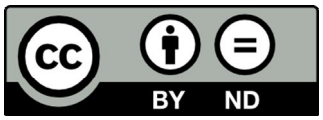
UNIVERSITAT DE BARCELONA

Development and optimization of high-performance computational tools for protein-protein docking

Brian Jiménez García



UNIVERSITAT DE BARCELONA

Facultat de Farmàcia

Programa de Doctorat en Biomedicina

RD 99/2011

Development and optimization of high-performance computational tools for protein-protein docking

Memòria presentada per Brian Jiménez García

per optar al títol de doctor per la Universitat de Barcelona

Director de tesis

Dr. Juan Fernández Recio

Barcelona Supercomputing Center

Doctorand

Brian Jiménez García

UNIVERSITAT DE BARCELONA

七転び八起き

“Fall down 7 times, stand up 8”

Acknowledgements

"The time will come when diligent research over long periods will bring to light things which now lie hidden. A single lifetime, even though entirely devoted to the sky, would not be enough for the investigation of so vast a subject…And so this knowledge will be unfolded only through long successive ages. There will come a time when our descendants will be amazed that we did not know things that are so plain to them…Many discoveries are reserved for ages still to come, when memory of us will have been effaced. Our universe is a sorry little affair unless it has in it something for every age to investigate…Nature does not reveal her mysteries once and for all."

Seneca, *Natural Questions*

This is the end, beautiful friend. A long journey has ended, but a new one has started. I am standing in the shore of this immense ocean, waiting for new challenges, for new histories to be told, for new horizons to be discovered.

There are many people to be acknowledged, I will try to be as exhaustive as possible:

First, to my supervisor Juan. Thank you for always having your door opened, for your guidance, for believing I could become a real scientist. It was an honor to work for and with you.

Second, to my parents, the ones that challenged me so much in the past. I remember you teaching me square roots when I did not even know how to multiply and how magical I felt when you explained me what was the *modus tollens*. Or when you gifted me my first Lego set and you did not help at all with that space rover. Then I learnt how joyful you felt when you succeed after a great effort. Challenges and fail attempts, they come all together. Thank you for teaching me how I could live my life.

To all the people I had the pleasure and the honor to collaborate with: Dr. Mizuguchi and all his lab members and Dr. Bernadó.

Moreover, to all the brilliant people I have met during this long race:

To my group colleagues: Carles, you were my mentor in the field; Laura, thanks to you I discovered what perseverance was; Chiara, our group's hard-worker, always with a smile, kind words and willing to help, thanks for listening to me; to Miguel, the calm and the kindness personified; Iain, always brilliant, I enjoyed sharing table and learning from you every minute; Mireia and Lucía aka "the Antonia's team", you were a life saver for my mental integrity; Manuel, my programming personal guru, I miss so much our talks at lunch time; Pau, I really have no words to

thank you your support, your time and our constructive talks; Didier, my new tablemate, you're the next one; Sergio and Luis Angel, the impossible project team, I loved your energy; Augustina, you shown me that blondes can be smart (and funny); Silvia, my antiparticle at the genomics group; the genomics group in general because of the noise you made me love and the rest of the life sciences department because of all the good moments we shared together.

To all the people that shared with me this journey in a different way: Alba, thanks for all, ramble on; Cris, my pillar and my light at the end of this trip; furry Nana, because of your daily meowing in the morning and random jumping moments; the "sub-humans" team Kaori and Snoo, I miss you a lot; my second family, my band mates in Audiolepsia.

Finally, to all the people that was convinced that engaging in a PhD was a loss of time. I knew you were wrong.

Contents

1. Introduction

Поехали!

Let's go!

Yuri Alekseyevich Gagarin

1.1. A paradigm shift in biology

"*Computing has changed biology forever; most biologists just don't know it yet*". Structural biologist professor Michael Levitt, one of the pioneers of computational chemistry and recipient of the 2013 Nobel Prize in Chemistry together with Martin Karplus and Arieh Warshel for "the development of multiscale models for complex chemical systems", made this observation during a lecture in 1998 at Stanford University (Wooley 2006). Levitt's statement was consistent with a series of new ideas on a change of paradigm in the biological sciences. For instance, in 1991 Walter and Gilbert reported that "The new paradigm, now emerging, is that all the 'genes' will be known (in the sense of being resident in databases available electronically), and that the starting point of a biological investigation will be theoretical. An individual scientist will begin with a theoretical conjecture, only then turning to experiment to follow or test that hypothesis" (Gilbert and Walter 1991). The same year, Lander *et al.* observed that "Biology is in the middle of a major paradigm shift driven by computing." (Lander et al. 1991). Since then, the biological community is aware of how the new developments in computational sciences have changed the way research is performed and interpreted. Actually, biology is just one of the many scientific disciplines affected by this paradigm shift, such as astronomy, physics, chemistry or atmospheric sciences also are (Misa and Thomas 2007).

Computing has experienced a major development in the past decades. Gordon Moore, co-founder of Intel, observed that "every eighteen months, the number of transistors that can be

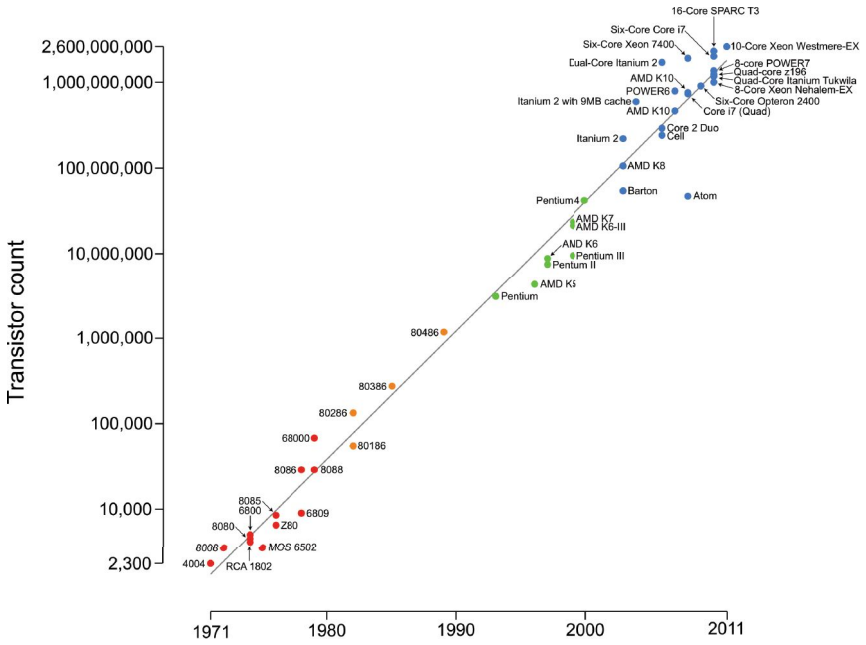


Figure 1. Number of transistors by year and processor model from 1971 to 2011. The tendency follows Moore's law (From Wgsimo, CC BY-SA 3.0)

placed on a chip doubles". This so-called Moore's Law is reflected in an exponential increase in computer performance during the past two decades (**Figure 1**). This has delivered increasingly faster and cheaper computers, which has transformed society since then. Current research in Biology is generating a high amount of data with high information content, and the discipline could be seen as an information science, since "the life sciences use computing and information technology as a language and a medium in which to manage the discrete,

asymmetric, largely irreducible, unique nature of biological systems and observations" (Wooley and Lin 2010). The born of the computational biology within the life sciences scope opens the field to new and exciting lines of research, sensitive to new achievements and developments in computational science. Nevertheless, the change in biology could be deeper than in the rest of scientific disciplines, since computational biology might be as essential for the next quarter century of biology as molecular biology was for the past quarter century.

In addition to the important need to manage high amount of data in biology, the description of cell processes at molecular level is one of the most demanding challenges in life sciences in terms of computational power. For example, molecular simulations at atomic scale, even using coarse grained models, depend on rich computational resources, mainly CPU power and physical memory. The manipulation in computers for many of these problems in structural biology were impractical until the last decades of the past century and currently, many of them would even require years of computation for fine grain simulations. Important milestones have been achieved in the field, such as the design of specific supercomputing architectures using FPGA (Belletti et al. 2006; Kasap et al. 2012) or custom solutions (Narumi et al. 2000; Taiji et al. 2004; Shaw et al. 2009; Shaw 2013), but the design of specific computing architectures may be unworkable in many other problems. In the context of the mentioned paradigm shift in biology, significant advances will mostly stem from the efficient description of cell processes at

molecular level, together with the development of new algorithmic solutions and computational tools optimized for the study of biological problems.

1.2. Proteins: structure and function

1.2.1. A brief introduction to the chemistry of the cell

The chemistry that supports life is extremely sophisticated, more that we had never imagined before. Accounting for about half of the total dry mass of cells (Alberts 1998), proteins play a major role in nature and are often described as the factories of the cell. These macromolecules are involved in the virtually all important functions in living organisms, a few examples of which are oxygen transportation (hemoglobin), sugar level regulation (insulin), signaling (cytokines, cell receptors), immunological system (antibodies) or biological protein synthesis (ribosome complex). In addition, nearly every major process in the cell is carried out by assemblies of biomolecules, which very often can contain a large number of protein molecules. And, as they carry out their biological function, these protein assemblies often interact with other large complexes of proteins (Alberts 1998), clearly showing the inherent complexity of the biochemical reactions within the cell processes.

1.2.2. Protein structure

Proteins are large biomolecules formed by 20 different building blocks called amino acids. Amino acids are small molecules composed of an amine and a carboxylic group and differ in the side chain attached to their alpha carbon (Cα) atom. Each amino acid is capable of polymerizing by forming a peptide bond between the carboxyl group of one amino acid and the amino group of another one, with results in the formation of large polypeptide chains. The 20 standard amino acids that make up proteins are encoded in the DNA biopolymer and are incorporated into a polypeptide chain after the transcription into the RNA biopolymer and the translation process. This was stated by the pioneer Francis Crick as the “central dogma of molecular biology: DNA creates RNA, RNA creates protein”, but the overall process is still under study as many of the mechanisms are not totally elucidated yet. The primary structure of a protein is the linear sequence of the amino acids forming the polypeptide chain. The secondary structure of a protein refers to the local three-dimensional structures with regular geometry formed by the polypeptide chain, which are categorized in two main types with regular geometry: alpha (α-) helixes and beta (β-) sheets. The secondary structure elements are folded into a compact 3D structure, mostly stabilized by hydrogen bonds, known as the tertiary structure of proteins. Different folded polypeptide chains can assemble into multi-subunit complexes that form the quaternary structure. It is worth mentioning at this point the special case of intrinsically disordered proteins (IDP), which lack

a fixed three-dimensional structure. The largely accepted structure-function paradigm, in which the protein function depends directly on the structure, clashes with the existence of these unstructured and extremely flexible proteins and shows the important role of protein dynamics in function.

1.3. Protein-protein interactions

1.3.1. The interactome and the importance of protein-protein interactions

In recent years, biomedical interest has changed its focus from the study of single proteins to the understanding of protein-protein interactions. Ongoing proteomic projects for many model species including humans (Rolland et al. 2014) have confirmed that the majority of proteins mediate their functions by physically interacting with other biomolecules such as other proteins, lipids, nucleic acids or small molecules and thus forming intricate, highly organized and dynamic interaction networks (Rual et al. 2005; Stelzl et al. 2005). A deep understanding of the structure and topology of these protein-protein interaction networks would not only shed light on the cellular processes they regulate, but would also give insight into evolutionary aspects of the proteins involved (Jeong et al. 2001; Fraser 2002; Khuri and Wuchty 2015). In addition, a complete knowledge of protein interactions would be crucial to understand complex pathological processes such as

cancer development and evolution, as well as to find new therapeutic treatments (Jonsson and Bates 2006; Sun et al. 2009; Sun and Zhao 2010; Choura and Rebaï 2012).

Protein interactions are as diverse as the life they sustain (Nooren and Thornton 2003). Regarding kinetics, their lifetime can range from milliseconds (transient) to days (permanent). The interacting surface between partners can present different shapes and topologies, with a total surface area of up to several thousand $Å^2$. The interaction can involve a varied range of movements, from a negligible conformational change (rigid-body) to a partial refolding of the partners (induced fit). Regarding thermodynamics, protein complexes can show binding affinities

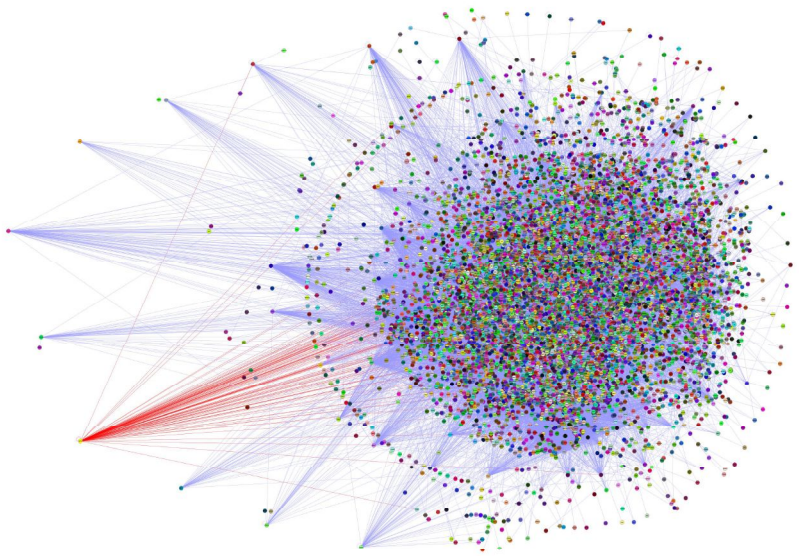


Figure 2. Human interactome constructed from publically available data and visualized in Cytoscape 2.6. © Andrew Garrow

ranging from 10^{-4} to 10^{-14} M.

We know that the complexity of a living organism is not related to its total number of genes. For example, the number of genes in human is approximately half the number of genes in rice (Goff et al. 2002). The complexity of a living organism seems more related to other dynamic and functional aspects of the genes, such as their expression and regulation or the interactions between the different biomolecules. This is reflected, for instance, in the size of the set of proteins expressed in a cell at a given time, the expressed proteome, or in the size of the entire network of protein-protein interactions in an organism, the interactome (Stumpf et al. 2008). Indeed, the total number of estimated protein-protein interactions in the human interactome ranges from 130,000 (Venkatesan et al. 2009) to around 650,000 (Stumpf et al. 2008), depending on the method used to estimate it. In any case, this number is larger than in other organisms, e.g. one order of magnitude larger than in *Drosophila* or three times larger than in *Caenorhabditis Elegans* (Stumpf et al. 2008).

The most popular techniques that have been extensively used for the identification of protein-protein interactions are yeast two-hybrid (Fields and Song 1989) and tandem affinity purification coupled to mass spectrometry experimental methods (Rigaut et al. 1999; Puig et al. 2001). Both techniques have been applied in a large-scale and high-throughput context, helping to identify many new interactions in different organisms including human. The analysis of these new interactions has many biological

applications. For instance, the functionality of unidentified proteins can be predicted on the basis of their interaction with another protein whose function is already known (Zhang 2008). On the other side, a deeper understanding on existing interactions could foster new strategies and methods in protein engineering and drug discovery projects (Rual et al. 2005). Finally, new interactions can be directly mapped to disease-associated proteins (Li and Li 2004; Oti et al. 2006), with clear applications in the biomedical field.

1.4. Protein-protein complex structural modeling

1.4.1. Experimental determination of protein-protein complex structure

As of February 2016, the total number of biomolecules deposited in the Protein Data Bank (PDB) (Bernstein et al. 1978) is 116,085, where 107,808 account for proteins (92.9%), 2,878 for nucleic acids (2.5%), 5,373 for protein-nucleic acid complexes (4.6%) and 26 for other. Concerning protein complexes, X-ray crystallography is the most represented experimental technique (96,963, 83.53%) followed by nuclear magnetic resonance (NMR) spectroscopy (9,896, 8.52%), electron microscopy (695, 0.6%) and HYBRID (84, 0.07%). X-ray crystallography, which has no limits on the sample size, can be used to describe the

atomic 3D structure of a crystallized macromolecule based on the diffraction pattern produced by an X-ray beam after contacting the electrons of the sample. Despite being a very popular and mature experimental technique, it presents problems concerning its applicability to some protein complexes. More in detail, systems such as membrane proteins, flexible complexes (frequently with flexible loops that cannot be solved), transient or low-affinity complexes, or intrinsically disordered proteins, are very challenging or directly not suitable for this technique, due to the problems in the crystallization. In addition, a long debate exist about whether crystallization conditions may or not represent *in vivo* environments or conformations that are biologically relevant (Ofran and Rost 2003; Bahadur et al. 2004; Bahadur and Zacharias 2008).

NMR spectroscopy can help to overcome the problems found in X-ray crystallography and, indeed, it represents the second experimental method in popularity in the PDB, as above mentioned. This method can be used either in solution or in solid state. It is based on the physical phenomenon in which nuclei in a magnetic field absorb and re-emit electromagnetic radiation at a specific resonance frequency. This frequency depends on the strength of the magnetic field and the magnetic properties of the isotope of the atoms, giving access to details of the electronic structure of the studied molecule. In recent years, NMR spectroscopy has incorporated many new technical advances (Kanelis et al. 2001; Castellani et al. 2002) to deal with proteins larger than 25 kDa, but this technique can suffer from certain

limitations in larger systems. On the other hand, the ability of sampling proteins in solution allows the researcher to perform the experiment in environment conditions that are more similar to the cell context. In this way, the method can efficiently describe the dynamics of the protein complex and define the 3D structure of mobile loops or sections of membrane proteins or amyloid systems (Castellani et al. 2002), but at the expense of a lower resolution as compared to X-ray crystallography.

Other promising techniques are small-angle scattering, either using a X-ray (SAXS) or a neutron beam (SANS), and cryo-electron microscopy (cryo-EM) (Bernadó 2011). These techniques can be used as complementary to other methods, i.e. NMR and SAXS (Sibille and Bernadó 2012) or computational modelling tools (Petoukhov and Svergun 2005; Schneidman-Duhovny et al. 2013), and provide promising results in the characterization of the general size and shape of large macromolecular complexes. Moreover, cryo-EM technique has experienced in the past years important advances in electron detection and image processing. The resolution by cryo-EM is now beginning to rival that of X-ray crystallography (Bai et al. 2015; Doerr and Allison 2015), with very encouraging results at near-atomic resolution of macromolecules including ribosomes from human pathogens or mitochondria, ion channels or a key enzyme in the biogenesis of methane (Kühlbrandt 2014).

Despite the number of deposited protein structures is relatively high (107,808 as of February 2016), the total number of structures corresponding to protein-protein complexes is only of 17,184 (Protein Data Bank in Europe, http://www.ebi.ac.uk/pdbe/entry/search/index?assembly_composition:%22protein/protein%20complex%22), a low number considering the number or estimated protein-protein interactions

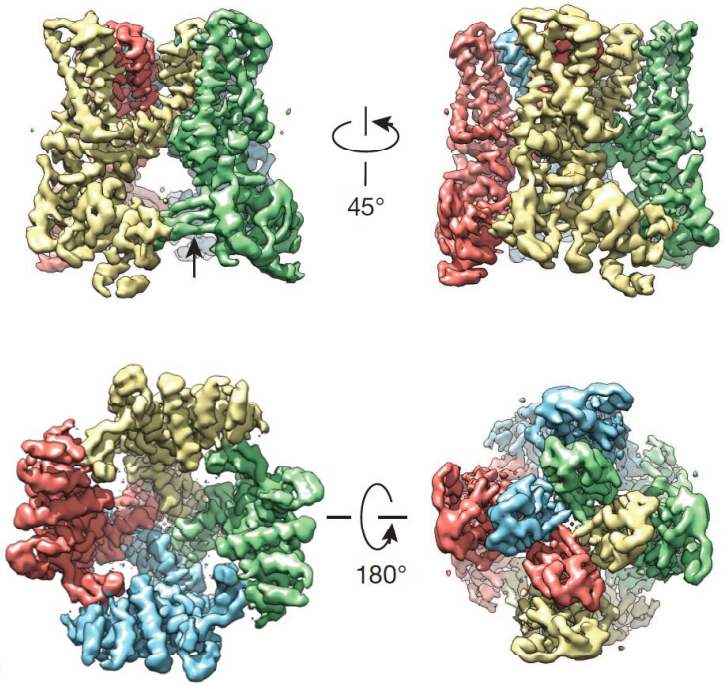


Figure 3. TRPV1 channel, a structure impossible to solve using X-ray crystallization, at 3.4-Å of resolution using cryo-EM. (From Liao et al. 2013)

in human is expected to be one order of magnitude larger than the number of proteins (Stumpf et al. 2008). Moreover, as of April

2016, there are only 4,586 protein-protein interactions in human with available 3D structure (Interactome3D; Mosca et al. 2013), a very small number in comparison to the estimated number of total protein-protein interactions in human, ranging from 130,000 (Venkatesan et al. 2009) to around 650,000 (Stumpf et al. 2008). This low structural coverage of protein-protein interactions could be explained by the actual limitations in experimental methods. In this context, computational methods for protein-protein complex structure prediction may help to overcome this problem.

