCK and CP_TB
PYDOCK and SDOCK
SDOCK and CP_BFKV
0 10 20 30 40 50 60
Success rate percentage
Top1
Top10
Top100

Fig. S5

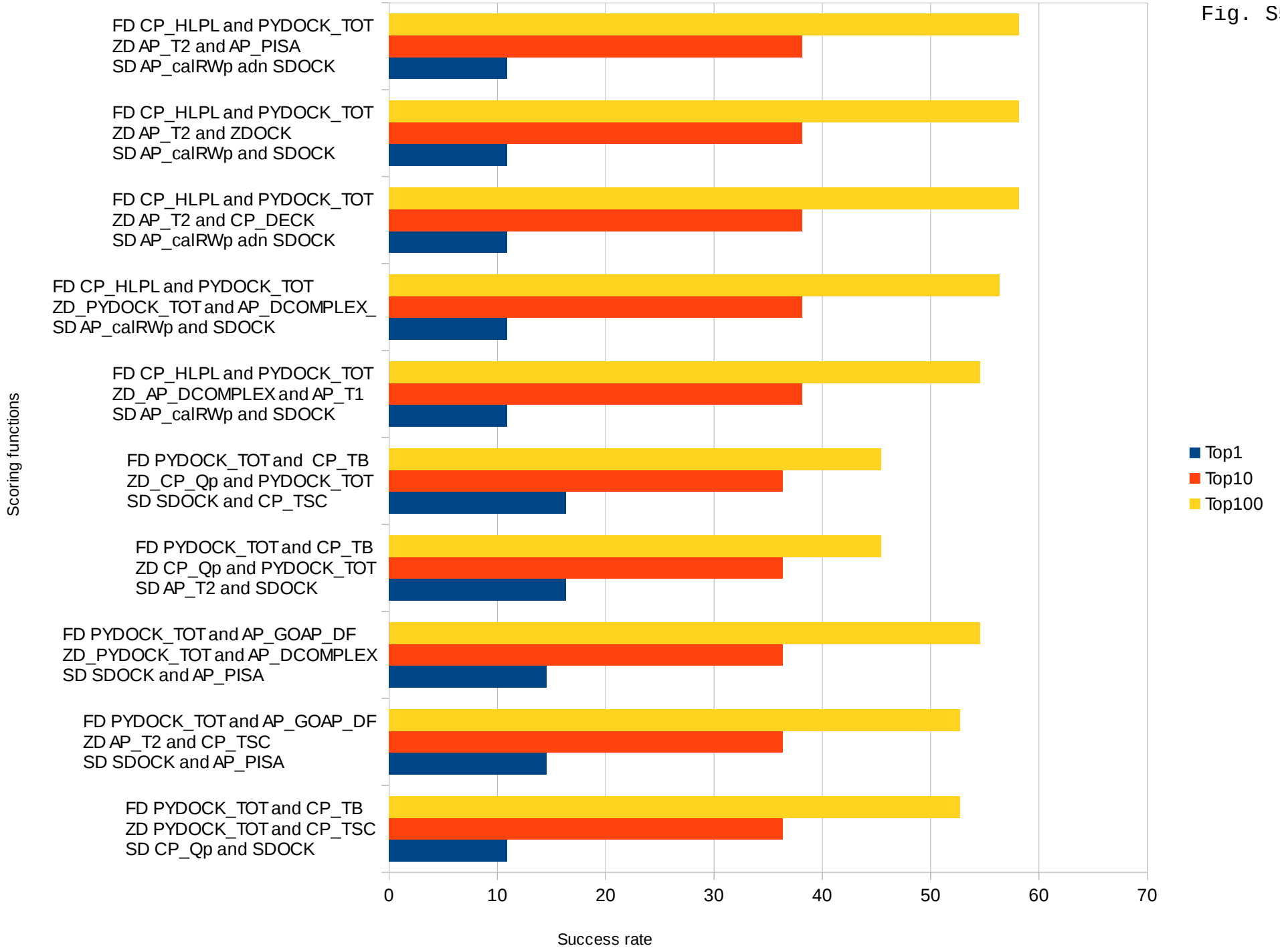

FD CP_HLPL and PYDOCK_TOT
ZD AP_T2 and AP_PISA
SD AP_calRWp adn SDOCK
FD CP_HLPL and PYDOCK_TOT
ZD AP_T2 and ZDOCK
SD AP_calRWp and SDOCK
FD CP_HLPL and PYDOCK_TOT
ZD AP_T2 and CP_DECK
SD AP_calRWp adn SDOCK
FD CP_HLPL and PYDOCK_TOT
ZD_PYDOCK_TOT and AP_DCOMPLEX_
SD AP_calRWp and SDOCK
FD CP_HLPL and PYDOCK_TOT
ZD_AP_DCOMPLEX and AP_T1
SD AP_calRWp and SDOCK
FD PYDOCK_TOT and CP_TB
ZD_CP_Qp and PYDOCK_TOT
SD SDOCK and CP_TSC
FD PYDOCK_TOT and CP_TB
ZD CP_Qp and PYDOCK_TOT
SD AP_T2 and SDOCK
FD PYDOCK_TOT and AP_GOAP_DF
ZD_PYDOCK_TOT and AP_DCOMPLEX
SD SDOCK and AP_PISA
FD PYDOCK_TOT and AP_GOAP_DF
ZD AP_T2 and CP_TSC
SD SDOCK and AP_PISA
FD PYDOCK_TOT and CP_TB
ZD PYDOCK_TOT and CP_TSC
SD CP_Qp and SDOCK
Scoring functions
0
10
20
30
40
50
60
70
Success rate
Top1
Top10
Top100

Table S1. Scoring functions used in this work.

Scoring Function	Description[a]
CP_BFKV	Contact potential calculated between intermolecular residues
CP_BL	Contact potential calculated between intermolecular residues
CP_BT	Contact potential calculated between intermolecular residues
CP_GKS	Contact potential calculated between intermolecular residues
CP_HLPL	Contact potential calculated between intermolecular residues
CP_MJPL	Contact potential calculated between intermolecular residues
CP_MJ3h	Contact potential calculated between intermolecular residues
CP_MJ2h	Contact potential calculated between intermolecular residues
CP_MJ1	Contact potential calculated between intermolecular residues
CP_MJ2	Contact potential calculated between intermolecular residues
CP_MSBM	Contact potential calculated between intermolecular residues
CP_MS	Contact potential calculated between intermolecular residues
CP_Qa	Contact potential calculated between intermolecular residues
CP_Qm	Contact potential calculated between intermolecular residues
CP_Qp	Contact potential calculated between intermolecular residues
CP_RO	Contact potential calculated between intermolecular residues
CP_SKOb	Contact potential calculated between intermolecular residues
CP_SKOa	Contact potential calculated between intermolecular residues
CP_SJKG	Contact potential calculated between intermolecular residues