1.4.2. Structural modeling of protein-protein complexes

From a computational point of view, there are two main approaches to model the structure of a protein-protein complex: *ab initio* docking or template-based modeling.

A*b initio* docking aims to predict the binding mode of two proteins, that are known to interact, starting from the 3D coordinates of the two interacting partners (Ritchie 2008). This will be analyzed in deep in the next section. Template-based modeling aims to model a protein-protein complex based on the structure of a homologous complex. The popularity of template-based methods has increased in the past years thanks to the development and support of many protein-protein interactions databases that can provide the required templates. However, the quality of template-based predictions clearly depends on the

availability of suitable templates (Vakser 2013; Aloy and Russell 2002; Kundrotas et al. 2012). According to Interactome3D database, only 4,294 complexes (as of April 2016) could be directly modelled based on homologous templates (using sequence alignments or domain-based templates), with around 50% accuracy. By using structural alignments to identify the templates, the number of complexes that could be modelled might increase, but at the expense of lower accuracy in the predictions.

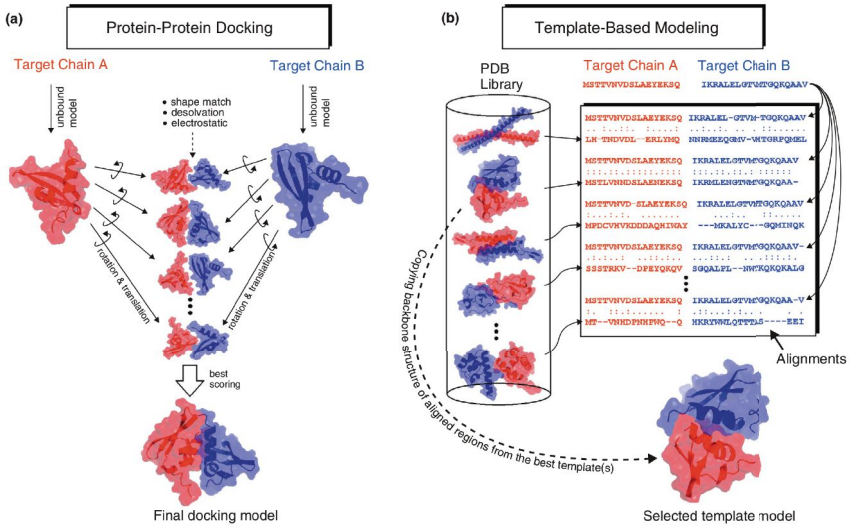


Figure 4. Two main protein-protein complex prediction methods. (a) Protein-protein docking, (b) Template-based modeling. (Szilagyi and Zhang 2014)

1.4.3. Protein-protein docking: a computational method for protein-protein complex structure prediction

Ab initio protein-protein docking can be used as a general method to model the structure of a protein-protein complex, not needed the existence of suitable templates. The objective of docking is to predict a complex structure from the separately determined protein structures (unbound docking). For testing and development purposes, the co-crystallized partners in a known protein complex structure can be separated and re-docked (bound docking), but obviously this has no practical value for biology.

Originally in the decade of the 70's of the last century, docking was understood as a technique for the refinement of the binding site of the two protein partners (Levinthal et al. 1975; Pincus et al. 1976). At that time, both computational power and the force-field models were very limited. Later in the 90's, the democratization of computing and a more accurate knowledge of the nature of the problem, i.e. more structures were resolved and deposited in the Protein Data Bank, led to remarkable advances in the field that are still valid (Katchalski-Katzir et al. 1992; Fischer et al. 1995). Typically, docking methods address the problem in a two-step process. In the first step, a pool of possible protein-protein docking poses is generated, usually considering the proteins as rigid-body or with very limited conformational

changes. Then, these generated docking models are scored in order to identify the correct models. Very often, this second step includes a flexible refinement, usually on the side-chains. Other docking methods use flexibility from the first sampling step, performing conformational sampling and scoring at the same time that the complex is refined in terms of flexibility.

Sampling

The most common strategy in docking is to neglect the conformational flexibility upon binding. In this so-called rigid-body docking approach, the sampling process explores the six-degrees of freedom of the translational and rotational space of the two-rigid body systems, and generates a pool of possible poses. One of the most popular approaches for rigid-body sampling is based on the use of a discrete grid search (protein atoms are mapped onto the different grid cells, given a fixed grid resolution). In this way, the initial search by a correlation function in the space of N^6, where N is the length of the side of the cubic grid, can be accelerated using Fast-Fourier Transform (FFT) libraries (Katchalski-Katzir et al. 1992). The first docking methods using this approach were MOLFIT (Katchalski-Katzir et al. 1992), GRAMM (Vakser 1997) and FTDock (Gabb et al. 1997), which incorporated an extra grid for taking into account electrostatics contributions, but other methods have included desolvation based on atom-contacts as in ZDOCK (Chen et al. 2002) or pairwise interaction potentials as in PIPER (Kozakov et al. 2006).

FFT-based methods can be performed in polar coordinates instead of in Cartesian ones as in Hex (Ritchie and Kemp 2000) or FRODOCK (Garzon et al. 2009). FFT-based methods can benefit from graphics processing units (GPU) computing accelerating the search several orders of magnitude (Sukhwani et al. 2009; Ritchie and Venkatraman 2010). In addition to FFT-based methods, geometric hashing is also utilized in the sampling step. The PatchDock program creates a Connolly-style representation of the protein restricted to concave, convex and flat shapes. Surface representations are scanned to find zones of high complex-complementarity (Duhovny et al. 2002).

A different approach in rigid-body docking comes from non-exhaustive search methods using either Monte Carlo or other energy-minimization techniques. This alternative to the FFT grid-based approaches use an explicit representation of the interacting proteins, at atomic or coarser-grained level, in search of the global energy minimum in order to identify the native orientation. However, the computational cost of conformational search in atomistic representation is high, so in practice, these methods are often used to perform a first search in which the molecules are rigid. Very often, the initial rigid-body docking search is followed by an additional flexible refinement step, within the same atomistic framework. The ICM-DISCO docking method pioneered the application of global energy Monte-Carlo optimization and side-chain refinement (Fernandez-Recio 2003). In RosettaDock refinement step, a side-chain minimization using a rotamer library is performed (Schueler-Furman et al. 2005), but

new versions of the software include more refinements. In HADDOCK approach, several flexible refinement steps are performed using molecular dynamics, with increasing levels of flexibility. In order to lower the computational costs, the number of degrees of freedom of the conformational search is dramatically reduced by using distance restraints from experimental data (Dominguez et al. 2003).

Explicit flexibility is not possible to be included in FFT-based methods. Therefore, in some cases, implicit flexibility is considered by using soft-potentials, or letting the proteins to intersect in a more laxative way at the surface, but strongly penalizing interactions with core residues.

Scoring and flexible refinement

After the sampling phase, several docking models (up to hundreds of thousands) are usually generated. These docking models are ranked according to the criteria used during the sampling, but accurate scoring functions cannot be efficiently used in FFT-based approaches (each scoring term and/or atom type would need a different grid, which would make the process impractical). Therefore, a more accurate scoring function in order to identify the near-native docking models is usually required after this first rigid-body step. The scoring process has to be sufficiently robust to include in the top ranked models one or more near-native solutions that could be sufficiently close to the

real complex structure in terms of ligand and interface RMSD. Sampling and scoring functions are typically highly coupled because scoring functions are generally optimized to deal with particularities of the sampling phase such as implicit flexibility, etc.

Some of the most successful scoring methods are based on biophysical energetic terms, such as pyDock (Cheng et al. 2007), which takes into account desolvation, electrostatics and van der Waals energy terms. The ZRANK (Pierce and Weng 2007) scoring function implemented in the ZDOCK method is composed of desolvation, based in pairwise atomic contact energies, short and long range attractive and repulsive electrostatics and attractive and repulsive van der Waals terms.

Docking scoring functions continue to be the object of active research (Moal et al. 2013a). Recent developments include coarse-grain models (Pons et al. 2011; Ravikumar et al. 2012), potentials derived from decoy structures (Liu and Vakser 2011), an asymmetric potential designed specifically for antibody–antigen docking (Brenke et al. 2012), or scores based on machine learning (Azé et al. 2011). Other approaches have focused on the inclusion of bioinformatics and experimental information (Schneidman-Duhovny et al. 2012a), or evolutionary information beyond sequence conservation (Andreani et al. 2013). Many of them have been compiled and benchmarked in a recent study to shed some light into the large amount of available information on scoring functions (Moal et al. 2013a, 2013b).

In addition to scoring, a flexible refinement process is usually performed after the first rigid-body sampling, thus aiming to bring back some of the structural information possibly lost during the initial sampling phase due to limitations on the resolution of the method (coarse-grained sampling) or to the paradigm used (rigid-body). This refinement usually includes explicit treatment of the backbone and/or the side-chains flexibility. One example is ICM-DISCO, which performs Monte-Carlo refinement of interface side-chains using internal coordinate representation, in combination with flexible minimization (Fernández-Recio et al. 2003). Another example is HADDOCK, which uses soft potentials to deal with clashes during the sampling phase and includes water molecules in the refinement phase. In RosettaDock, a side-chain minimization using a rotamer library is performed (Chaudhury et al. 2007). FireDock (Andrusier et al. 2007) uses a scoring function based on electrostatics, van der Waals, hydrogen and disulfide bonds, solvation and the change in internal energy in order to optimize the conformation of the side-chains based on a rotamer library. FiberDock (Mashiach et al. 2010) extends the protocol of FireDock, but performing backbone refinement using normal mode analysis.

Flexible docking

In flexible docking, the sampling process takes into account the dynamics of the protein, either globally, or at least at the protein-protein interface. Usually, the flexible conformational search is

driven by energy minimization. In ATTRACT docking method (Zacharias 2003; May and Zacharias 2008), a reduced protein model (each residue is represented by four beans) and the first non-trivial normal modes of the anisotropic network model (ANM) are used starting from a large number of precalculated models. SwarmDock (Li et al. 2010; Moal and Bates 2010) method is based in the swarm intelligence particle swarm optimization (PSO) algorithm, which makes use of normal modes extents to simultaneously optimize docking poses with an electrostatics and van der Waals scoring function. Another example of flexible docking is FlexDock, which includes domain-domain flexibility by previous identification of hinges (Sandak et al. 1998).

Use of structural and biological available information

Protein-protein docking methods can benefit from the use of available structural or biological data on a particular system of interest. This information can be integrated during the sampling process, as in HADDOCK, can be used to select regions of interest and to filter false-positive models, as in ClusPro (Comeau et al. 2004), or can be implemented as post-filtering restraints, as in pyDockRST (Chelliah et al. 2006). Particularly interesting is the use of available experimental SAXS data for protein-protein docking, such as in pyDockSAXS (Pons et al. 2010), FoXS (Schneidman-Duhovny et al. 2012b) and ClusPro (Xia et al. 2015), or the use of cryo-EM data (de Vries et al. 2016).

1.4.4. Protein-protein docking benchmarks and datasets

In order to properly evaluate the performance of the different protein-protein docking methods in the same conditions, a common reference benchmark is ideally needed. While several collections of protein-protein complex structures have been reported by different groups, the most popular benchmark for protein-protein docking is the one developed by Weng's group (Chen et al. 2003). The first version of the benchmark included 57 test cases where the structure of the complex and that of the unbound partners were known. The benchmark has been periodically updated (Mintseris et al. 2005; Hwang et al. 2008, 2010), regarding the number and the variability of the different cases (176 in the version 4.0). The version 5.0 of the protein-protein docking benchmark includes the second version of the affinity benchmark (Kastritis et al. 2011), which contains the dissociation constant for a total number of 179 entries out of 230 non-redundant, high-quality structures of protein–protein complexes (Vreven et al. 2015).

DOCKGROUND (Douguet et al. 2006, Gao et al. 2007) is another important resource for the development and optimization of protein docking methods. This database is formed by protein-protein complex structures from the PDB, and is regularly updated. In its first release included a comprehensive collection of co-crystallized (bound-bound) protein-protein complexes, but

at this time it offers information about unbound models and decoy sets too.

Other valuable sets of benchmark data come from false positive decoys generated by ZDOCK and ClusPro (Liu et al. 2008); (Douguet et al. 2006; Kirys et al. 2015), flexible docking decoy sets (Launay and Simonson 2011), low-homology models (Kundrotas et al. 2011); (Anishchanka et al. 2014) or SKEMPI, a database containing data on the changes in thermodynamic parameters and/or kinetic rate constants upon more than 3,000 mutations for protein-protein interactions of which at least one co-crystallized complex structure has been solved and is available in the PDB (Moal and Fernández-Recio 2012).

1.4.5. Validation of protein-protein complex structure prediction methods: the CAPRI community experiment

Inspired by the CASP community experiment in protein structure prediction (Lattman 1995), the Critical Assessment of PRedicted Interactions (CAPRI) established in 2001, was designed to test the performance of docking algorithms (Janin 2002; Janin et al. 2003). CAPRI has played an important role in advancing the field of protein complex modeling, just as CASP fostered the development of methods for protein structure prediction. The initial goals of the CAPRI experiment were focused on protein-protein docking and scoring procedures, but in the most recent

editions new challenges were added, such as modeling protein-peptide and protein-nucleic acids interactions (Pallara et al. 2013), estimating the binding affinity of protein-protein complexes (Lensink and Wodak 2013; Moretti et al. 2013), or predicting the position of water molecules at protein-protein interfaces (Lensink et al. 2014).

The CAPRI experiment allows the comparison of different docking methods on a set of previously chosen targets by the organizers, which consists on experimentally determined complex structures that are not yet publicly available. There are usually two participation modalities for each target: predictors and scorers. For each CAPRI round, which can have multiple targets, predictor groups are requested to submit a total of ten complex models starting from the separately crystallized structures of the complex components, or from homologous templates supplied by the CAPRI organizers. In a second step, the scorer groups are invited to evaluate a common pool of docking models gathered from the contributions of the uploader groups, and to submit their ten best ranked selected models from the uploaders pool. At the end of each round, the ten models submitted by each of the predictor and scorer groups are evaluated by the organizers, based on the fraction of native contacts, ligand and interface RMSD with respect to the real (and confidential) complex structure (Lensink et al. 2007) and (Lensink and Wodak 2010). The evaluation criteria have been revisited and slightly adapted to the assessment of protein-peptide interactions (Lensink and Wodak 2013).

Since 2001, five different CAPRI editions have been completed, corresponding to 35 prediction rounds, with a total of more than 100 targets. During this time, the community has gathered in five meetings with the sixth one planned to be celebrated in April 2016. The analyses of the docking and prediction results obtained in all the previous CAPRI editions (Méndez et al. 2003; Méndez et al. 2005; Lensink et al. 2007; Lensink and Wodak 2010, 2013) offer a useful resource to track the evolution of the protein docking field and an important tool for anticipating the future challenges in modeling of protein-protein interactions (Lensink and Wodak 2014).

Moreover, the CASP and CAPRI communities established closing ties in the summer of 2014 with the first joint CASP-CAPRI round with the final goal of better integrating the different computational approaches for modeling macromolecular assemblies and their building blocks (Lensink et al. 2016).

1.5. Current limitations of computational methods for protein-protein complex prediction

In spite of the advances, structural modeling of protein-protein complexes by docking is still a very challenging problem. The overall performance of the different existing docking methods in the last CAPRI evaluated round (5th edition) corroborates this

fact (Lensink and Wodak 2013). In total, 64 groups including 12 web-servers participated in at least one of the ten targets. Of these groups, 38 (including eight servers) submitted a model ranked acceptable or higher for at least one target (Lensink and Wodak 2013). Many of the successful groups/servers were different from those submitting correct predictions in the previous assessment, which reflects the rapid evolution of the docking community (Lensink and Wodak 2013). Our group performed within the top five among a total of 63 participants in both predictor and scorer categories. Top performing docking methods were HADDOCK, SwarmDock, GRAMM, ClusPro and pyDock in descending order, which submitted high-accuracy models only for 20% of the cases, but medium-accuracy models for 60% of the cases and acceptable models for 90% of the cases. This clearly shows that there are still many limitations in current docking methods that should be addressed.

1.5.1. Conceptual challenges in ab initio docking

Flexibility

The dynamic nature of proteins is undeniable, and can be described at different structural levels (atomic vibration, local movements, domain motions, global rearrangements, etc.), covering an extensive spectrum of amplitudes and energies as well as a huge time-scale range. In this fashion, from the fastest to the slowest motions one can find covalent bond vibrations occurring in the scale of femtoseconds, side chain rotations or loop flips usually on the pico to nanosecond timescale and large

domain motions, macromolecular associations or protein folding that range from milliseconds to seconds and even minutes or hours.

Capturing this inherent flexibility in computational protein-protein docking models is still an ongoing and very active field of research in the discipline. As described in section **1.4.3**, the different docking methods incorporate flexibility at different points of the simulation, during the sampling or in the refinement steps. While some methods use soft potentials to model the classic lock-and-key scenario (Fig 1a), other use flexible refinement or normal modes analysis to model induced fit (Fig 1b), and others precomputed ensembles to model conformational selection mechanism (Fig 1c, 1d).

Multi-protein complexes

Many important biological processes in cells are mediated by assemblies of ten or more proteins (Alberts 1998) which makes multi-protein complexes prediction of vital importance if we want to fully understand protein-protein interactions. Multimeric complex prediction is a hard problem due to its combinatorial complexity nature. Some advances have been made in the recent years: Multi-LZerD (Esquivel-Rodríguez et al. 2012) takes into account the pairwise interactions already calculated in order to estimate the final multimeric complex, CombDock performs the search in a similar fashion too (Inbar et al. 2005). Some assumptions can simplify the combinatorial problem when

complex symmetry or stoichiometry is known in advance: triangular trimers (Popov et al. 2014) or other types of symmetry (Schneidman-Duhovny et al. 2005; Pierce et al. 2005). Some other approaches use available experimental data to restraint the combinatorial search, as in DockStar (Amir et al. 2015). The limitations on docking methods for multimeric complex prediction are still a major drawback, and actually, dealing with biological real cases still provides very poor predictive results (Lensink et al. 2016).

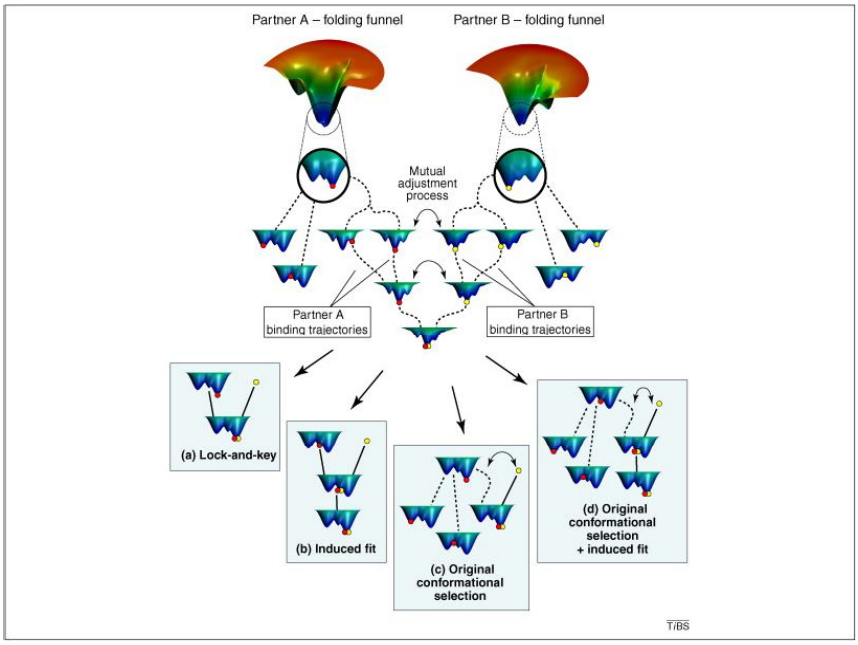


Figure 5. Description of the binding process: (a) the classical lock-and-key model (b) the classical induced-fit (c) the classical conformational-selection model (d) the conformational-selection-plus-induce-fit model. From (Csermely et al., 2010).

Docking of models

For the majority of complexes for which protein-protein docking can be applied (Interactome3D, April 2016) the structure of one or both interacting partners is not available and therefore need to be modelled. Dealing with modelled structures for one or both partners in protein-protein docking adds an extra difficulty to the problem (Kundrotas et al. 2011; Anishchanka et al. 2014). When suitable templates can be found for the modeled protein, i.e. close in sequence identity and high coverage, docking results are similar to those when using x-ray structures. However, when only remote homologues can be found for the interacting proteins, results tend to be very poor as the CAPRI experiment has shown (Lensink et al. 2016).

Low-affinity binding

Transient interactions, which involve protein interactions that are formed and broken easily, are important in many aspects of cellular function (Perkins et al. 2010), especially in the regulation of biochemical pathways and signaling cascades in the cells (Acuner Ozbabacan et al. 2011). Low-affinity binders suppose a huge challenge for docking scoring functions. In these scenarios of encounter complexes or transient complexes, the ability to capture the correct binding pose decreases dramatically as has been reported in the literature (Kastritis and Bonvin 2010) as well as in past CAPRI editions (Lensink and Wodak 2013). External experimental information such as SAXS, as well as new scoring

functions that take into account the nature of these interactions (Joachimiak et al. 2006; Tobi 2010) can be extremely useful in order to overcome this challenge.

1.5.2. Technical challenges in computational docking

In addition to the above described conceptual challenges in docking, there are technical limitations that need to be addressed before efficient application of docking methods to interactomic scale.

High-performance computing architectures

Computers are general purpose programmable machines based on integrated circuits that can perform arithmetic and logical operations automatically. A computer consists of at least one processing element, typically a central processing unit (CPU), and some sort of memory bank to store and retrieve information. The processing element performs arithmetic and logic operations and the sequence of operations is defined by the nature of the instruction and the information stored in memory. Many techniques can be applied to increase the performance of a computer, for example using multiple layers of cache memory, different CPU technology (vector and superscalar processors, graphical process units, etc.) or even programmable hardware

(FPGA, etc.). A supercomputer is a computer with a high-level computational capacity compared to a general-purpose computers and their performance is measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS). Commonly, they are highly-sophisticated systems with large arrays of computing nodes interconnected by fast multi-level network lines and with access to massive storage systems. Not only the hardware is sophisticated, but software and operating systems have to be adapted to the supercomputing environment. Their architecture has evolved enormously since the apparition of the Cray-1 in 1976, one of the first and most successfully commercial supercomputers, with 160 MFLOPS. Nowadays, it is common to have hybrid GPU-CPU solutions coexisting in the same system, as the currently most powerful supercomputer in the world, the Tianhe-2 (MilkyWay-2) located at the National Super Computer Center in the Chinese city of Guangzhou, which offers a performance of up to 33,862.7 TFLOPS with a total number of cores of 3,120,000 (http://www.top500.org/lists/2015/11/).

Supercomputers are extremely useful tools in research and have fostered many recent advances in many disciplines, but they suffer from important drawbacks. First, the energetic consumption of the power and cooling system is simply exorbitant: Tianhe-2 has a power consumption of 17,808 KWh. Initiatives in this context have been proposed to help overcoming this problem such as the MontBlanc project (Rajovic et al. 2013) and the creation of a new list of the top 500 green

supercomputers (http://www.green500.org/). Second, supercomputers are complex and very expensive systems that require a very skilled team to operate them, therefore they are not publicly accessible. In that sense, many official initiatives to open HPC infrastructures to researchers have been fostered, such as the PRACE program in the European Union (http://www.prace-ri.eu/). Finally, an extensive debate over the necessity of exascale machines (capable to deliver up to the exaFLOP) has involved the research community in the past years. Despite some projects as the Human Brain Project (Markram and Henry 2012) would benefit from exascale machines, many people in the community stand up for building more regular supercomputing facilities instead of defraying the cost of such a complex system.

Parallelization and optimization to HPC architectures

In computer clusters and supercomputer architectures that are composed of multiple cores or nodes, parallelization techniques have to be applied to the software in order to take advantage of the hardware. Parallelization is the ability of a given program to perform multiple calculations at the same time. Parallelization can be implemented at hardware level (multiple instructions being executed at the same time), but the term is usually applied to the software context.

The most basic form of parallelism, *embarrassingly parallel*, is usually implemented at process level and does not require of communication between execution threads. When the different threads are in need of communication, synchronization and/or information exchange, there exist two main approaches depending on how memory can be accessed: shared or distributed memory. In shared memory, the most popular libraries are POSIX Threads (Pthreads, http://pubs.opengroup.org/onlinepubs/9699919799/mindex.html) and OpenMP (http://openmp.org). In distributed memory, MPI (Message Passing Interface) is the most popular. Both techniques can be hybridized, the *distributed shared* memory paradigm.

Software can be parallelized at different levels depending on the characteristics of the different code parts. The best theoretical speedup is defined by the Amdahl's law, despite Gustafson's law can offer a more realistic approach (McCool et al. 2012) in the parallel context. In any case, parallelization techniques can not be applied to any software and its an ongoing and exciting field of research.

Technical limitations in computational docking

The computational cost of protein-protein docking predictions varies depending on the paradigm considered. In rigid-body docking, as mentioned in previous sections, the sampling in the

six degrees of freedom (position and orientation of the ligand respecting to the receptor protein) can be performed with techniques borrowed from other computational problems such as computer vision or signal processing. New computational advances on FFT-based and patch recognition techniques have been widely explored by many docking methods. For instance, Hex (Ritchie and Venkatraman 2010) and ClusPro (Landaverde and Herbordt 2014) make use of GPU accelerators, FTDock (Gabb et al. 1997) has been successfully ported to MPI (Jiménez-García et al. 2013), FFT-based methods haven been implemented in FPGA architectures (Varma et al. 2013, 2016), and in other more exotic architectures such as the Cell BE processor (Pons et al. 2012). All of these different approaches are perfect candidates for running in HPC architectures in order to perform large-scale experiments, for instance to provide docking models for entire interactomes (Mosca et al. 2009). However, despite the technical advances in computational speed, the rigid-body paradigm suffers from several drawbacks, the most important is the explicit treatment of flexibility.

When flexibility is explicitly considered, the number of degrees of freedom of the model increases linearly with the number of atoms, in full-atom representation, or with the number or residues or beans in other coarse-grained models. Internal coordinates models may capture the dynamics of the protein in a more efficient way in terms of computational cost, but the description of the dynamics in a physically accurate way is still a challenging task. Ideally, molecular dynamics could describe complex

dynamics, but it is not feasible with the actual computational capabilities in a reasonable time, even in the largest HPC facilities. These limitations encourage new developments in the treatment of flexibility for large sets of protein complexes, and their optimization for HPC architectures.

1.6. Research software quality

Reproducibility, one of the pillars of the scientific method, is the ability of a researcher of duplicating an existing experiment or a study initially performed by other researcher. On the other hand, repeatability is the degree of agreement of tests or measurements on replications by the same observer in the same conditions. As the Irish chemist Robert Boyle argued in the 17th century, by repeating the same experiment over and over again the certainty of fact will emerge (Hannaway et al. 1988). Although these important concepts were developed while most of the scientific research was done experimentally, it is important to update them and put them in the context of the current digital era in which a significant part of scientific research relies on computational experiments.

Computers have proven to be essential tools for scientific research, and the rise of computational science and the increase of computer performance has led to impressive developments in many scientific areas such as chemistry, materials science, astrophysics, climate modeling or biology (Gilbert and Walter 1991; Aebersold et al. 2000; Misa and Thomas 2007). This fact represents a paradigm shift in many scientific disciplines where computers are now considered essential tools for collecting and managing huge amounts of data that would be impossible to analyze without them. The acquisition, analysis and management of data in big international projects such as the Human Genome Project (Lander et al. 2001) with over 3 billion base pairs, the

SETI program, with radio signals data collected over decades and analyzed used distributed grid resources (Anderson et al. 2002), or the Higgs boson discovery in CERN's Large Hadron Collider (Aad et al. 2012), 25 petabytes per year, would be simply impossible without computers.

A computer program can be considered deterministic under certain circumstances, if: i) it is a sequential program, that is, a program which is executed sequentially from the beginning to the end of the computation with no parallel threads and race conditions, and ii) the program starts always in the same conditions, that is, it does not use pseudorandom or random numbers and all the variables are correctly initialized in memory. In these conditions, we could assure the reproducibility and repeatability of the results using the same dataset as input of the program. However, these conditions are difficult to achieve, due to differences in hardware (different CPU technology, possible design bugs, etc.), operating system (different OS or even different releases) or dependencies in the libraries. But those are not the only drawbacks when dealing with reproducibility in computing software. Computer programs created in the course of research can range from single-command line scripts to multi-gigabyte code repositories. Many scientist-created programs are *ad hoc* efforts never intended for distribution or release, but very relevant pieces of code in terms of reproducibility of the experiments (Morin et al. 2012). In addition, common software engineering quality practices such as testing are often ignored during scientific software production, because they are seen as

mere prototypes and efforts tend to be focused on more intellectually challenging problems, or simply because development time is easy to be underestimated.

Reproducibility in the computational scientific research has been explored by other authors in the past. In a recent survey, scientists from different disciplines and positions were asked for questions such as sharing code and data, licensing the software, etc., with interesting results from the philosophical and ethical points of view (Stodden 2010). More recently, it was proposed a methodology to check for reproducibility in published works with the final purpose of improving reproducibility and detecting factors that hinder it (González-Barahona and Gregorio 2011). A technological solution is developed (Hinsen and Konrad 2011) where data, program code and presentation are stored together in a single file which can be executed in a cross-platform fashion thanks to the *Java Virtual Machine*. Madagascar (Freire et al. 2012) is an interesting resource on the top of the RSF file format in order to encapsulate the writing process of an article in the same platform. Other authors reviewed the problem and proposed some common guidelines to tackle it (Kauppinen and Espindola 2011) and entire issues have been dedicated to reproducibility (Fomel et al. 2009). Many principles concerning reproducibility and repeatability have been compiled and endorsed by many researches in a manifesto (http://sciencecodemanifesto.org).

Reproducibility and repeatability in the research software development remains a hot topic as many good practices set as a standard in the industry are not still being applied in. This gap between industry and research justifies more work on the problem

Fortunately, it is possible to find excellent examples of research software produced by the scientific community. GROMACS (Berendsen et al. 1995) is a free software package to perform molecular dynamics. It is used by many research groups around the world. It is a huge project in terms of lines of written code (1,735,563 lines as of January 2016) and has many contributors: 32 official ones, with a total of 37 branches and 14,735 commits (https://github.com/gromacs/gromacs). The project has an exquisite documentation both for contributing and for final users of the package, and a huge community supporting it. It is a good example of the common good practices mentioned in the previous sections of this article and a successfully case of study for future projects.

Biopython (Cock et al. 2009) is a more modest project in comparison to the GROMACS package, but it is still used and developed by many research scientists in the bioinformatics field. It has 104 contributors at present time, and a huge number of commits (10,368). Tests are especially well designed and have a good code coverage (https://github.com/biopython/biopython/).

There are other examples of excellent projects, as has been recently reported (Baxter et al. 2006).

2. Objectives

“Silence, I discover,
is something you can actually hear.”

Haruki Murakami, *Kafka on the shore*

The main purpose of this thesis is the development of computational tools for the problem of protein-protein docking and their optimization for high-performance computing. In the past years, different computational protein-protein docking tools have been developed in order to address this important problem with deep implications in the understanding of crucial cellular processes. In spite of the advances, there is a strong need for new developments to address the important conceptual and technical challenges that the field is facing. Two important aspects are especially considered during all technical and conceptual developments in this thesis: the final purpose of high-performance computing, and having in mind a series of best-practices guidelines for scientific software development. In this context, this thesis has the following specific objectives:

1. Optimization of pyDock docking method for high-performance computing architectures, in order to facilitate docking at interactome level, and efficient benchmarking.

2. Implementation of web applications for the analysis of protein-protein interactions: docking prediction, computational characterization of protein interfaces and integration of SAXS data.

3. Validation of the developed tools in the CAPRI community experiment, as well as in a new update of the Protein-Protein Docking Benchmark.

4. Compilation of the first worldwide protein-RNA docking benchmark for the evaluation of protein-RNA docking methods.

5. Development of a new protein-protein docking methodology for efficient inclusion of flexibility and multi-scale framework.

3. Articles

“Equipped with his five senses,
man explores the universe around him
and calls the adventure Science.”

Edwin Powell Hubble

3.1 Optimization of complex modeling tools for HPC architectures and implementation in web applications

Web applications are especially useful for the biology community. First, it is the easiest way to encapsulate a workflow formed by different computational tools and to make it ready for non-expert users. The research groups that make their computational tools available to the community have a centralized way to track the changes on the software and a direct feedback from their users about the usefulness of their protocols. Second, it allows the opening of many protocols to the general public, without making distinction on the software or the resources required to use it, e.g. many potential users could not have access to HPC platforms to run a specific software. Finally, many protocols might be integrated in meta servers or databases which incorporate knowledge and capabilities of heterogeneous online tools.

Three different works are presented in this section. The first manuscript, describes a web server for protein-protein complex prediction using the pyDock (Cheng et al. 2007) protocol developed in our group. The second manuscript presents the CCharPPI web server, an online tool that helps characterizing protein-protein interfaces by using up to 108 different energetic descriptors. These descriptors come from the public domain distributed software or have been be re-implemented in this web application. They can be applied to characterize experimental complex structures, but can be also a valuable tool to score docking models. The third manuscript describes a public web

server pyDockSAXS to integrate experimental SAXS data into our protein-protein docking protocol, pyDock (Pons et al. 2010).

Manuscripts presented in this section:

Jiménez-García B., Pons C., and Fernández-Recio J. (2013) **"pyDockWEB: A Web Server for Rigid-Body Protein-Protein Docking Using Electrostatics and Desolvation Scoring."** Bioinformatics 29 (13): 1698–99.

Moal IH., Jiménez-García B., and Fernández-Recio J. (2015) **"CCharPPI Web Server: Computational Characterization of Protein-Protein Interactions from Structure."** Bioinformatics 31 (1): 123–25.

Jiménez-García B., Pons C., Svergun DI., Bernadó P., and Fernández-Recio J. (2015) **"pyDockSAXS: Protein-Protein Complex Structure by SAXS and Computational Docking."** Nucleic Acids Research 43 (W1): W356–61.

3.1.1. pyDockWEB: A Web Server for Rigid-Body Protein-Protein Docking Using Electrostatics and Desolvation Scoring

Brian Jiménez-García[1], Carles Pons[1], and Juan Fernández-Recio[1*]

[1]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain*

*Corresponding author

BIOINFORMATICS APPLICATIONS NOTE Vol. 29 no. 13 2013, pages 1698–1699
doi:10.1093/bioinformatics/btt262

Structural bioinformatics

Advance Access publication May 9, 2013

pyDockWEB: a web server for rigid-body protein–protein docking using electrostatics and desolvation scoring

Brian Jiménez-García[1], Carles Pons[1,2] and Juan Fernández-Recio[1,*]

[1]Joint BSC-IRB Research Programme in Computational Biology, Department of Life Sciences, Barcelona Supercomputing Center and [2]Computational Bioinformatics, National Institute of Bioinformatics (INB), Jordi Girona 29, 08034 Barcelona, Spain

Associate Editor: Anna Tramontano

ABSTRACT

Summary: pyDockWEB is a web server for the rigid-body docking prediction of protein–protein complex structures using a new version of the pyDock scoring algorithm. We use here a new custom parallel FTDock implementation, with adjusted grid size for optimal FFT calculations, and a new version of pyDock, which dramatically speeds up calculations while keeping the same predictive accuracy. Given the 3D coordinates of two interacting proteins, pyDockWEB returns the best docking orientations as scored mainly by electrostatics and desolvation energy.

Availability and implementation: The server does not require registration by the user and is freely accessible for academics at http://life.bsc.es/servlet/pydock

Contact: juanf@bsc.es

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

Received on January 23, 2013; revised on April 8, 2013; accepted on May 2, 2013

1 INTRODUCTION

Protein–protein interactions mediate most cellular functions; thus, a detailed description of the association process at molecular level is essential to comprehend the fundamental processes that sustain life. In such line, protein–protein docking tools aim to identify the native binding mode between two proteins (Ritchie, 2008). Such predictions are required to complement experimental techniques that cannot provide structural information at a proteomics scale given their current technical limitations. pyDock (Cheng *et al.*, 2007) is a rigid-body docking method in which sampling is performed by means of FTDock (Gabb *et al.*, 1997) and scoring implements an efficient empirical potential, composed of electrostatics and desolvation terms, with a limited contribution from van der Waals energy. The method has been successfully tested in CAPRI (Grosdidier *et al.*, 2007; Mendez *et al.*, 2003; Pons *et al.*, 2010a). Here, we present pyDockWEB server, a new fast implementation that allows easy access to non-expert users to state-of-the-art docking predictions.

*To whom correspondence should be addressed.

2 PYDOCKWEB SERVER

pyDockWEB server is a web application for the use of the protein–protein docking and scoring program pyDock. Users can easily send pyDock jobs to be executed in a five-step process via a user-friendly front-end (Fig. 1). In the first step, users have to introduce a project name and a notification email address. In the second step, the scoring algorithm is selected. In the third step, users can either upload their protein coordinate files or indicate the PDB code, in which case, PDB files will be automatically downloaded from RCSB Protein Data Bank. In both cases, PDB files are automatically parsed to select available receptor and ligand chains. An option to automatically set-up a docking job with example PDB files is also available. In the fourth step, users may specify optional distance restraints, which will be computed using pyDockRST (Chelliah *et al.*, 2006) module. Finally, in the fifth step, users will double-check whether data provided are correct and submit a docking job to the server queues. After job submission, user is redirected to a web page where project status is automatically updated and result files can be downloaded after computation is finished. In this web page, the top 10 models scored by pyDock are displayed using Jmol (http://jmol.sourceforge.net/).

pyDockWEB is technically constituted by three different components: a web front-end, *pydockd*, a daemon in charge of managing pyDock executions and a data storage system. The web front-end has been implemented using JSF (Java Server Faces, a Java-based web application framework), Ajax4sf (an open source framework that adds Ajax capabilities to JSF framework) and JSP (Java Server Pages) technologies. Data storage system has been implemented via one of the most popular choices in web applications databases, MySQL (http://www.mysql.com). Data tables have been designed to efficiently store the relevant job information and to gather a few statistics about usage and computation and queued times. The controller, *pydockd*, is an application written in Python version 2.7, which periodically polls job requests created from the web front-end and stored in the MySQL database and submits them as pyDock job instances to the Slurm batch queuing system (https://computing.llnl.gov/linux/slurm/slurm.html).

pyDockWEB uses an optimized pyDock version 3, which also includes a custom parallel version of FTDock, implemented using the MPI (Message Passing Interface) library MPICH2 (Bouteiller *et al.*, 2003) to generate docking poses, which is capable to scale to multiple processors/cores. Another optimization

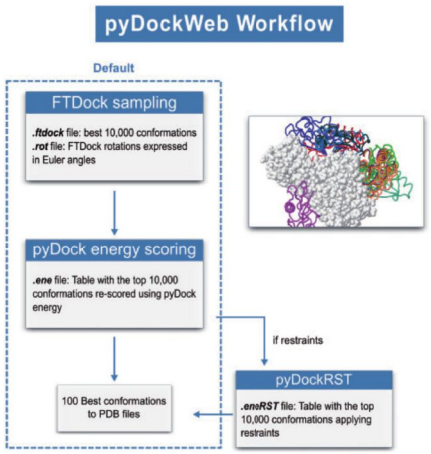


Fig. 1. pyDockWEB workflow

has been implemented, as follows. FTDock makes use of the FFTW 2.1.5 (Frigo and Johnson, 1998) library to perform a global scan of translational and rotational space having the two molecules discretized onto orthogonal grids. The size of the transform in the FFTW scope is proportional to the FTDock grid size in number of cells, which was automatically calculated from the single grid unit size and the size of the proteins. However, according to FFTW's documentation, FFTW algorithms are optimal for sizes that follow Equation (1),

$$n = 2^a \cdot 3^b \cdot 5^c \cdot 7^d \cdot 11^e \cdot 13^f \quad (1)$$

where $e+f$ is either 0 or 1, and the other exponents are arbitrary. Other sizes are calculated by FFTW using slower algorithms. Therefore, we have adjusted the FTDock grid size, n, to follow Equation (1). This grid size optimization has been implemented in the new custom parallel FTDock version. Supplementary Figure S1 shows the difference of execution times between the original and the grid optimized FTDock versions, as well as the stability in terms of time of the parallel version using the grid size optimization.

The server runs on a multi-user cluster with two nodes. Each node has 16 cores (4 Intel Xeon E5620 Quad Core) at 2.4 GHz. Two cores are reserved for MySQL, JBoss and interactive shells. Physical memory is 65 GB, with 11 TB of total available disk space.