CP_TD	Contact potential calculated between intermolecular residues
CP_TEl	Contact potential calculated between intermolecular residues
CP_TEs	Contact potential calculated between intermolecular residues
CP_TS	Contact potential calculated between intermolecular residues
CP_VD	Contact potential calculated between intermolecular residues
CP_TSC	The residue level interaction two-step potential
CP_SKOIP	The residue level interaction contact potential
AP_DCOMPLEX	The DComplex potential
AP_dDFIRE	Interaction energy calculated using the dDFIRE potential
AP_DFIRE2	Interaction energy calculated using the DFIRE2 potential
CP_RMFCEN1	The 6bin-HRSC centroid-centroid potential
CP_RMFCEN2	The 7bin-HRSC centroid-centroid potential
CP_RMFCA	The C_alpha-C_alpha potential
CP_TB	The residue level interaction contact potential

CP_TSC	The residue level interaction two-step potential
AP_T1	The first atomic two-step potential
AP_T2	The second atomic two-step potential
AP_DOPE	The DOPE statistical potential
ELE	Total electrostatic energy as calculated using PyDock
DESOLV	Desolvation energy as calculated using PyDock
VDW	Van der Waals energy as calculated using PyDock
PYDOCK_TOT	Total pyDock energy
ODA	The optimal docking area (ODA) score
PROPNSTS	Amino acid propensity score
SIPPER	The SIPPER potential
AP_DARS	The DARS potential
AP_URS	The URS potential
AP_MPS	The MPS potential
AP_WENG	The pair-wise statistical potential implemented in Zdock
CP_DECK	The residue level distance-dependent potential
CP_ZPAIR_CB	The E_pair Z-score C_beta potential
CP_ZLOCAL_CB	The E_local Z-score C_beta potential
CP_ZS3DC_CB	The E_ZS3DC z-score C_beta potential
CP_Z3DC_CB	The E_3DC Z-score C_beta potential
CP_EPAIR_CB	The E_pair C_beta potential
CP_ELOCAL_CB	The E_local C_beta potential
CP_ES3DC_CB	The E_ZS3DC C_beta potential
CP_E3DC_CB	The E_3DC C_beta potential
CP_E3D_CB	The E_3D C_beta potential
CP_ZPAIR_MIN	The E_pair Z-score R_min potential
CP_ZLOCAL_MIN	The E_local Z-score R_min potential
CP_ZS3DC_MIN	The E_ZS3DC z-score R_min potential
CP_Z3DC_MIN	The E_3DC Z-score R_min potential
CP_EPAIR_MIN	The E_pair R_min potential
CP_ELOCAL_MIN	The E_local R_min potential
CP_ES3DC_MIN	The E_ZS3DC R_min potential
CP_E3DC_MIN	The E_3DC R_min potential
CP_E3D_MIN	The E_3D R_min potential
AP_calRW	The calRW distance-dependent atomic potential