3 BENCHMARKING AND DISCUSSION

The pyDockWEB server provides a user-friendly web front-end to allow the academic community to use the pyDock rigid-body docking and scoring method. The user is notified on completion of the execution and is able to visualize online the top 10 models of the predicted complex using Jmol. We have evaluated the performance of pyDockWEB server on the standard protein–protein docking benchmark 4.0 (Hwang *et al.*, 2010). The quality of the results in terms of generated near-native solutions (ligand RMSD within 10 Å from that in the X-ray complex structure) has not been affected by the optimization and implementation procedure, and the top 10 success rate (i.e. number of cases with near-native solutions within top 10 scored poses) reached 17.0% (Supplementary Table S1), in line with previous benchmarks (Cheng *et al.*, 2007; Pons *et al.*, 2010b). This performance is comparable with other reported servers, as shown on available protein–protein targets from current CAPRI edition (Supplementary Table S2). Interestingly, sampling with FTDock with the new custom parallel and variable grid size implementation achieved speed-ups of up to 181 (50 as average) with respect to the default FTDock distribution, whereas the scoring process based on the new pyDock version 3 achieved speed-ups of up to 40 (38 as average) with respect to the previously available version (Cheng *et al.*, 2007).

Additional pyDock modules and new developments are planned to be implemented in pyDockWEB in the future: patch prediction (pyDockNIP), optimal docking area (pyDockODA), domain–domain assembly (pyDockTET) and SIPPER scoring energy (pyDockSIPPER).

ACKNOWLEDGEMENTS

The authors would like to thank Dmitry Repchevsky for his invaluable help and experience designing and implementing Java Server Faces web applications, and all the different users that have collaborated in testing and giving feedback to improve the server.

Funding: Spanish Ministry of Science and Innovation [BIO2010-22324].

Conflict of Interest: none declared.

REFERENCES

Bouteiller,A. *et al.* (2003) MPICH-V2: a fault tolerant MPI for volatile nodes based on pessimistic sender based message logging. In: *Proceedings of the SC2003 ACM/IEEE conference on Supercomputing*. Phoenix, AZ, USA, p. 25.

Chelliah,V. *et al.* (2006) Efficent restraints for protein-protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment. *J. Mol. Biol.*, **357**, 1669–1682.

Cheng,T.M. *et al.* (2007) pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins*, **68**, 503–515.

Frigo,M. and Johnson,S.G. (1998) FFTW: an adaptive software architecture for the FFT. *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process*, **3**, 1381–1384.

Gabb,H.A. *et al.* (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. *J. Mol. Biol.*, **27**, 106–120.

Hwang,H. *et al.* (2010) Protein-protein docking benchmark version 4.0. *Proteins*, **78**, 3111–3114.

Grosdidier,S. *et al.* (2007) Prediction and scoring of docking poses with pyDock. *Proteins*, **69**, 852–858.

Mendez,R. *et al.* (2003) Assessment of blind predictions of protein-protein interactions: current status of docking methods. *Proteins*, **52**, 51–67.

Pons,C. *et al.* (2010a) Optimization of pyDock for the new CAPRI challenges: docking of homology-based models, domain-domain assembly and protein-RNA binding. *Proteins*, **78**, 3182–3188.

Pons,C. *et al.* (2010b) Present and future challenges and limitations in protein-protein docking. *Proteins*, **78**, 95–108.

Ritchie,D.W. (2008) Recent progress and future directions in protein-protein docking. *Curr. Protein Pept. Sci.*, **9**, 1–15.

Supplementary information

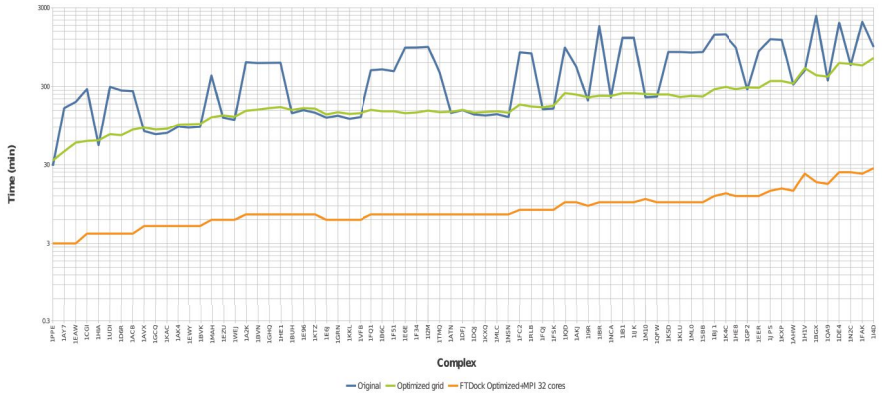


Fig. S.1 - Execution times for the different versions of FTDock.

The execution times for the final version of FTDock used in pyDockWEB server, which includes the grid optimization and the parallel implementation, is shown in orange. For comparison, the original (blue) and grid optimized (green) versions of FTDock are also shown.

Table S.1: Summary of the predictor results for the servers participating in CAPRI who obtained at least one acceptable solution for targets 46, 47, 50 or 58. The quality of the predictions has been calculated in basis to (Mendez et al., 2003): "*" indicates that at least one of the submitted predictions was in the acceptable range, "**" indicates that at least one of the submitted predictions was of medium range, "***" indicates that at least one prediction was of high accuracy (none of the servers predicted a high accuracy structure), "-" means for not even an acceptable solution found and "N/A" means for data not available (the server did not participate on that target). In the case of pyDockWEB, the predictions were calculated using the public PDB reference and the number of predictions found in top 10 is also indicated. "Predictor summary" field indicates the sum of acceptable, medium and high accuracy. Targets T48, T49, T53 and T54 have not been considered because PDB reference is not publicly available. Targets T51, T55, T56 and T57 have not been tested on pyDockWEB because the server does not offer support for multi-docking (T51), affinity prediction (T55, T56) nor polysaccharide structures (T57) at this time. For Target T58, SAXS experimental data was not used. Provided restraint data for targets T47 and T58 was used.

Predictor server	T46	T47	T50	T58	Predictor summary
ClusPro	-	-	**	-	1 / 1 **
HADDOCK	*	**	-	-	2 / 1 ** + 1 *
SwarmDock	N/A	N/A	N/A	**	1 / 1 **
DOCK/PIE	-	-	**	-	1 / 1 **
pyDockWEB	-	1*	3*	1** + 1*	3 / 1 ** + 2 **

Table S.2: Protein-Protein Benchmark 4.0 results for pyDockWEB server. First column indicates the complex, second column indicates the best ranked solution over 10000 conformations scored by pyDockWEB with L-RMSD < 10.0Å and the third column indicates the computation time in minutes (running time + queued time).

Complex	Best ranked L-RMSD < 10.0Å	Computation time (min)
1A2K	104	38.9
1ACB	385	21.8
1AHW	3718	78.5

1AK4	2099	27.2
1AKJ	453	65
1ATN	2806	54.8
1AVX	90	38.6
1AY7	18	13.7
1AZS	33	120
1B6C	2	35.9
1BGX	-	136.4
1BJ1	-	59.5
1BKD	498	134.7
1BUH	70	39.9
1BVK	24	31.4
1BVN	2	44
1CGI	10	19.1
1CLV	4	36.8
1D6R	1348	18.8
1DE4	-	220.8
1DFJ	416	45.9
1DQJ	324	46.2
1E4K	1066	76
1E6E	3	44
1E6J	28	42.4
1E96	1	33.3
1EAW	553	20.5
1EER	1707	80.7
1EFN	230	11.1
1EWY	8	29.6
1EZU	1946	41.5
1F34	208	38.6
1F51	6	46.6

1F6M	1537	39.7
1FAK	5333	89.1
1FC2	-	42.9
1FCC	453	45.4
1FFW	68	13.7
1FLE	2	22.7
1FQ1	3939	47.6
1FQJ	316	48.2
1FSK	3	48.7
1GCQ	1222	28.2
1GHQ	-	46.2
1GL1	54	23.4
1GLA	49	73.8
1GP2	-	71.4
1GPW	1	39.8
1GRN	830	43.6
1GXD	-	111.2
1H1V	-	207.8
1H9D	27	24.3
1HCF	5003	32.5
1HE1	4179	35.6
1HE8	2858	242.5
1HIA	24	18.9
1I2M	-	49.2
1I4D	-	292
1I9R	1359	109.2
1IB1	-	95.8
1IBR	-	75.1
1IJK	1357	92.2
1IQD	10	60.2

1IRA	-	41.9
1J2J	24	22.1
1JIW	3022	51.3
1JK9	321	29.8
1JMO	5253	87.2
1JPS	550	72.6
1JTG	1	47.3
1JWH	-	131.6
1JZD	256	63.1
1K4C	-	89.7
1K5D	345	72.8
1K74	12	52.5
1KAC	1564	27.3
1KKL	47	56.9
1KLU	1479	85.4
1KTZ	3483	23.6
1KXP	22	88.1
1KXQ	237	50.5
1LFD	480	23.4
1M10	79	49.6
1MAH	28	48.8
1ML0	108	72.4
1MLC	15	61.2
1MQ8	-	42.4
1N2C	-	333.6
1N8O	63	31.4
1NCA	5	110.9
1NSN	381	47.4
1NW9	13	31.4
1OC0	85	33.9

1OFU	-	47.5
1OPH	19	56.3
1OYV	51	28.6
1PPE	3	14.2
1PVH	972	51.1
1PXV	2100	30.3
1QA9	7624	110.2
1QFW	174	46
1R0R	5	22
1R6Q	81	37.3
1R8S	-	42.2
1RLB	3188	53.7
1RV6	1	26.1
1S1Q	1207	16
1SBB	221	51.4
1SYX	629	17
1T6B	30	102.3
1TMQ	1	54.2
1UDI	1	22.8
1US7	1078	28.7
1VFB	30	30
1WDW	-	316.3
1WEJ	228	39.6
1WQ1	2380	58.5
1XD3	1	19.1
1XQS	14	59.6
1XU1	12	39.1
1Y64	-	335.6
1YVB	26	46.5
1Z0K	10	18.6

1Z5Y	19	21.9
1ZHH	-	52.7
1ZHI	3	36.5
1ZLI	-	53.4
1ZM4	-	384.9
2A5T	175	49.4
2A9K	323	40.5
2ABZ	5	30.4
2AJF	1737	74.6
2AYO	23	51.7
2B42	2	47.7
2B4J	1622	30.7
2BTF	33	43.9
2C0L	3786	36.3
2CFH	2144	31.9
2FD6	22	72.8
2FJU	-	93.7
2G77	13	50.4
2H7V	-	58.7
2HLE	13	32
2HMI	-	271.2
2HQS	31	43.8
2HRK	17	27.8
2I25	44	23.5
2I9B	662	37.8
2IDO	101	19.4
2J0T	2534	23.7
2J7P	-	70.4
2JEL	23	57.5
2MTA	73	50.9

2NZ8	8	50.1
2O3B	349	31
2O8V	26	24.2
2OOB	110	10.5
2OOR	-	112.8
2OT3	5	39.2
2OUL	1	31.8
2OZA	-	59.3
2PCC	7	44.4
2SIC	12	30.9
2SNI	4	26.3
2UUY	4206	19.2
2VBD	3	56.1
2VIS	-	234.4
2Z0E	-	50
3BP8	499	95.1
3CPH	1012	56.1
3D5S	164	29.6
3SGQ	156	14
4CPA	11	26
7CEI	18	15.8
9QFW	-	46.3
BOYV	-	35.4

3.1.2. CCharPPI web server: computational characterization of protein-protein interactions from structure

Iain H. Moal[1], Brian Jiménez-García[1] and Juan Fernández-Recio[1*]

[1]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain*

*Corresponding author

BIOINFORMATICS APPLICATIONS NOTE Vol. 31 no. 1 2015, pages 123–125
doi:10.1093/bioinformatics/btu594

Structural bioinformatics

Advance Access publication September 2, 2014

CCharPPI web server: computational characterization of protein–protein interactions from structure

Iain H. Moal†, Brian Jiménez-García† and Juan Fernández-Recio*

Joint BSC-IRB Research Programme in Computational Biology, Department of Life Sciences, Barcelona Supercomputing Center, C/Jordi Girona 29, 08034 Barcelona, Spain

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: The atomic structures of protein–protein interactions are central to understanding their role in biological systems, and a wide variety of biophysical functions and potentials have been developed for their characterization and the construction of predictive models. These tools are scattered across a multitude of stand-alone programs, and are often available only as model parameters requiring reimplementation. This acts as a significant barrer to their widespread adoption. CCharPPI integrates many of these tools into a single web server. It calculates up to 108 parameters, including models of electrostatics, desolvation and hydrogen bonding, as well as interface packing and complementarity scores, empirical potentials at various resolutions, docking potentials and composite scoring functions.

Availability and implementation: The server does not require registration by the user and is freely available for non-commercial academic use at http://life.bsc.es/pid/ccharppi

Contact: juanf@bsc.com

Received on June 5, 2014; revised on August 3, 2014; accepted on August 28, 2014

1 INTRODUCTION

Protein–protein interactions are involved in most cell processes, and their structural and functional annotation is essential to understand biological and pathological phenomena and to develop new therapeutic approaches. The increasing volume of experimental data on protein–protein interactions at the molecular level offers many opportunities for functional characterization and the construction of predictive models based on properties arising from structure, such as interface geometry, hydrogen bonding, electrostatics and desolvation energy, which act as an intermediate layer between structure and function (Chothia and Janin, 1975; Jones and Thornton, 1997). Indeed, the selection and combination of structure-based potentials within a learning framework have been used for many tasks, often beyond their original development purpose, such as the prediction of binding affinity (Moal *et al.*, 2011), kinetics (Moal and Bates, 2012), mutational effects (Agius *et al.*, 2013; Moretti *et al.*, 2013; Pallara *et al.*, 2013), interface design (Fleishman *et al.*, 2011; Yu *et al.*, 2012), protein–protein docking (Moal *et al.*, 2013b) and the detection of hotspots (Lise *et al.*, 2009; Zhu and Mitchell 2011), with many further possibilities remaining to be explored (Moal *et al.*, 2013a). Although many tools have been developed to calculate structural properties, some of which are available online (Tuncbag *et al.*, 2009), their availability and ease of use are an impediment, often requiring the installation of stand-alone programs with different library dependencies, reimplementation of models for which only parameters are given and reformatting of pdb files. Thus, there is a need to consolidate these methods into a single implementation. Here, we present CCharPPI, a web server, which gathers together a large number of these functions, including those on which many of our previous models were based, into a single easy-to-use interface.

2 THE WEB SERVER

CCharPPI incorporates many parameter calculation tools into a single web application, which is freely available for academic non-commercial use. Up to 108 intermolecular parameters are calculated for the input protein–protein interface/s, including 43 potential functions, which have been reimplemented (Chuang *et al.*, 2008; Feng *et al.*, 2010; Lu *et al.*, 2003; Liu and Vakser, 2011; Liu *et al.*, 2004; Mintseris *et al.*, 2007; Moal and Fernandez-Recio, 2013; Pokarowski *et al.*, 2005; Rajgaria *et al.*, 2008, 2006; Shen and Sali, 2006; Tobi, 2010; Tobi and Bahar, 2006), as well as terms calculated with 11 stand-alone programs (Feliu *et al.*, 2011; Li and Liang, unpublished; Lu *et al.*, 2008; Mitra and Pal, 2010; Pierce and Weng, 2007, 2008; Ravikant and Elber, 2010; Viswanath *et al.*, 2013; Yang and Zhou, 2008a,b; Zhang and Zhang, 2010; Zhou and Skolnick, 2011) and 4 packages: FireDock (Andrusier *et al.*, 2007), PyRosetta (Chaudhury *et al.*, 2010), SIPPER (Pons *et al.*, 2011) and PyDock (Cheng *et al.*, 2007). A detailed list of individual parameters is given online (http://life.bsc.es/pid/ccharppi/info/faq_and_help#descriptors). Users can easily calculate descriptors of interest using a clear workflow. There are three different input sources: a protein databank ID code for automatic retrieval, an uploaded complex in PDB format, or a compressed batch job file for analysing multiple interfaces, for instance, those derived from docking predictions. The web front end acts as user input source and makes results available for display and download. The back end polls for queued projects and schedules jobs for parallel execution. The distribution of descriptor values can be visualized by clicking the descriptor name on the results page. For comparison, values are shown against a background distribution pre-calculated using a set of diverse non-redundant

*To whom correspondence should be addressed.
†The authors wish it be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

complexes of known affinity (Kastritis *et al.*, 2011), with the relationship between affinity values and the pre-calculated descriptor values indicated by a scatter plot. Two pre-calculated datasets are available from the website. The first consists of the structural affinity benchmark (Kastritis *et al.*, 2011), a set of 144 complexes with experimentally determined affinity. The second consists of 157 wild-type complexes and 2731 unique mutations in the SKEMPI database (Moal and Fernandez-Recio, 2012), as modelled using FoldX (Guerois *et al.*, 2002). Computational time for calculating all descriptors is typically <5 min and took <15 min for the largest complex tested (the FAB/influenza haemagglutinin, PDBid 2VIS). Calculations are quicker when executed in parallel using the batch mode, with the 157 wild-type and 2731 unique mutants in the SKEMPI set taking 18 h to complete, and the 144 complexes in the structural affinity benchmark completing in 1 h 20 min. The server has been tested on major browser for MacOS, Ubuntu 12.4, Windows 7 and Windows 8.

3 CONCLUSIONS

In conclusion, we have brought together many different methods for characterizing protein–protein interactions, and provide pre-calculated descriptors for two datasets, one of which is also used to provide a visual comparison of uploaded complexes with complexes of known affinity. The ease with which these descriptors can be calculated can accelerate the prototyping of reproducible predictive models, allow users to mix and match different functional forms to model physical phenomena, find new terms for their scoring functions and characterize their complexes of interest. For researchers interested in local execution or incorporation into their own software, all scripts and code are available on request. We intent to expand the pre-calculated datasets, as well as the features as new methods become available.

ACKNOWLEDGEMENTS

The methods implemented have come from many laboratories, and the authors would like to thank all those who have made their parameters, code and software available, and for clarifying questions regarding licencing.

Funding: The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/2007-2013) under REA grant agreement PIEF-GA-2012-327899 and grant BIO2013-48213-R from Spanish Ministry of Economy and Competitiveness.

Conflict of interest: none declared.

REFERENCES

Agius,R. *et al.* (2013) Characterizing changes in the rate of protein-protein dissociation upon interface mutation using hotspot energy and organization. *PLoS Comput. Biol.*, **9**, e1003216.

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

Chaudhury,S. *et al.* (2010) PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics*, **26**, 689–691.

Cheng,T.M. *et al.* (2007) pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins*, **68**, 503–515.

Chothia,C. and Janin,J. (1975) Principles of protein-protein recognition. *Nature*, **256**, 705–708.

Chuang,G.Y. *et al.* (2008) DARS (Decoys As the Reference State) potentials for protein-protein docking. *Biophys. J.*, **95**, 4217–4227.

Feliu,E. *et al.* (2011) On the analysis of protein-protein interactions via knowledge-based potentials for the prediction of protein-protein docking. *Protein Sci.*, **20**, 529–541.

Feng,Y. *et al.* (2010) Potentials'R' Us web-server for protein energy estimations with coarse-grained knowledge-based potentials. *BMC Bioinformatics*, **11**, 92.

Fleishman,S.J. *et al.* (2011) Community-wide assessment of protein-interface modeling suggests improvements to design methodology. *J. Mol. Biol.*, **414**, 289–302.

Guerois,R. *et al.* (2002) Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J. Mol. Biol.*, **320**, 369–387.

Jones,S. and Thornton,J.M. (1997) Analysis of protein-protein interaction sites using surface patches. *J. Mol. Biol.*, **272**, 121–132.

Kastritis,P. *et al.* (2011) A structure-based benchmark for protein-protein binding affinity. *Protein Sci.*, **20**, 482–491.

Lise,S. *et al.* (2009) Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. *BMC Bioinformatics*, **10**, 365.

Liu,S. and Vakser,I.A. (2011) DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking. *BMC Bioinformatics*, **12**, 280.

Liu,S. *et al.* (2004) A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. *Proteins*, **56**, 93–101.

Lu,H. *et al.* (2003) Development of unified statistical potentials describing protein-protein interactions. *Biophys. J.*, **84**, 1895–1901.

Lu,M. *et al.* (2008) OPUS-PSP: an orientation-dependent statistical all-atom potential derived from side-chain packing. *J. Mol. Biol.*, **376**, 288–301.

Mintseris,J. *et al.* (2007) Integrating statistical pair potentials into protein complex prediction. *Proteins*, **69**, 511–520.

Mitra,P. and Pal,D. (2010) New measures for estimating surface complementarity and packing at protein-protein interfaces. *FEBS Lett.*, **584**, 1163–1168.

Moal,I.H. and Bates,P.A. (2012) Kinetic rate constant prediction supports the conformational selection mechanism of protein binding. *PLoS Comput. Biol.*, **8**, e1002351.

Moal,I.H. and Fernandez-Recio,J. (2012) SKEMPI: a Structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, **28**, 2600–2607.

Moal,I.H. and Fernandez-Recio,J. (2013) Intermolecular contact potentials for protein-protein interactions extracted from binding free energy changes upon mutation. *J. Chem. Theory Comput.*, **9**, 3715–3727.

Moal,I.H. *et al.* (2011) Protein-protein binding affinity prediction on a diverse set of structures. *Bioinformatics*, **27**, 3002–3009.

Moal,I.H. *et al.* (2013a) Scoring functions for protein-protein interactions. *Curr. Opin. Struct. Biol.*, **23**, 862–867.

Moal,I.H. *et al.* (2013b) The scoring of poses in protein-protein docking: current capabilities and future directions. *BMC Bioinformatics*, **14**, 286.

Moretti,R. *et al.* (2013) Community-wide evaluation of methods for predicting the effect of mutations on protein-protein interactions. *Proteins*, **81**, 1980–1987.

Pallara,C. *et al.* (2013) Expanding the frontiers of protein-protein modeling: from docking and scoring to binding affinity predictions and other challenges. *Proteins*, **81**, 2192–2200.

Pierce,B. and Weng,Z. (2007) ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins*, **67**, 1078–1086.

Pierce,B. and Weng,Z. (2008) A combination of rescoring and refinement significantly improves protein docking performance. *Proteins*, **72**, 270–279.

Pokarowski,P. *et al.* (2005) Inferring ideal amino acid interaction forms from statistical protein contact potentials. *Proteins*, **59**, 49–57.

Pons,C. *et al.* (2011) Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein-protein docking. *J. Chem. Inf. Model.*, **51**, 370–377.

Rajgaria,R. *et al.* (2006) A novel high resolution Calpha–Calpha distance dependent force field based on a high quality decoy set. *Proteins*, **65**, 726–741.

Rajgaria,R. *et al.* (2008) Distance dependent centroid to centroid force fields using high resolution decoys. *Proteins*, **70**, 950–970.

Ravikant,D.V. and Elber,R. (2010) PIE-efficient filters and coarse grained potentials for unbound protein-protein docking. *Proteins*, **78**, 400–419.

Shen,M.Y. and Sali,A. (2006) Statistical potential for assessment and prediction of protein structures. *Protein Sci.*, **15**, 2507–2524.

Tobi,D. (2010) Designing coarse grained-and atom based-potentials for protein-protein docking. *BMC Struct. Biol.*, **10**, 40.

Tobi,D. and Bahar,I. (2006) Optimal design of protein docking potentials: efficiency and limitations. *Proteins*, **62**, 970–981.

Tuncbag,N. *et al.* (2009) A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Brief. Bioinformatics*, **10**, 217–232.

Viswanath,S. *et al.* (2013) Improving ranking of models for protein complexes with side chain modeling and atomic potentials. *Proteins*, **81**, 592–606.

Yang,Y. and Zhou,Y. (2008a) *Ab initio* folding of terminal segments with secondary structures reveals the fine difference between two closely related all-atom statistical energy functions. *Protein Sci.*, **17**, 1212–1219.

Yang,Y. and Zhou,Y. (2008b) Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins*, **72**, 793–803.

Yu,C.M. *et al.* (2012) Rationalization and design of the complementarity determining region sequences in an antibody-antigen recognition interface. *PLoS One*, **7**, e33340.

Zhang,J. and Zhang,Y. (2010) A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PLoS One*, **5**, e15386.

Zhou,H. and Skolnick,J. (2011) GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophys. J.*, **101**, 2043–2052.

Zhu,X. and Mitchell,J.C. (2011) KFC2: a knowledge-based hot spot prediction method based on interface solvation, atomic density, and plasticity features. *Proteins*, **79**, 2671–2683.

3.1.3. pyDockSAXS: protein-protein complex structure by SAXS and computational docking

Brian Jiménez-García[1], Carles Pons[2], Dmitri I. Svergun[3], Pau Bernadó[4] and Juan Fernández-Recio[1*]

[1]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain*

[2]*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA*

[3]*European Molecular Biology Laboratory, Hamburg Outstation, 22603 Hamburg, Germany*

[4]*Centre de Biochimie Structurale, INSERM U1054, CNRS UMR 5048, Universite´ Montpellier 1 and 2, F-34090 Montpellier, France*

*Corresponding author

W356–W361 *Nucleic Acids Research, 2015, Vol. 43, Web Server issue* *Published online 20 April 2015*
doi: 10.1093/nar/gkv368

pyDockSAXS: protein–protein complex structure by SAXS and computational docking

Brian Jiménez-García[1], Carles Pons[2], Dmitri I. Svergun[3], Pau Bernadó[4] and Juan Fernández-Recio[1,*]

[1]Joint BSC-CRG-IRB Research Program in Computational Biology, Barcelona Supercomputing Center, 08034 Barcelona, Spain, [2]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA, [3]European Molecular Biology Laboratory, Hamburg Outstation, 22603 Hamburg, Germany and [4]Centre de Biochimie Structurale, INSERM U1054, CNRS UMR 5048, Université Montpellier 1 and 2, F-34090 Montpellier, France

Received February 07, 2015; Revised March 21, 2015; Accepted April 02, 2015

ABSTRACT

Structural characterization of protein–protein interactions at molecular level is essential to understand biological processes and identify new therapeutic opportunities. However, atomic resolution structural techniques cannot keep pace with current advances in interactomics. Low-resolution structural techniques, such as small-angle X-ray scattering (SAXS), can be applied at larger scale, but they miss atomic details. For efficient application to protein–protein complexes, low-resolution information can be combined with theoretical methods that provide energetic description and atomic details of the interactions. Here we present the pyDockSAXS web server (http://life.bsc.es/pid/pydocksaxs) that provides an automatic pipeline for modeling the structure of a protein–protein complex from SAXS data. The method uses FTDOCK to generate rigid-body docking models that are subsequently evaluated by a combination of pyDock energy-based scoring function and their capacity to describe SAXS data. The only required input files are structural models for the interacting partners and a SAXS curve. The server automatically provides a series of structural models for the complex, sorted by the pyDockSAXS scoring function. The user can also upload a previously computed set of docking poses, which opens the possibility to filter the docking solutions by potential interface residues or symmetry restraints. The server is freely available to all users without restriction.

INTRODUCTION

Protein–protein interactions orchestrate the vast majority of biological processes in cell. The atomic level description of these interactions, the so-called interactome (1), gives access to the molecular bases of biological activity and the eventual rational intervention for medical purposes. At present, only a tiny fraction of complexes from the estimated number of all possible protein–protein interactions (2) have an available 3D structure due to the limitations of high-resolution structural biology methods, such as X-ray crystallography or nuclear magnetic resonance (NMR) (3). Fortunately, low-resolution methods, especially small-angle scattering (SAS), are of more general application and could be applied in a high-throughput fashion as compared to X-ray crystallography or NMR techniques (4–5).

Small-angle X-ray scattering (SAXS) is a powerful methodology for the structural and dynamic characterization of biomolecules at low resolution (6–9). Recent advances in SAXS instrumentation and the development of software for the comprehensive interpretation of SAXS data in terms of structure make this technique an optimal tool to address the structural characterization of the interactome. Methods based on rigid-body modeling of SAXS data, such as SASREF (10), can generate structural models for protein–protein complexes by simultaneously fitting multiple SAXS/SANS data using simulated annealing algorithm. However, given that these methods rely exclusively on the SAS data, the resulting models display an inherent degeneracy. In addition, these techniques miss the high-resolution information reporting on the details of intermolecular interactions. Therefore, other strategies are necessary to incorporate the interacting surfaces of the partners to enrich the quality of the resulting models. One such strategy is the use of SAXS data in combination with advanced computational approaches, such as protein–protein dock-

*To whom correspondence should be addressed. Tel: +34 93 413 77 29; Fax: +34 93 413 77 21; Email: juanf@bsc.es
Correspondence may also be addressed to Pau Bernadó. Tel: +33 4 67 41 7705; Fax: +33 4 67 41 7913; Email: pau.bernado@cbs.cnrs.fr

Nucleic Acids Research, 2015, Vol. 43, Web Server issue W357

ing, to generate meaningful models of biomolecular assemblies.

Several docking methods for structural prediction of protein–protein interactions have been reported. These methods are mostly based on rigid-body (or semi-flexible) sampling of the interacting molecules, followed by scoring and/or energy minimization (11–15). Completely automatic docking can provide good models for specific protein–protein interactions (16–20). However, the recent CAPRI experiments (http://www.ebi.ac.uk/msd-srv/capri/) (21–25) have highlighted the limitations of current docking approaches and the necessity of using experimental information to help to identify the correct docking models (20,26).

Computational docking tools can be used to generate a large number of poses that are subsequently filtered and scored based on their capacity to describe the experimental data. This strategy has been applied to specific cases (27–30) and has been implemented and systematically benchmarked in a few computational methods that combine SAXS and docking for the structural modeling of protein–protein complexes, such as pyDockSAXS (31), FoXSDock (32) or HADDOCK (33). Among them, we previously reported the first of such methods, pyDockSAXS (31), which provided a 2-fold increase in the success rate for the prediction of protein complexes as compared to that of the individual approaches based on energy-based docking or SAXS data alone (31). Here, a server that makes pydockSAXS available is described. This server provides comprehensive structural models of biomolecular assemblies using the experimental SAXS curve and the structure of the interacting partners as the only input. This strategy can be efficiently used for the high-throughput resolution of protein complexes at large scale with SAXS data.

MATERIALS AND METHODS

The pyDockSAXS method integrates SAXS data and pyDock energy-based scoring (16) to determine the structure of a protein–protein complex from its components.

This integrative method uses FTDock to generate 10 000 rigid-body docking poses, which are re-scored by a combination of pyDock energy and the χ value defining the goodness of fit to the SAXS data computed with CRYSOL 2.8 (34):

$$\text{pyDockSAXS} = E_{\text{pyDock}} + w_c \cdot \chi_{\text{CRYSOL}}, \qquad (1)$$

where w_c is a parameter that was previously optimized on 62 cases of the protein–protein docking benchmark 2.0, using synthetic SAXS data obtained from the complex structures after adding noise.

The structural modeling capabilities of the server have been validated on 81 complexes of the Protein–Protein Benchmark 4.0 (35) which were not present in the previous training of the scoring function, using SAXS data synthetically obtained from the complex structure after adding noise. We considered only complexes in which the molecular mass did not significantly vary between the unbound and the complex structures, as previously described (31). Figure 1 shows the predictive success rates obtained in this benchmark. The pyDockSAXS server identifies an acceptable docking model (i.e. with ligand RMSD < 10 Å from the reference structure after superimposing the receptor molecules) within the top 10 predictions in 25.9% of the cases (as compared to 13.6% success rate when using energy-based scoring alone) (Figure 1). This is a similar improvement as that previously reported for the stand-alone version on the benchmark 2.0 (31). SAXS-based scoring is sensitive to large conformational changes between the unbound structures (used in docking) and the bound state (from which SAXS data are derived). Indeed, in rigid cases, i.e. those with unbound–bound interface Cα RMSD < 1.5 Å, the pyDockSAXS server improved the top 10 success rate up to 36.5% (as compared to 15.4% when using docking alone). This means that in rigid cases, the SAXS-based scoring is more efficient in identifying the correct docking models. On the other hand, the overall results strongly depend on the quality of the docking poses generated by FTDock. When considering only those rigid cases in which FTDock is able to generate at least a near-native solution with ligand RMSD < 5 Å, the success rate for pyDockSAXS is 47.4% (as compared to 26.3% for docking alone). This observation suggests that future improvements in the docking algorithm used to generate the docking poses will have a strong impact on the predictive capabilities of the server.

We have also successfully validated the server on experimental systems of interest. As an example, we have applied pyDockSAXS to rebuild the structure of the *Alvinella pompejana* Cu,Zn superoxide dismutase homo-dimer (PDB 3F7L), using the X-ray coordinates of one monomer (chain A) (36) and the experimental SAXS data deposited in Biosis database (37). This complex presents a spherical shape, which is challenging for modeling based only on SAXS data (31). Thus, it represents an excellent case to test the robustness of the method. The server finds a near-native docking solution as rank 1, and additional acceptable solutions within the top 10 docking models. Actually, six of the top 10 docking models were within (or slightly above) acceptance criteria in CAPRI (Figure 2). However, the other four docking models (not shown in Figure 2) were significantly far from the correct orientation, which indicates that docking results in blind conditions should always be considered with caution.

As another example, we used the pyDockSAXS server to model the complex between the redox proteins adrenodoxin (Adx) and cytochrome c (Cc) which has been identified as a short-lived encounter complex (38). The authors stabilized the complex by engineering both proteins in order to cross-link them using two cysteine mutants: L80C and V28C from Adx and Cc, respectively. The cross-linked complex was structurally characterized by NMR and SAXS (38). This is a challenging case involving expectedly weak interaction forces given its transient nature. In this type of cases, pyDockSAXS can be easily used to generate models compatible with the experimental SAXS profile and energetically accurate. Using the experimental SAXS data stored in the SASBDB repository (39), and the X-ray structures of Adx (PDB 1AYF) and Cc (PDB 2YCC), the pyDockSAXS server generated many different docking orientations. After manually filtering the results from the pyDockSAXS server to keep only the docking poses with the residues Adx C80 and Cc C28 within 10-Å distance in order to describe the cross-linked complex, a model similar

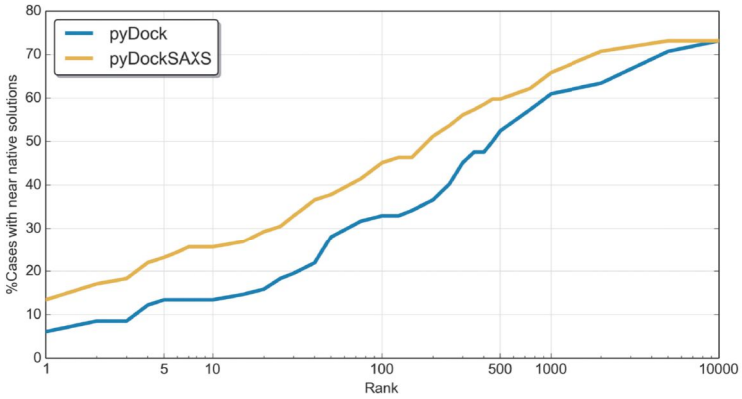


Figure 1. Success rate for pyDockSAXS on a set of 81 cases of protein–protein docking benchmark 4.0 which were not used for training, as compared to that of pyDock alone.

to the NMR structure (PDB 2JQR) was found within the top 10 pyDockSAXS docking poses. The other nine of the top 10 docking poses showed large variability in the mutual orientation between the two molecules. Interestingly, without using the SAXS data, this near-native solution would not have been identified within the top 10 docking poses. This example highlights the capacities of integrating SAXS data with computational docking, and the power that additional residue-specific information has to enrich final solutions. However, while pyDockSAXS provides a reduced set of models that typically includes one or several correct solutions, the existence of high-scoring incorrect models could complicate the identification of the correct assembly.

DESCRIPTION OF THE WEB SERVER

Input

The user is requested to upload the structure files for the two interacting proteins in the Protein Data Bank (PDB) format (40). The choice of molecules as receptor or ligand is arbitrary, although for the sake of efficiency it is recommended to set the receptor as the largest molecule. The user can specify the exact chains that will be included for modeling. Incomplete residues are rebuilt with SCWRL 3.0 (41). At present, cofactors are not considered in the calculations but this possibility will be implemented in future versions of the server. In addition, the server expects a file with the SAXS experimental curve compatible with CRYSOL software. Thus, it should be a plain-text file where the first line is a title ignored by the software and the following lines are composed by three columns of numerical data separated by blanks or commas, which represent momentum transfer, scattering intensity and experimental error, respectively. If experimental errors are not specified, they are automatically estimated by CRYSOL (2% of intensity values). All input file formats are described in the help section of the server.

Users can customize some CRYSOL execution parameters. At present, the available options for CRYSOL calculations implemented in pyDockSAXS are: (i) the use or not of constant subtraction and (ii) to specify different angular units of the SAXS experimental data provided. Other parameters such as the number of spherical harmonics are set to their standard values that have been proven to provide accurate estimation of theoretical SAXS curves.

The option of specifying a rigid-body docking set from previous pyDockWeb (42) executions has also been implemented for the convenience of advanced users. This option allows the user to upload pre-filtered rigid-body docking poses to be evaluated by the server. This could be used to include residue–residue distance restraints based on binding site residues, already implemented in the general pyDockWeb server (26), or to filter manually specific orientations of the complex by the user. This possibility is relevant when residue-specific information is available from other techniques, i.e. NMR, mutagenesis data or bioinformatics tools.

Output and representation of results

After submitting the job for calculation, the user is redirected to the job information and results page. This page is unique for the job and its URL is highly recommended to bookmark, if a contact e-mail address was not provided by the user. The job information and results page is periodically auto-refreshed to provide the user updated information of the status of the submitted job. Once the calculation has finished, the results are shown in this page. The information displayed is (i) an energy table of the top 100 complex orientations predicted and scored by pyDockSAXS (including other relevant energetic terms as pyDock scoring energy and CRYSOL χ^2 value) and available to download as a PDF format file (Figure 2), (ii) a graphical representation of the fitting of the top 10 docking mod-

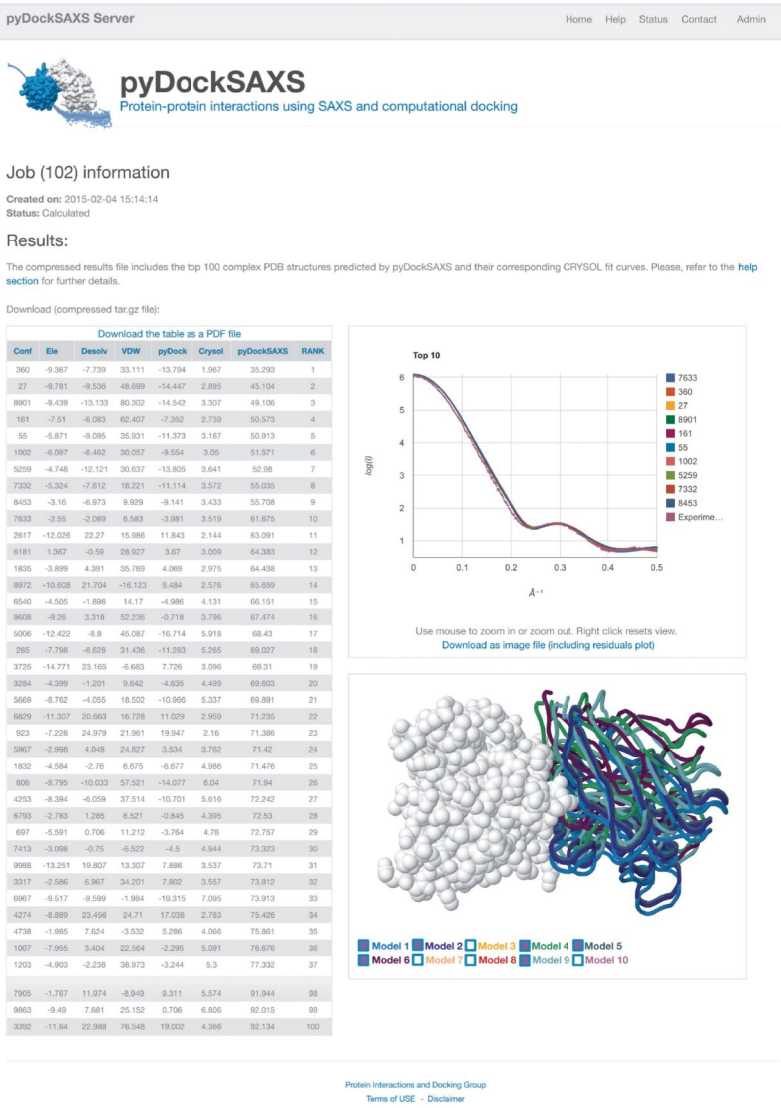


Figure 2. Output of the pyDockSAXS server showing the results for rebuilding the dismutase oxidase homo-dimer (PDB 3F7L). Models 1 and 2 represent near-native solutions (ligand RMSD < 10 Å). Model 9 would also be acceptable by CAPRI criteria, since interface RMSD < 4 Å. Other models (e.g. 4, 5, 6) have also good interface-RMSD values just above the usual acceptance cutoff.

els to the experimental SAXS data provided and (iii) a JSmol (jsmol.sourceforge.net/) interactive representation of the top 10 models predicted by the server (Figure 2). The output of the server is also available for downloading as a gzip (gzip.org) compressed tar file and includes all the result files organized by folders. Those folders are (i) 'input_data' which include the different input files provided by the user, (ii) 'pydock' with the protein–protein docking information data generated by pyDock method, (iii) 'fit_top10_SAXS' contains the fitting files for the top 10 docking orientations according to CRYSOL χ^2 value and (iv) 'top100' folder, containing the top 100 structures scored by pyDockSAXS in PDB file format (the CRYSOL fit parameters are included in the header of each structure as a 'REMARK' section for user convenience). The organization and format of the result files has been carefully optimized following the feedback provided by community users of the server and it is well described in the 'FAQ and Help' section of the server as well as in the 'README.txt' file included in the compressed results.