AP_calRWp	The calRWplus orientation-dependent atomic potential
AP_GOAP_ALL	The total GOAP energy
AP_GOAP_DF	The DFIRE term in the GOAP energy
AP_GOAP_G	The GOAP_ag term in the GOAP energy
AP_PISA	The PISA score

[a] A more detailed description for each function, including references can be found in the SKEMPI web site (https://life.bsc.es/pid/mutation_database/)

Table S2. Union cardinality for triplets formed by scoring function pairs with each docking method.

FTDock	ZDOCK	SDOCK	UNION
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	30
PYDOCK_TOT and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	29
AP_PISA and CP_TSC	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	29
PYDOCK_TOT and AP_GOAP_DF	PYDOCK_TOT and AP_T1	AP_T2 and CP_TSC	28
PYDOCK_TOT and AP_GOAP_DF	CP_DECK and ZDOCK	AP_T2 and CP_TSC	28
PYDOCK_TOT and AP_GOAP_DF	AP_T2 and PYDOCK_TOT	AP_T2 and CP_TSC	28
PYDOCK_TOT and AP_GOAP_DF	AP_T2 and CP_DECK	AP_T2 and CP_TSC	28
PYDOCK_TOT and AP_GOAP_DF	AP_PISA and PYDOCK_TOT	AP_T2 and CP_TSC	28
PYDOCK_TOT and AP_GOAP_DF	AP_PISA and CP_TSC	AP_T2 and CP_TSC	28
CP_RMFCA and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
CP_HLPL and PYDOCK_TOT	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	28
CP_HLPL and PYDOCK_TOT	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
CP_HLPL and PYDOCK_TOT	AP_PISA and CP_TSC	CP_QP and CP_TS	28
CP_HLPL and PYDOCK_TOT	AP_PISA and CP_TSC	AP_calRWp and CP_TS	28
CP_HLPL and CP_RMFCA	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_QP and CP_TS	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_QP and CP_HLPL	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_QP and AP_PISA	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_QP and AP_dDFIRE	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	CP_HLPL and AP_PISA	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	AP_dDFIRE and CP_HLPL	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	AP_calRWp and CP_TSC	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	AP_calRWp and CP_TS	28
CP_HLPL and CP_DECK	PYDOCK_TOT and ZDOCK	AP_calRWp and AP_dDFIRE	28
CP_HLPL and CP_DECK	AP_T2 and PYDOCK_TOT	CP_QP and AP_calRWp	28
CP_HLPL and CP_DECK	AP_PISA and CP_TSC	CP_QP and CP_TS	28
CP_HLPL and CP_DECK	AP_PISA and CP_TSC	CP_QP and AP_PISA	28
CP_HLPL and CP_DECK	AP_PISA and CP_TSC	CP_QP and AP_calRWp	28
CP_HLPL and CP_DECK	AP_PISA and CP_TSC	CP_HLPL and AP_PISA	28
CP_HLPL and CP_DECK	AP_PISA and CP_TSC	AP_calRWp and CP_TS	28

CP_HLPL and AP_PISA	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	28
CP_HLPL and AP_PISA	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
CP_HLPL and AP_PISA	PYDOCK_TOT and ZDOCK	AP_calRWp and CP_TSC	28
CP_HLPL and AP_PISA	AP_PISA and CP_TSC	CP_QP and AP_PISA	28
CP_HLPL and AP_PISA	AP_PISA and CP_TSC	CP_QP and AP_calRWp	28
CP_HLPL and AP_PISA	AP_PISA and CP_TSC	CP_HLPL and AP_PISA	28
CP_HLPL and AP_PISA	AP_PISA and CP_TSC	AP_T2 and CP_TSC	28
CP_HLPL and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
AP_PISA and PYDOCK_TOT	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	28
AP_PISA and PYDOCK_TOT	PYDOCK_TOT and ZDOCK	AP_calRWp and CP_HLPL	28
AP_PISA and CP_RMFCA	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
AP_PISA and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	28
SIPPER and CP_TB	PYDOCK_TOT and ZDOCK	CP_QP and CP_HLPL	27
SIPPER and CP_TB	PYDOCK_TOT and ZDOCK	CP_HLPL and AP_PISA	27
SIPPER and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	CP_QP and CP_HLPL	27
SIPPER and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	CP_QP and AP_PISA	27
SIPPER and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	CP_QP and AP_calRWp	27
SIPPER and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	CP_HLPL and AP_PISA	27
SIPPER and AP_GOAP_DF	PYDOCK_TOT and ZDOCK	AP_T2 and CP_TSC	27