Implementation

The implementation of the web server is based on a three-components architecture: (i) a web front end that acts as user input source and makes results available to display and download when job is completed, (ii) a relational database where the job information is stored and (iii) a back end application which periodically polls the database for queued user projects and schedules jobs for parallel calculation of pyDockSAXS using the Slurm batch queuing system (slurm.schedmd.com). The web front end has been implemented using the web2py (www.web2py.com) free and open source web framework, and has been tested in all major modern web browsers. In addition, it adapts fluently to mobile devices screens. The back end application has been written in Python version 2.7 with the use of external libraries as numpy and matplotlib. The relational database has been designed and implemented using MySQL (www.mysql.com). The pyDockSAXS method is part of the pyDock software version 3 and calls internally CRYSOL software to evaluate the fitness of each of the predicted protein–protein complexes to the SAXS experimental data.

The server runs on a multi-user cluster with access to two nodes composed of 16 cores (4 Intel Xeon E5620 Quad Core at 2.4 GHz) and 32 cores (2 AMD Opteron Abu Dhabi 6376 cpus), respectively, with 11 TB of total available disk space and 256 GB of physical memory.

CONCLUSIONS AND FUTURE DEVELOPMENT

The motivation behind the pyDockSAXS web server was to provide access to the scientific community to the efficient pyDockSAXS method, which integrates SAXS experimental data with pyDock protein–protein scoring energy for improved structural predictions of protein–protein complexes.

The pyDockSAXS web server is an on-going project that will implement new features according to the future scientific community feedback. In the next upgrade, cofactors, ions and other non-peptidic molecules will be able to be considered during calculations. We also plan to implement a filter by symmetry, for the use on homo-meric complexes, and an extended input to analyze docking sets from different docking methods.

FUNDING

Programa Estatal I+D+i, the Spanish Ministry of Economy and Competitiveness [BIO2013-48213-R to J.F.-R.]; Agence Nationale de la Recherche [SPIN-HD-ANR-CHEX-2011 and ATIP-Avenir Program to P.B.]; FPU Fellowship, the Spanish Ministry of Science and Innovation [BES-2011-045634 to B.J.G.]. Funding for open access charge: Spanish Ministry of Economy and Competitiveness [BIO2013-48213-R]; Agence Nationale de la Recherche [SPIN-HD-ANR-CHEX-2011].

Conflict of interest statement. None declared.

REFERENCES

1. Sanchez,C., Lachaize,C., Janody,F., Bellon,B., Roder,L., Euzenat,J., Rechenmann,F. and Jacq,B. (1999) Grasping at molecular interactions and genetic networks in Drosophila melanogaster using FlyNets, an Internet database. *Nucleic Acids Res.*, **27**, 89–94.
2. von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.
3. Mosca,R., Ceol,A. and Aloy,P. (2013) Interactome3D: adding structural details to protein networks. *Nat. Methods*, **10**, 47–53.
4. Jeffries,C.M. and Svergun,D.I. (2015) High-throughput studies of protein shapes and interactions by synchrotron small-angle X-ray scattering. *Methods Mol. Biol.*, **1261**, 277–301.
5. Bernadó,P. (2011) Low-resolution structural approaches to study biomolecular assemblies. *WIRES Comput. Mol. Sci.*, **1**, 283–297.
6. Koch,M.H., Vachette,P. and Svergun,D.I. (2003) Small-angle scattering: a view on the properties, structures and structural changes of biological macromolecules in solution. *Q. Rev. Biophys.*, **36**, 147–227.
7. Mertens,H.D. and Svergun,D.I. (2010) Structural characterization of proteins and complexes using small-angle X-ray solution scattering. *J. Struct. Biol.*, **172**, 128–141.
8. Putnam,C.D., Hammel,M., Hura,G.L. and Tainer,J.A. (2007) X-ray solution scattering (SAXS) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution. *Q. Rev. Biophys.*, **40**, 191–285.
9. Jacques,D.A. and Trewhella,J. (2010) Small-angle scattering for structural biology—expanding the frontier while avoiding the pitfalls. *Protein Sci.*, **19**, 642–657.
10. Petoukhov,M.V. and Svergun,D.I. (2005) Global rigid body modeling of macromolecular complexes against small-angle scattering data. *Biophys. J.*, **89**, 1237–1250.
11. Fernandez-Recio,J., Totrov,M. and Abagyan,R. (2003) ICM-DISCO docking by global energy optimization with fully flexible side-chains. *Proteins*, **52**, 113–117.
12. Smith,G.R. and Sternberg,M.J. (2002) Prediction of protein-protein interactions by docking methods. *Curr. Opin. Struct. Biol.*, **12**, 28–35.
13. Bonvin,A.M. (2006) Flexible protein-protein docking. *Curr. Opin. Struct. Biol.*, **16**, 194–200.
14. Gray,J.J. (2006) High-resolution protein-protein docking. *Curr. Opin. Struct. Biol.*, **16**, 183–193.
15. Ritchie,D.W. (2008) Recent progress and future directions in protein-protein docking. *Curr. Protein Pept. Sci.*, **9**, 1–15.
16. Cheng,T.M., Blundell,T.L. and Fernandez-Recio,J. (2007) pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. *Proteins*, **68**, 503–515.
17. Fernandez-Recio,J., Abagyan,R. and Totrov,M. (2005) Improving CAPRI predictions: optimized desolvation for rigid-body docking. *Proteins*, **60**, 308–313.
18. Grosdidier,S., Pons,C., Solernou,A. and Fernandez-Recio,J. (2007) Prediction and scoring of docking poses with pyDock. *Proteins*, **69**, 852–858.

19. Pons,C., Solernou,A., Perez-Cano,L., Grosdidier,S. and Fernandez-Recio,J. (2010) Optimization of pyDock for the new CAPRI challenges: docking of homology-based models, domain-domain assembly and protein-RNA binding. *Proteins*, **78**, 3182–3188.
20. Dominguez,C., Boelens,R. and Bonvin,A.M. (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.*, **125**, 1731–1737.
21. Mendez,R., Leplae,R., De Maria,L. and Wodak,S.J. (2003) Assessment of blind predictions of protein-protein interactions: current status of docking methods. *Proteins*, **52**, 51–67.
22. Mendez,R., Leplae,R., Lensink,M.F. and Wodak,S.J. (2005) Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. *Proteins*, **60**, 150–169.
23. Lensink,M.F., Mendez,R. and Wodak,S.J. (2007) Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins*, **69**, 704–718.
24. Lensink,M.F. and Wodak,S.J. (2010) Docking and scoring protein interactions: CAPRI 2009. *Proteins*, **78**, 3073–3084.
25. Lensink,M.F. and Wodak,S.J. (2013) Docking, scoring, and affinity prediction in CAPRI. *Proteins*, **81**, 2082–2095.
26. Chelliah,V., Blundell,T.L. and Fernandez-Recio,J. (2006) Efficient restraints for protein-protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment. *J. Mol. Biol.*, **357**, 1669–1682.
27. Sondermann,H., Nagar,B., Bar-Sagi,D. and Kuriyan,J. (2005) Computational docking and solution x-ray scattering predict a membrane-interacting role for the histone domain of the Ras activator son of sevenless. *Proc. Natl. Acad. Sci. U.S.A.*, **102**, 16632–16637.
28. Bernado,P., Perez,Y., Blobel,J., Fernandez-Recio,J., Svergun,D.I. and Pons,M. (2009) Structural characterization of unphosphorylated STAT5a oligomerization equilibrium in solution by small-angle X-ray scattering. *Protein Sci.*, **18**, 716–726.
29. Ribeiro Ede,A. Jr, Leyrat,C., Gerard,F.C., Albertini,A.A., Falk,C., Ruigrok,R.W. and Jamin,M. (2009) Binding of rabies virus polymerase cofactor to recombinant circular nucleoprotein-RNA complexes. *J. Mol. Biol.*, **394**, 558–575.
30. Fislage,M., Brosens,E., Deyaert,E., Spilotros,A., Pardon,E., Loris,R., Steyaert,J., Garcia-Pino,A. and Versees,W. (2014) SAXS analysis of the tRNA-modifying enzyme complex MnmE/MnmG reveals a novel interaction mode and GTP-induced oligomerization. *Nucleic Acids Res.*, **42**, 5978–5992.
31. Pons,C., D'Abramo,M., Svergun,D.I., Orozco,M., Bernado,P. and Fernandez-Recio,J. (2010) Structural characterization of protein-protein complexes by integrating computational docking with small-angle scattering data. *J. Mol. Biol.*, **403**, 217–230.
32. Schneidman-Duhovny,D., Hammel,M. and Sali,A. (2011) Macromolecular docking restrained by a small angle X-ray scattering profile. *J. Struct. Biol.*, **173**, 461–471.
33. Karaca,E. and Bonvin,A.M. (2013) On the usefulness of ion-mobility mass spectrometry and SAXS data in scoring docking decoys. *Acta Crystallogr. D Biol. Crystallogr.*, **69**, 683–694.
34. Svergun,D.I., Barberato,C. and Koch,M.H.J. (1995) CRYSOL—a program to evaluate X-ray solution scattering of biological macromolecules from atomic coordinates. *J. Appl. Crystallogr.*, **28**, 768–773.
35. Hwang,H., Vreven,T., Janin,J. and Weng,Z. (2010) Protein-protein docking benchmark version 4.0. *Proteins*, **78**, 3111–3114.
36. Shin,D.S., Didonato,M., Barondeau,D.P., Hura,G.L., Hitomi,C., Berglund,J.A., Getzoff,E.D., Cary,S.C. and Tainer,J.A. (2009) Superoxide dismutase from the eukaryotic thermophile Alvinella pompejana: structures, stability, mechanism, and insights into amyotrophic lateral sclerosis. *J. Mol. Biol.*, **385**, 1534–1555.
37. Hura,G.L., Menon,A.L., Hammel,M., Rambo,R.P., Poole,F.L. II, Tsutakawa,S.E., Jenney,F.E. Jr, Classen,S., Frankel,K.A., Hopkins,R.C. *et al.* (2009) Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS). *Nat. Methods*, **6**, 606–612.
38. Xu,X., Reinle,W., Hannemann,F., Konarev,P.V., Svergun,D.I., Bernhardt,R. and Ubbink,M. (2008) Dynamics in a pure encounter complex of two proteins studied by solution scattering and paramagnetic NMR spectroscopy. *J. Am. Chem. Soc.*, **130**, 6395–6403.
39. Valentini,E., Kikhney,A.G., Previtali,G., Jeffries,C.M. and Svergun,D.I. (2015) SASBDB, a repository for biological small-angle scattering data. *Nucleic Acids Res.*, **43**, D357–D363.
40. Bernstein,F.C., Koetzle,T.F., Williams,G.J., Meyer,E.F. Jr, Brice,M.D., Rodgers,J.R., Kennard,O., Shimanouchi,T. and Tasumi,M. (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, **112**, 535–542.
41. Canutescu,A.A., Shelenkov,A.A. and Dunbrack,R.L. Jr (2003) A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.*, **12**, 2001–2014.
42. Jimenez-Garcia,B., Pons,C. and Fernandez-Recio,J. (2013) pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring. *Bioinformatics*, **29**, 1698–1699.

3.2 Validation and current challenges of protein-protein docking methods

The growing interest in protein-protein interactions and the technical advances in the computational field have fostered the number of *in silico* tools developed in the past years. With the aim of modeling protein complexes starting from the isolated component structures, testing and comparing these computational methodologies have become fundamental in order to assess their performance, identify their limitations, and encourage new developments in the field. In this context, community-wide experiments such as CAPRI provide a common ground for testing the predictive capability of currently available docking methods.

First, the performance of our pyDock protocol (Cheng et al., 2007) on the last CAPRI round (Lensink and Wodak, 2013) will be evaluated and discussed. Second, the participation of our group in the special CAPRI-CASP experiment will be presented. Third, a suitable set of protein-RNA complex structures has been compiled in order to establish a common framework for the evaluation of different protein-RNA interaction predicting methods. The last manuscript describes an update of the protein-protein docking benchmark, including new affinity data, which has been applied to evaluate the predictive performance of our docking tools.

Manuscripts presented in this section:

Chiara Pallara, Brian Jiménez-García, Laura Pérez-Cano, Miguel Romero-Durana, Albert Solernou, Solène Grosdidier, Carles Pons, Iain H. Moal and Juan Fernandez-Recio (2013) **"Expanding the Frontiers of Protein-Protein Modeling: From Docking and Scoring to Binding Affinity Predictions and Other Challenges."** Proteins 81 (12): 2192–2200.

Laura Pérez-Cano, Brian Jiménez-García and Juan Fernández-Recio (2012) **"A Protein-RNA Docking Benchmark (II): Extended Set from Experimental and Homology Modeling Data."** Proteins 80 (7): 1872–82.

Thom Vreven, Iain H. Moal, Anna Vangone, Brian G. Pierce, Panagiotis L. Kastritis, Mieczyslaw Torchala, Raphael Chaleil, Brian Jiménez-García, Paul A. Bates, Juan Fernández-Recio, Alexander M. Bonvin, Zhiping Weng (2015) **"Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2."** Journal of Molecular Biology 427 (19): 3031–41.

3.2.1. Expanding the Frontiers of Protein-Protein Modeling: From Docking and Scoring to Binding Affinity Predictions and Other Challenges.

Chiara Pallara[1#], Brian Jiménez-García[1#], Laura Pérez-Cano[1], Miguel Romero-Durana[1], Albert Solernou[1], Solène Grosdidier[1], Carles Pons[1], Iain H. Moal[1] and Juan Fernandez-Recio[1*]

[1]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain*

*Corresponding author

[#]Equal contribution

PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Expanding the frontiers of protein–protein modeling: From docking and scoring to binding affinity predictions and other challenges

Chiara Pallara,[1] Brian Jiménez-García,[1] Laura Pérez-Cano,[1] Miguel Romero-Durana,[1] Albert Solernou,[1] Solène Grosdidier,[1] Carles Pons,[1,2] Iain H. Moal,[1] and Juan Fernandez-Recio[1]*

[1] Joint BSC-IRB Research Programme in Computational Biology, Barcelona Supercomputing Center, Barcelona, Spain
[2] Computational Bioinformatics, National Institute of Bioinformatics (INB), Barcelona, Spain

ABSTRACT

In addition to protein–protein docking, this CAPRI edition included new challenges, like protein–water and protein–sugar interactions, or the prediction of binding affinities and $\Delta\Delta G$ changes upon mutation. Regarding the standard protein–protein docking cases, our approach, mostly based on the *pyDock* scheme, submitted correct models as predictors and as scorers for 67% and 57% of the evaluated targets, respectively. In this edition, available information on known interface residues hardly made any difference for our predictions. In one of the targets, the inclusion of available experimental small-angle X-ray scattering (SAXS) data using our *pyDockSAXS* approach slightly improved the predictions. In addition to the standard protein–protein docking assessment, new challenges were proposed. One of the new problems was predicting the position of the interface water molecules, for which we submitted models with 20% and 43% of the water-mediated native contacts predicted as predictors and scorers, respectively. Another new problem was the prediction of protein–carbohydrate binding, where our submitted model was very close to being acceptable. A set of targets were related to the prediction of binding affinities, in which our pyDock scheme was able to discriminate between natural and designed complexes with area under the curve = 83%. It was also proposed to estimate the effect of point mutations on binding affinity. Our approach, based on machine learning methods, showed high rates of correctly classified mutations for all cases. The overall results were highly rewarding, and show that the field is ready to move forward and face new interesting challenges in interactomics.

Proteins 2013; 81:2192–2200.

Key words: complex structure; CAPRI; protein–protein docking; pyDock; protein–carbohydrate interactions.

INTRODUCTION

One of the major challenges in structural biology is to provide structural data for all complexes formed between proteins and other macromolecules. Current structural coverage of protein–protein interactions (i.e., available experimental structures plus potential models based on homologous complex structures) is below 4% of the estimated number of possible complexes formed between human proteins.[1,2] The pace of experimental determination of complex structures is still behind the determination of individual protein structures. In addition, many of these interactions will never be determined by X-ray crystallography because of their transient nature. For these reasons, computational docking methods aim to become a complementary approach to solve the structural interactome. The field of protein docking has experienced an explosion in recent years, partially propelled by the CAPRI experiment. Past editions showed an increasing amount of participant groups and computational approaches, and a

Grant sponsor: Spanish Ministry of Science; Grant number: BIO2010-22324; Grant sponsor: FPU (to LP-C) and FPI (to BJ-G) fellowships from the Spanish Ministry of Science; Grant sponsor: People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA (to IHM); Grant number: PIEF-GA-2012-327899.
Pallara and Jiménez-García contributed equally to this work.
*Correspondence to: Juan Fernandez-Recio, Barcelona Supercomputing Center (BSC), Jordi Girona 29, 08034 Barcelona, Spain. E-mail: juanf@bsc.es
Received 14 June 2013; Revised 31 July 2013; Accepted 1 August 2013
Published online 12 August 2013 in Wiley Online Library (wileyonlinelibrary.com).
DOI: 10.1002/prot.24387

large variety of targets. We have participated in all targets of this fifth CAPRI edition. In addition to the standard prediction of protein–protein targets, this edition has entered into related areas including binding affinity predictions and free energy changes upon mutation, as well as prediction of sugar binding and interface water molecules. Our overall experience has been highly rewarding and we describe here the details of our participation and the key factors of our success.

MATERIALS AND METHODS

Generation of rigid-body docking poses for the predicting experiment

In all targets, we used FTDock[3] with electrostatics and 0.7 Å grid resolution and ZDOCK 2.1[4] to generate 10,000 and 2000 rigid-body docking poses, respectively, as previously described.[5] For the final four targets of this edition (T53, T54, T57, and T58) we generated an additional pool of flexible docking poses using SwarmDock. For these runs, the standard protocol was employed,[6–8] with the Dcomplex score used as the objective function,[9] but without the final clustering and rescoring phase. In T46 we generated an additional pool of 10,000 solutions using FTDock without electrostatics and at lower resolution (1.2 Å), as part of an old protocol used with previous targets, but these conditions were not applied for the rest of the targets since we saw in the past that this step was not increasing the chances of correct predictions. In T46 and T47, we used RotBUS[10] to generate 59,112 and 41,021 additional docking poses, respectively, but this method was not used for the rest of the targets since we previously checked that this procedure did not improve the results. In Target T50, given the large size of 1918 H1N1 influenza virus hemagglutinin protein, we generated a total of 92,432 FTDock docking poses, increasing the number of translations selected from each rotation from 3 (default) to 10. Cofactors, water molecules and solvent ions were not included in our docking calculations.

Scoring of rigid-body docking poses for both the predicting and the scoring experiments

We scored the docking models generated by the above described methods with our pyDock protocol,[11] based on energy terms previously optimized for rigid-body docking. The binding energy is basically composed of accessible surface area-based desolvation, Coulombic electrostatics and van der Waals energy (with a weighting factor of 0.1 to reduce the noise of the scoring function). Electrostatics and van der Waals were limited to ± 1.0 and 1.0 kcal/mol for each interatomic energy value, respectively, to avoid excessive penalization from possible clashes in the structures generated by the rigid-body approach. The same protocol was used in the scoring experiment to score all the docking models that were proposed. We did not include van der Waals in the T46 scoring experiment, although this did not affect the results. Cofactors, water molecules and solvent ions were not considered for scoring.

Removal of redundant docking poses

After scoring, we eliminated redundant predictions to increase the variability of the predictions and maximize the success chances using a simple clustering algorithm with a distance cutoff of 4.0 Å, as previously described.[12] In target T47, since the resulting solutions looked correct [according to the available structure of a highly homologous complex with protein data bank (PDB) code 2WPT], we reduced this cutoff to 0.5 Å.

Minimization of final models

The final 10 selected docking poses were minimized to improve the quality of the docking models and reduce the number of interatomic clashes. In the majority of the targets we used TINKER[13] as previously described.[12,14] In targets T53 and T54 we used CHARMM (50 steps conjugate gradient, 500 steps adopted-basis Newton–Raphson and 50 steps steepest decent, with the CHARMM19 force field).[15] In target T58 we used AMBER10 with AMBER parm99 force field.[16,17] The minimization protocol consisted of a 500-cycle steepest descent minimization with harmonic restraints applied at a force constant of 25 kcal/(mol·Å^2) to all the backbone atoms to optimize the side chains, followed by another 500-cycle conjugate gradient minimization without restraints. This minimization step was performed after ranking, solely to remove clashes.

Modeling of subunits with no available structure

For several targets, the structures of the subunits were not available and needed to be modeled. In most of the targets, we used Modeler 9v6 with default parameters[18] based on the template/s suggested by the organizers or on other homologue proteins found by BLAST[19] search. The final selected model was that with the lowest DOPE score.[20] For targets T53 and T54 we used POPULUS (http://bmm.cancerresearchuk.org/~populus/) with default template selection and model building settings.[21]

RESULTS AND DISCUSSION

In this CAPRI edition we submitted predictions for all the proposed targets. Our results for the standard protein–protein docking assessment are summarized in Table I. In addition, there were new challenges like the

Table I
Results of Our pyDock Protocol for All Protein–Protein Targets of the Last CAPRI Edition

		Predictors			Scorers		
Target	Type	Submission rank[a]	Quality[b]	Successful groups[c]	Submission rank[a]	Quality[b]	Successful groups[c]
T46	HH	—	—	2 (40)	—	—	8 (16)
T47	HU	1	***	25 (29)	2[d]	***	13 (14)
T48	UU	3	*	14 (32)	No scorers	No scorers	No scorers
T49	UU	4	*	14 (33)	6	*	7 (13)
T50	UH	1	**	18 (40)	4	**	12 (17)
T51	DHD	—	—	3 (46)	—	—	5 (13)
T53	UH	3[e]	**	20 (42)	1	**	11 (13)
T54	UH	—	—	4 (41)	—	—	0 (13)
T58	UU	5	**	11 (23)	No scorers	No scorers	No scorers

U, unbound; H, homology-based model; D, domain.
[a]Rank of the best model within our submission to CAPRI.
[b]Quality of our best model according to CAPRI criteria.
[c]Number of successful groups for each target; in brackets, total number of participants.
[d]Model Rank 1 had medium accuracy (**).
[e]Model Rank 1 had acceptable accuracy (*).

prediction of protein–water and protein–sugar interactions, as well as the estimation of binding affinities and energy changes upon mutation. Hereinafter, we describe in detail our submissions for each of the targets.

Standard protein–protein docking assessment: successful predictions

Target T47 (model/pseudounbound)

Target T47 was the structural prediction of the complex between the DNase domain of colicin E2 and the immunity protein Im2. The real challenge in this target was the prediction of interface water molecules, however, the protein–protein docking predictions were already assessed, and therefore we have included them in this section. The colicin E2 was modeled based on the structure of colicin E9 (85% sequence identity) in complex with Im9 immunity protein (PDB 1EMV).[22] The coordinates of the immunity protein Im2 were extracted from its structure in complex with colicin E9 (PDB 2WPT). Given the existence of this homologous colicin E9/Im2 complex structure (PDB 2WPT),[23] the binding mode for target T47 was easy to determine by template-based docking. However, we performed the template-free docking calculations to assess the automatic docking protocol. We only applied distance restraints after pyDock protocol by selecting those docking poses in which two key contacting residues, Im2 Y54 and colicin E2 F85 (equivalent to colicin E9 F86 in 2WPT),[23] were within an arbitrary distance of 6 Å (same distance used in standard restraints with pyDockRST module.[24] We submitted five correct models (one high accuracy, one medium accuracy, and three acceptable). Our first submitted model (Rank 1 according to pyDock energy, and generated by ZDOCK), was a high-quality model (Table I), with 75% native contacts, 2.48 Å ligand root mean square deviation (RMSD), and 0.75 Å interface RMSD with respect to the crystal structure (Fig. 1; PDB 3U43).[25] This docking model had the lowest ligand RMSD with respect to the homologous colicin E9/Im2 complex (PDB 2WPT) amongst all solutions (although we did not use this homologous structure for docking), and even more interestingly, we would have obtained exactly the same result without applying the above-mentioned distance restraint filter.

For the scoring experiment, we evaluated the provided 1051 models with our pyDock scoring function, and applied the same distance filter that we used as predictors (see above). All our submitted predictions resulted to be successful, consisting of six medium and four high-quality models. We had a high-accuracy model ranked second after pyDock scoring and distance filter (uploaded by Weng), with 77% native contacts, 0.9 Å ligand RMSD, and 0.4 Å interface RMSD with respect to the crystal structure (PDB 3U43[25]; Table I; Fig. 1). Interestingly, our Rank 5 model was the best model submitted among all 14 participants, with 79% native contacts, 0.7 Å ligand RMSD, and 0.5 Å interface RMSD. Two better models uploaded by Weng were not found by any of the participants. Remarkably, as in predictors, our results would not have changed had we not applied the distance restraints filter.

Target T48 (unbound/unbound)

Target T48 was the structural prediction of the complex between the diiron-hydroxylase toluene 4-monooxygenase and the Rieske-type ferredoxin T4moC protein (PDB 1VM9).[26] As suggested by the organizers, the heterohexameric biological unit of the diiron-hydroxylase was built by applying crystal symmetry operations to its trimeric structure in complex with the

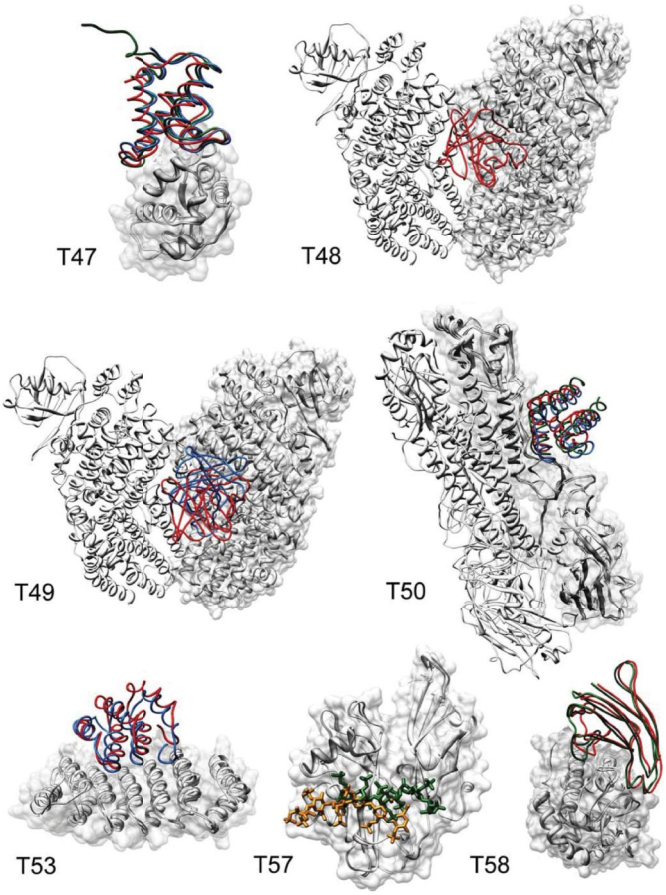


Figure 1

Representation of our best models for targets T47, T48, T49, T50, T53, T57, and T58. For each target, receptors are superimposed and shown in white. Ligand in our best model as predictors is shown in red, and as scorers in blue. For comparison, the structure of the experimental complex (if available) is represented in green.

T4moD effector protein (PDB 3DHH).[27] We used the hexameric construct for the generation of docking poses, which were scored by pyDock. Then, we selected those docking poses that had any of the diiron-hydroxylase Fe^{2+} and ferredoxin S_2Fe_2 atoms within 23 Å distance to allow for the electron transfer between these groups[27] (the distance cutoff we used was arbitrary, based on the expected distance of 16 Å in 3DHH plus a margin to allow the inclusion of some low-energy solutions). For the submission, we removed chains D, E, and F from the hexamer as we misinterpreted some of the organizers' instructions, but this did not affect the quality of the submitted models. The analysis of the results showed that we submitted three models of acceptable quality. Our prediction ranked third after pyDock scoring and electron transfer distance filtering (generated by FTDock)

had 14% native contacts, 8.4 Å ligand RMSD, and 3.6 Å interface RMSD with respect to the complex crystal structure (not yet available). We found another acceptable model (ranked 10th in our submission set) that had 49% native contacts, 6.3 Å ligand RMSD, and 2.2 Å interface RMSD with respect to the complex crystal structure.

Target 49 (unbound/unbound)

Target T49 was the same complex as T48 but with a different hexameric conformation for diiron-hydroxylase toluene 4-monooxygenase (unbound coordinates not released). We applied the same protocol as for target T48 (pyDock scoring and electron transfer distance filtering). We submitted four acceptable quality models. The model ranked fourth of our submission set had acceptable quality, with 26% native contacts, 12.4 Å ligand RMSD, and 3.5 Å interface RMSD with respect to the complex crystal structure (not yet available). We also submitted another model with 11% native contacts, 6.9 Å ligand RMSD, and 2.7 Å interface RMSD.

For the scoring experiment, the 1085 solutions were scored by the same protocol, based on pyDock scoring and electron transfer distance filtering. In some models, the monooxygenase was uploaded as a trimer, therefore we reconstructed the biological hexamer (based on symmetry) to calculate the electron transfer distance filter. Since it was not clear whether in these cases the hexamer was going to be rebuilt for the assessment, our submission set was formed by the top five solutions obtained after rebuilding the hexamer, and by the top five solutions obtained by just considering the structure submitted by uploaders (i.e., without rebuilding the hexamer in cases of uploaded trimer). Our ranked sixth submission was an acceptable model (uploaded by Nakamura), with 11% native contacts, 7.9 Å ligand RMSD, and 2.9Å interface RMSD with respect to the complex crystal structure (not yet available).

Target 50 (unbound/model)

Target T50 was the structural prediction of the complex between the 1918 H1N1 influenza virus hemagglutinin and the HB36.3 *de novo* designed protein. The coordinates of the hemagglutinin were taken from its structure in complex with an antibody (PDB 3GBN)[28] and the biological hexamer was rebuilt by applying symmetry operations. We modeled the HB36.3 based on the crystal structure of the homologous (83% sequence identity) protein APC36109 from *Bacillus stearothermophilus* (PDB 1U84), using the target-template protein alignment offered by the organizers. Given the size of the system, we increased the number of rigid-body docking solutions generated by FTDock (see Materials and Methods section). Our submission as predictors contained nine successful models (five acceptable and four medium-quality solutions). Our Rank 1 submission (generated by FTDock) was a medium-quality model with 47% native contacts, 6.1 Å ligand RMSD, and 1.8 Å interface RMSD with respect to the complex crystal structure (Fig. 1; PDB 3R2X).[29] Interestingly, our Rank 4 submission, with 41% native contacts, 3.4 Å ligand RMSD, and 1.6 Å interface RMSD, was the best model submitted among all participants as predictors.

For the scoring experiment, we evaluated the 1451 models in the same conditions as in predictors. We found five acceptable and one medium-quality solutions. Our Rank 4 submission was a medium-quality model, with 44.9% native contacts, 4.71 Å ligand RMSD, and 1.93 Å interface RMSD with respect to the complex crystal structure (PDB 3R2X[29]; Fig. 1).

Target T53 (unbound/model)

Target T53 was a complex between two artificial alpha helicoidal repeat proteins (alpha-Rep), alpha-rep4 (PDB 3LTJ)[30] and alpha-rep2, both designed on the basis of thermostable HEAT-like repeats. The ligand alpha-rep2 was built using as template alpha-rep4 (PDB 3LTJ), with 77% sequence identity. All the docking poses, generated using Zdock, Ftdock, and SwarmDock, were scored by pyDock. We submitted four successful predictions (three acceptable and one medium-quality models). Our Rank 3 submission, a medium accuracy model generated by SwarmDock, had 44% native contacts, ligand RMSD 4.4 Å, and interface RMSD 1.8 Å with respect to the crystal structure (not yet available).

For the scoring experiment, we evaluated 1400 alpha-rep4/alpha-rep2 complex models applying the same protocol as in predictors in a completely automated fashion. We found three acceptable and a medium-quality models. Our Rank 1 submission, a medium-quality model (uploaded by Yan Shen), had 62% native contacts, 3.6 Å ligand RMSD, and 1.3 Å interface RMSD with respect to the complex crystal structure (not yet available).

Target T58 (unbound/unbound)

This target was a complex between the unbound G-Type Lysozyme (PDB 3MGW)[31] and the unbound *Escherichia coli* Plig lysozyme inhibitor (PDB 4DY3).[32] There was available small-angle X-ray scattering (SAXS) data for this complex, which we used for scoring with our module pyDockSAXS, previously developed to combine pyDock scoring and fitting to SAXS data.[33] In addition, there was some available information indicating a central role of the G-type lysozyme E73, D86, and D97 residues and the *E. coli* Plig lysozyme inhibitor R119 and Y47 residues.[34] Based on these residues, we imposed ambiguous distance restraints with our module pyDockRST.[35] We submitted one medium-accuracy and two acceptable models. Our Rank 5 model, generated by SwarmDock, was a medium-quality model, and resulted

to be the fourth best model submitted among all the 23 participants, with 43% native contacts, 4.9 Å ligand RMSD, and 1.8 Å interface RMSD with respect to the complex crystal structure (PDB 4G9S).[36] Interestingly, although the distance restraints proved to be essential for this target, we would have obtained only slightly worse results without using the SAXS data (Rank 10 medium accuracy model). This is probably due to the shape of the complex, classified as spherical according to the anisotropy value (1.4) computed from the ratio between the size of the largest axis and the smallest ones. Indeed, we previously showed that SAXS data does not provide much beneficial information in this type of cases.[33]

Protein–protein docking: unsuccessful cases

In three of the protein–protein cases (T46, T51, and T54) we were not able to submit any correct model, either as predictors or as scorers. These cases seemed to be highly difficult for the majority of participants, since in all of them there were no more than three successful groups as predictors or as scorers or both (Table I). In target T46 (*model/model*), the interacting subunits Mtq2 and Trm112 were modeled based on the homologue templates with low sequence identity (Mtq2 was based on template with PDB code 1T43, 28% sequence identity; Trm112 was based on template with PDB code 2J6A, 36% sequence identity). The inaccuracies in the modeling added too much error and the docking was not successful. Target T51 (*bound/model/unbound*) was a difficult case of a multidomain protein, with interactions between GH5-CBM6/CBM13/Fn3 domains. This could be divided in two different docking cases both involving CBM13 domain, which needed to be modeled based on template with PDB code 1KNL (38% sequence identity). Again, a model based on a template with that level of homology can deteriorate docking results. Target 54 (*unbound/model*) was in principle easy, involving the modeling of Rep16 based on the template with PDB code 3LTJ (88% sequence identity), but the submitted solutions were incorrect for us as well as for the majority of participants. Indeed, despite the scoring set contained several acceptable models, no group was able to identify them (Table I).

Prediction of protein–water interactions

Target T47 was the prediction of a protein–protein complex structure, as described in above sections, but the real challenge was to predict the location of water molecules. After generating the protein–protein docking poses as above described, we predicted the water positions in each docking model using DOWSER[37] with default parameters (with a probe radius of 0.2Å and the default atoms dictionary). Our Rank 1 submitted model (generated by ZDOCK) had 20% of water native contacts, and was classified as fair (+). If we consider only the prediction of the buried water molecules, our success rates do not significantly change.

For the scoring experiment, we just applied our standard pyDock scoring function, plus distance restraints as described in above sections. The water molecules proposed in the different docking poses were not included in the scoring. Our Rank 8 submitted model (uploaded by Bates) had 43% of water native contacts and was classified as good (++). More details can be found in an upcoming publication.

Prediction of protein–carbohydrate complex structure

Target 57 (*unbound/model*) was a challenging target consisting in the prediction of the interaction between BT4661 protein and heparin. The structure of heparin in the complex was not known, so we modeled it using molecular dynamics starting with the provided conformation. We ran 10 ns using the force field AMBER parm99 of the Amber10 package[16,17] and extracted 1000 representative snapshots. Since our pyDock protocol was not intended for protein–sugar interactions, we had to devise a new *ad hoc* docking procedure. For that, we used FTDock to dock each of the 1000 heparin conformations to BT4661 protein. We selected the top 10,000 docking poses as scored by FTDock (no electrostatics). Then we applied different scoring functions to this set of docking poses: (i) PScore without minimization; (ii) PScore with minimization; and (iii) AMBER after minimization. We selected the 1000 best-scoring solutions from each method and finally we removed redundant solutions within 6.5 Å ligand RMSD. No correct submission was submitted. However, our Rank 4 submission was almost acceptable, with 65% native contacts, 11.2 Å ligand RMSD, and 4.3 Å interface RMSD with respect to the complex crystal structure (PDB 4AK2; Fig. 1). We checked *a posteriori* that there were several correct models within our docking sets, but our scoring approach failed to place them in the lowest scoring positions.

Other challenges: binding affinity and $\Delta\Delta G$ predictions

This CAPRI edition also involved the challenging problem of predicting binding affinities and energy changes upon mutations. Round 21 was the discrimination between 87 designed protein–protein interactions involving three proteins of interest (Spanish influenza HA; *Mt* ACP-2; Fc region of human IgG1) and 120 naturally occurring complexes. The pyDock function, although initially developed for the scoring of docking poses, was previously shown to have some correlation with the binding affinity data collected by Kastritis and Bonvin.[38] This was later confirmed on a subset of

complexes with high-confidence affinity data, where pyDock ranked among the best performing scoring functions with a correlation of 0.63.[39] For round 21 predictions, we evaluated the correlation of each of the different pyDock individual terms with the binding affinities on the provided set of 120 naturally occurring complexes. We found that desolvation correlation with binding affinity data was not clear, showing even negative correlation with data obtained by ITC experiments. It seems that, although desolvation is essential for rigid-body docking (perhaps to compensate inaccurate calculation of electrostatics and van der Waals), it is not the most important factor for binding affinity predictions from the complex structure (in which electrostatics can be more accurately calculated). Based on these results, we devised a binding affinity descriptor (*pyDockAFF* = electrostatics $-1.0 \times$ desolvation), with confidence thresholds for the discrimination of complexes according to their binding affinities. Our predictions had area under the curve 83%, with good discrimination between designed and native interfaces. More details can be found in a recent publication.[40] It remains to be seen whether the *pyDockAFF* binding affinity predictor is suitable only for the cases in this CAPRI round, or it has more general applicability (further details in an upcoming publication).

Targets T55-56 aimed to predict the binding affinity changes upon mutations on two designed influenza hemagglutinin protein binders. We applied a multiparametric predictive model with 85 descriptors using an ensemble of models which were combined to produce consensus predictions. The models were trained upon a data set of 930 changes in affinity upon mutation which were taken from the literature. Due to the fairly low cases to descriptors ratio (10.9), we preferentially employed models with inherent overfitting avoidance bias, such as prepruning or feature selection using the Akaike information criterion, methods which construct multiple models using subsets of the descriptors and the training data, and by rejecting learners that performed poorly using leave-complex-out cross-validation.[41] To further avoid overfitting, we did not combine the selected learners together using stacking, instead opting for the unweighted mean for our consensus predictions. This approach provided an excellent ability to predict the effect of mutation, more details of which can be found in a recent publication.[42] We have since expanded this data set to form the SKEMPI database, which now includes 3047 $\Delta\Delta G$ values, as well as kinetic and thermodynamic data,[43] and have used the data to derive contact potentials that can circumvent some of the approximations associated with statistical potentials.[44]

CONCLUSIONS

We have continued our participation in CAPRI with pyDock, submitting models for all the predicting, scoring, and binding affinity prediction experiments. For the generation of docking poses, the better grid resolution used for FTDock and the use of flexible SwarmDock for the last targets were key for the success. This produced docking poses of sufficient quality to be identified by the pyDockSER scoring scheme. In selected targets, distance restraints were introduced by pyDockRST, but in most cases this did not make a difference. In one target, SAXS data was used for complementary scoring with pyDockSAXS, which slightly improved the scoring. We obtained consistently good models for all nondifficult cases, although they were far from being trivial, since their subunits were unbound or needed to be modeled based on homology templates. In all cases but one our successful models were ranked within our first five submitted solutions, being ranked first in several cases. In this CAPRI edition we learned that our automated protocol is useful to provide correct models in easy-to-medium difficulty protein–protein docking cases, but we need further methodological development for difficult cases, especially when subunits need to be modeled based on homologues with low sequence identity. On the other side, interface water placement and sugar-binding proved to be highly challenging for our protein–protein methodology, but the results have encouraged us to develop new methods for these problems. Finally, prediction of binding affinity based on the pyDockSER scoring, and energy changes upon mutation based on multiparametric regression models showed excellent results. The overall experience has been highly rewarding and has shown once again the importance of community-wide assessment of prediction methods.

REFERENCES

1. Venkatesan K, Rual JF, Vazquez A, Stelzl U, Lemmens I, Hirozane-Kishikawa T, Hao T, Zenkner M, Xin X, Goh KI, Yildirim MA, Simonis N, Heinzmann K, Gebreab F, Sahalie JM, Cevik S, Simon C, de Smet AS, Dann E, Smolyar A, Vinayagam A, Yu H, Szeto D, Borick H, Dricot A, Klitgord N, Murray RR, Lin C, Lalowski M, Timm J, Rau K, Boone C, Braun P, Cusick ME, Roth FP, Hill DE, Tavernier J, Wanker EE, Barabasi AL, Vidal M. An empirical framework for binary interactome mapping. Nat Methods 2009;6:83-90.
2. Stumpf MP, Thorne T, de Silva E, Stewart R, An HJ, Lappe M, Wiuf C. Estimating the size of the human interactome. Proc Natl Acad Sci USA 2008;105:6959-3964.
3. Gabb HA, Jackson RM, Sternberg MJ. Modelling protein docking using shape complementarity, electrostatics and biochemical information. J Mol Biol 1997;272:106-120.
4. Chen R, Weng Z. A novel shape complementarity scoring function for protein–protein docking. Proteins 2003;51:397-408.
5. Grosdidier S, Pons C, Solernou A, Fernandez-Recio J. Prediction and scoring of docking poses with pyDock. Proteins 2007;69:852-858.
6. Moal IH, Bates PA. SwarmDock and the use of normal modes in protein–protein docking. Int J Mol Sci 2010;11:3623-3648.
7. Torchala M, Moal IH, Chaleil RA, Fernandez-Recio J, Bates PA. SwarmDock: a server for flexible protein–protein docking. Bioinformatics 2013;29:807-809.
8. Li X, Moal IH, Bates PA. Detection and refinement of encounter complexes for protein–protein docking: taking account of macromolecular crowding. Proteins 2010;78:3189-3196.

9. Liu S, Zhang C, Zhou H, Zhou Y. A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. Proteins 2004;56:93-101.
10. Solernou A, Fernandez-Recio J. Protein docking by Rotation-Based Uniform Sampling (RotBUS) with fast computing of intermolecular contact distance and residue desolvation. BMC Bioinformatics 2010; 11:352.
11. Cheng TM, Blundell TL, Fernandez-Recio J. pyDock: electrostatics and desolvation for effective scoring of rigid-body protein–protein docking. Proteins 2007;68:503-515.
12. Pons C, Solernou A, Perez-Cano L, Grosdidier S, Fernandez-Recio J. Optimization of pyDock for the new CAPRI challenges: docking of homology-based models, domain–domain assembly and protein–RNA binding. Proteins 2010;78:3182-3188.
13. Ponder JW, Richards FM. An efficient Newton-like method for molecular mechanics energy minimization of large molecules. J Comput Chem 1987;8:1016-1024.
14. Pons C, Grosdidier S, Solernou A, Perez-Cano L, Fernandez-Recio J. Present and future challenges and limitations in protein–protein docking. Proteins 2010;78:95-108.
15. Brooks BR, Brooks CL, 3rd, Mackerell AD, Jr., Nilsson L, Petrella RJ, Roux B, Won Y, Archontis G, Bartels C, Boresch S, Caflisch A, Caves L, Cui Q, Dinner AR, Feig M, Fischer S, Gao J, Hodoscek M, Im W, Kuczera K, Lazaridis T, Ma J, Ovchinnikov V, Paci E, Pastor RW, Post CB, Pu JZ, Schaefer M, Tidor B, Venable RM, Woodcock HL, Wu X, Yang W, York DM, Karplus M. CHARMM: the biomolecular simulation program. J Comput Chem 2009;30:1545-1614.
16. Case DA, Cheatham TE, III, Darden T, Gohlke H, Luo R, Merz KM, Jr., Onufriev A, Simmerling C, Wang B, Woods RJ. The Amber biomolecular simulation programs. J Comput Chem 2005;26:1668-1688.
17. Wang J, Cieplak P, Kollman PA. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? J Comput Chem 2000; 21:1049-1074.
18. Sali A, Blundell TL. Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 1993;234:779-815.
19. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403-410.
20. Shen MY, Sali A. Statistical potential for assessment and prediction of protein structures. Protein Sci 2006;15:2507-2524.
21. Offman MN, Fitzjohn PW, Bates PA. Developing a move-set for protein model refinement. Bioinformatics 2006;22:1838-1845.
22. Kuhlmann UC, Pommer AJ, Moore GR, James R, Kleanthous C. Specificity in protein–protein interactions: the structural basis for dual recognition in endonuclease colicin-immunity protein complexes. J Mol Biol 2000;301:1163–1178.
23. Meenan NA, Sharma A, Fleishman SJ, Macdonald CJ, Morel B, Boetzel R, Moore GR, Baker D, Kleanthous C. The structural and energetic basis for high selectivity in a high-affinity protein–protein interaction. Proc Natl Acad Sci USA 2010;107:10080-10085.
24. Cheng TM, Blundell TL, Fernandez-Recio J. Structural assembly of two-domain proteins by rigid-body docking. BMC Bioinformatics 2008;9:441.
25. Wojdyla JA, Fleishman SJ, Baker D, Kleanthous C. Structure of the ultra-high-affinity colicin E2 DNase–Im2 complex. J Mol Biol 2012; 417:79-94.
26. Moe LA, Bingman CA, Wesenberg GE, Phillips GN, Jr., Fox BG. Structure of T4moC, the Rieske-type ferredoxin component of toluene 4-monooxygenase. Acta Crystallogr D Biol Crystallogr 2006;62: 476-482.
27. Bailey LJ, McCoy JG, Phillips GN, Jr, Fox BG. Structural consequences of effector protein complex formation in a diiron hydroxylase. Proc Natl Acad Sci USA 2008;105:19194-19198.
28. Ekiert DC, Bhabha G, Elsliger MA, Friesen RH, Jongeneelen M, Throsby M, Goudsmit J, Wilson IA. Antibody recognition of a highly conserved influenza virus epitope. Science 2009;324:246-251.
29. Fleishman SJ, Whitehead TA, Ekiert DC, Dreyfus C, Corn JE, Strauch EM, Wilson IA, Baker D. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. Science 2011;332:816-821.
30. Urvoas A, Guellouz A, Valerio-Lepiniec M, Graille M, Durand D, Desravines DC, van Tilbeurgh H, Desmadril M, Minard P. Design, production and molecular structure of a new family of artificial alpha-helicoidal repeat proteins (alphaRep) based on thermostable HEAT-like repeats. J Mol Biol 2010;404:307-327.
31. Kyomuhendo P, Myrnes B, Brandsdal BO, Smalas AO, Nilsen IW, Helland R. Thermodynamics and structure of a salmon cold active goose-type lysozyme. Comp Biochem Physiol B Biochem Mol Biol 2010;156:254-263.
32. Leysen S, Vanderkelen L, Van Asten K, Vanheuverzwijn S, Theuwis V, Michiels CW, Strelkov SV. Structural characterization of the PliG lysozyme inhibitor family. J Struct Biol 2012;180:235-242.
33. Pons C, D'Abramo M, Svergun DI, Orozco M, Bernado P, Fernandez-Recio J. Structural characterization of protein–protein complexes by integrating computational docking with small-angle scattering data. J Mol Biol 2010;403:217-230.
34. Helland R, Larsen RL, Finstad S, Kyomuhendo P, Larsen AN. Crystal structures of g-type lysozyme from Atlantic cod shed new light on substrate binding and the catalytic mechanism. Cell Mol Life Sci 2009;66:2585-2598.
35. Chelliah V, Blundell TL, Fernandez-Recio J. Efficient restraints for protein–protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment. J Mol Biol 2006;357:1669-1682.
36. Leysen S, Vanderkelen L, Weeks SD, Michiels CW, Strelkov SV. Structural basis of bacterial defense against g-type lysozyme-based innate immunity. Cell Mol Life Sci 2013;70:1113-1122.
37. Zhang L, Hermans J. Hydrophilicity of cavities in proteins. Proteins 1996;24:433-438.
38. Kastritis PL, Bonvin AM. Are scoring functions in protein–protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. J Proteome Res 2010;9:2216-2225.
39. Moal IH, Agius R, Bates PA. Protein–protein binding affinity prediction on a diverse set of structures. Bioinformatics 2011;27:3002-3009.
40. Fleishman SJ, Whitehead TA, Strauch EM, Corn JE, Qin S, Zhou HX, Mitchell JC, Demerdash ON, Takeda-Shitaka M, Terashi G, Moal IH, Li X, Bates PA, Zacharias M, Park H, Ko JS, Lee H, Seok C, Bourquard T, Bernauer J, Poupon A, Aze J, Soner S, Ovali SK, Ozbek P, Tal NB, Haliloglu T, Hwang H, Vreven T, Pierce BG, Weng Z, Perez-Cano L, Pons C, Fernandez-Recio J, Jiang F, Yang F, Gong X, Cao L, Xu X, Liu B, Wang P, Li C, Wang C, Robert CH, Guharoy M, Liu S, Huang Y, Li L, Guo D, Chen Y, Xiao Y, London N, Itzhaki Z, Schueler-Furman O, Inbar Y, Potapov V, Cohen M, Schreiber G, Tsuchiya Y, Kanamori E, Standley DM, Nakamura H, Kinoshita K, Driggers CM, Hall RG, Morgan JL, Hsu VL, Zhan J, Yang Y, Zhou Y, Kastritis PL, Bonvin AM, Zhang W, Camacho CJ, Kilambi KP, Sircar A, Gray JJ, Ohue M, Uchikoga N, Matsuzaki Y, Ishida T, Akiyama Y, Khashan R, Bush S, Fouches D, Tropsha A, Esquivel-Rodriguez J, Kihara D, Stranges PB, Jacak R, Kuhlman B, Huang SY, Zou X, Wodak SJ, Janin J, Baker D. Community-wide assessment of protein–interface modeling suggests improvements to design methodology. J Mol Biol 2011;414: 289-302.
41. Witten IH, Frank E, Hall MA. Data mining: practical machine learning tools and techniques, 3rd ed. San Francisco, CA: Morgan Kaufmann; 2011.
42. Moretti R, Fleishman SJ, Agius R, Torchala M, Bates PA, Kastritis PL, Rodrigues JPGLM, Trellet M, Bonvin AMJJ, Cui M, Rooman M, Gillis D, Dehouck Y, Moal IH, Romero M, Perez-Cano L, Pallara C, Jimenez B, Fernandez-Recio J, Samuel Flores S, Pacella

M, Kilambi KP, Gray JJ, Grudinin S, Umeyama H, Iwadate M, Esquivel-Rodríguez J, Kihara D, Zhao N, Korkin D, Zhu X, Demerdash ON, Mitchell JC, Nakamura H, Lee H, Park H, Seok C, Standley D, Shimoyama H, Terashi G, Takeda-Shitaka M, Beglov D, Hall DR, Kozakov D, Vajda S, Pierce BG, Hwang H, Vreven T, Weng Z, Huang Y, Li H, Yang X, Ji X, Liu S, Xiao Y, Zacharias M, Qin S, Zhou H-X, Huang S-Y, Zou X, Velankar S, Janin J, Wodak SJ, Baker D. Community-wide evaluation of methods for predicting the effect of mutations on protein–protein interactions. Proteins, in press.

43. 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.
44. Moal IH, Fernandez-Recio J. Intermolecular contact potentials for protein–protein interactions extracted from binding free energy changes upon mutation. J Chem Theory Comput 2013;9:3715-3727.

3.2.2. A Protein-RNA Docking Benchmark (II): Extended Set from Experimental and Homology Modeling Data.

Laura Pérez-Cano[1], Brian Jiménez-García[1] and Juan Fernandez-Recio[1*]

[1]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, Barcelona, Spain*

*Corresponding author

PROTEINS
STRUCTURE • FUNCTION • BIOINFORMATICS

A protein-RNA docking benchmark (II): Extended set from experimental and homology modeling data

Laura Pérez-Cano, Brian Jiménez-García, and Juan Fernández-Recio*

Joint BSC-IRB research programme in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center (BSC), Barcelona 08034, Spain

ABSTRACT

We present here an extended protein–RNA docking benchmark composed of 71 test cases in which the coordinates of the interacting protein and RNA molecules are available from experimental structures, plus an additional set of 35 cases in which at least one of the interacting subunits is modeled by homology. All cases in the experimental set have available unbound protein structure, and include five cases with available unbound RNA structure, four cases with a pseudo-unbound RNA structure, and 62 cases with the bound RNA form. The additional set of modeling cases comprises five unbound-model, eight model-unbound, 19 model-bound, and three model-model protein–RNA cases. The benchmark covers all major functional categories and contains cases with different degrees of difficulty for docking, as far as protein and RNA flexibility is concerned. The main objective of this benchmark is to foster the development of protein–RNA docking algorithms and to contribute to the better understanding and prediction of protein–RNA interactions. The benchmark is freely available at http://life.bsc.es/pid/protein–rna-benchmark.

Proteins 2012; 80:1872–1882.

Key words: protein–RNA interactions; structural prediction; computational docking; docking benchmark; protein and RNA flexibility.

INTRODUCTION

Protein interactions are essential in life processes, and theoretical and computational approaches can complement existing experimental data and contribute to their study and understanding. Certainly, the field of protein–protein interactions has benefited from the recent developments in docking techniques[1–4] and blind tests like Critical Assessment of PRediction of Interactions (CAPRI; http://www.ebi.ac.uk/msd-srv/capri), an international blind prediction experiment to evaluate the performances of protein–protein computational docking methods.[5–8]

The interactions of proteins with other biomolecules, such as nucleic acids, are also becoming the focus of structural and computational studies to understand essential biological processes. In recent years, the growing awareness for the importance of protein–RNA interactions, together with the publication of the 50S and 30S ribosome subunits,[9,10] have increased the volume of data on protein–RNA complexes. As a consequence, a number of studies have used available structural data of real protein–RNA interfaces to understand this type of interaction and extract better parameters for predictions.[11] Indeed, characterizing the molecular mechanism of protein–RNA recognition to understand and predict protein–RNA complexes from their separate components is one of the grand challenges in structural biology.

However, there are still very few reported methods for protein–RNA docking and scoring,[12–15] and practically inexistent systematic benchmarks on large data sets in comparison with that of other biomolecules. In this context, the above-mentioned CAPRI experiment recently encouraged modeling groups to adapt existing protein–protein docking methods or develop new ones for the

Additional Supporting Information may be found in the online version of this article.

Abbreviations: CAPRI, critical assessment of PRediction of interactions; NMR, nuclear magnetic resonance.

Grant sponsor: Plan Nacional I+D+i from the Spanish Ministry of Science; Grant number: BIO2010-22324
*Correspondence to: Juan Fernández-Recio; Life Sciences Department, Barcelona Supercomputing Center (BSC), Jordi Girona 29, 08034 Barcelona, Spain.
E-mail: juanf@bsc.es.
Received 27 October 2011; Revised 28 February 2012; Accepted 30 March 2012
Published online 6 April 2012 in Wiley Online Library (wileyonlinelibrary.com).
DOI: 10.1002/prot.24075

protein–RNA docking problem. The experiment (round 15, targets 33 and 34) nicely showed how some docking methods can be adapted to predict the tridimensional structure of a protein–RNA complex. Indeed, RosettaDock[16,17] submitted a medium quality model to the predictors section, with ligand RMSD of 1.7Å when compared with the complex crystal structure.[18] In the scorers section, MDockPP identified the best model among all participants,[19] an acceptable model with ligand RMSD of 3.1Å. In addition, a variation of the pyDock docking and scoring protocol,[20] which achieved successful results for protein–protein docking in past CAPRI tests,[21] also found an acceptable model that was the second best one among all participants in the scorers section, with remarkable ligand RMSD of 3.8Å with respect to the X-ray structure of the complex.[22] However, this experiment pointed out also the limitations of current methods, as all successful models were generated using the bound RNA structure, while no success was achieved using a model of the unbound RNA molecule. In this context, the main challenge is the high degree of conformational movement in RNA molecules,[23] for which a better treatment of flexibility is required. In addition, better physics-based or empirical scoring parameters need to be specifically adapted for protein–RNA binding.

To help to these purposes, we have compiled here an extended protein-RNA benchmark set composed of 106 docking test cases in a similar manner that has been previously reported for protein–protein and protein–DNA docking.[24,25] It represents a realistic test set that covers major functional categories of RNA,[26] containing cases with different degrees of difficulty. The aim of the benchmark is to facilitate and foster the development of protein–RNA docking algorithms.

MATERIALS AND METHODS

Structural set compilation

We based our benchmark on our previously reported set of 315 protein–RNA complex structures, which included cases with nonredundant proteins (up to 70% sequence identity), as well as cases with redundant proteins that were bound to different RNA molecules to achieve more variety in protein–RNA interfaces.[27] We also included nonredundant cases from the PRIDB (http://pridb.gdcb.iastate.edu/).[28] Then, we screened the PDB[29] in search of structures for the unbound protein and RNA partners. For each protein and RNA molecule in the complex data set, we obtained a list of related structures using the PDB Advanced Sequence search (with the Blast option). We considered as unbound structures those ones free from other molecules with sequence identity higher than 95% and *E*-value smaller than 0.003 with respect to the reference complex structure, and containing >70% of the interface residues. Supporting Information Table S1 shows the percentage of interface residues contained in the unbound protein or RNA structures, since this can be important to be considered in docking. In the case of RNA molecules, given the scarcity of fully free structures, we also included pseudo-unbound RNA structures, that is, those bound to a protein that had less than 35% sequence identity with respect to that in the reference complex structure. We extended the set to cases without available structure for the unbound protein or RNA, for which a reasonable model could be built (see next section). For that, we searched for homologous unbound template structures with sequence identity ≥35% for proteins and ≥70% for RNA. We also considered cases in which the template was bound to other protein or RNA molecule, as long as the latter showed <70% of sequence identity with respect to the partner molecule in the complex structure. Finally, we also included cases with no available unbound or homologue RNA structure for which the bound RNA coordinates can be used, since they can be also useful for developing and testing docking tools. All considered X-ray structures showed resolution better than 3.5 Å. In the case of nuclear magnetic resonance (NMR) ensemble structures, we selected the first model. We discarded one unbound-unbound case (1A1T complex PDB) in which the NMR unbound protein structure (1MFS PDB) is mostly disordered. Residue numbering correspondence between unbound molecules and target complexes is provided for each case on the downloaded version of the benchmark.

Model building

For protein or RNA cases with no available unbound structure, we built models provided we could find a reasonable template (see below). The input sequences for the modeling procedure were extracted from the complex PDB structures, and the chain IDs of the models were inherited from the template PDB structures.

The protein models were built with MODELLER 8v1,[30] based on homologous template structures with more than 35% sequence identity. For each case, we produced 10 different models and selected the one with the best DOPE score. We used for all cases the MODELLER 8v1 alignment, except for one case (PDB code 1HQ1) in which we changed the default alignment to the one provided by BLAST (http://blast.ncbi.nlm.nih.gov/Blast.cgi)[31] to achieve a GA341 score proximal to 1.0 (a MODELLER score which indicates the reliability of the model).

The RNA models were built with the standalone version of the ModeRNA v1.6.0 program,[32] which produced models based on the provided alignments. It has to be noted that the method does not extensively optimize van der Waals interactions, which might imply an additional difficulty for protein–RNA docking methods. However, given that the RNA models in current benchmark are

based on highly homologous templates (≥70% sequence identity; cutoff based on data from the ModeRNA authors)[32] and hence have very few insertion/deletions, we relied on the default ModeRNA modeling procedure. The program uses a fragment-based approach for modeling indels, selecting optimal conformations from the 50 best-scoring fragment candidates, and remodeling imperfect backbone conformations by optimizing interatomic clashes and acceptable bond lengths, bond angles, and torsion angles. We actually checked that the final modeled structures had very few clashes. For all cases, we used the global alignment provided by LALIGN server (http://www.ch.embnet.org/software/LALIGN_form.html). In cases with unusual or post-transcriptionally modified nucleotides in the templates, they were included in the alignment with the specific nomenclature in the MODOMICS database (http://modomics.genesilico.pl/modifications/) to be processed by the modeling procedure. We checked that these residues were mostly conserved between the experimental unbound and bound RNA structures, so they will be probably important for docking. Some modified nucleotides in the complex were not found in the template and therefore were not modelled, but this is within the expected noise in a realistic scenario of docking predictions.

Structural analysis

We analyzed the variability, the size of the interaction interface between protein and RNA and the conformational changes between the bound and the unbound forms:

1. We classified each docking case according to the SCOR 1.2 functional classification of the RNA[26] as tRNA, rRNA, ribozyme, snRNA, SRP RNA, genetic control elements, vRNA, SELEX RNA, synthetic RNA, and "other."
2. The size of the protein–RNA interface is expressed in terms of BSA. We calculated both the protein and the RNA BSA as the difference between the accessible surface area (ASA) of the dissociated and bound states [Eq. (1)], using the ICM software (http://www.molsoft.com).

$$BSA_{bound} = ASA_{dissoc} - ASA_{bound} \quad (1)$$

3. We analyzed the conformational changes due to the RNA backbone flexibility and protein domain reorientations based on the RMSD of their phosphorus or Cα atoms, respectively, after superimposing these atoms in the protein or the RNA unbound structures onto the corresponding ones in the reference structure. This is consistent with the recently reported protein–DNA benchmark in which they used the P atom to compute RMSD values for DNA.[25] Nevertheless, we checked that using the RNA C4' atom instead would not significantly change the global RMSD values. To check the effect of conformational changes in the protein–RNA interfaces, we also calculated the interface RMSD, considering only Cα and phosphorus atoms located at 10 Å distance from any other atom of the partner molecule.

RESULTS

Composition of the protein–RNA benchmark

We have compiled an extended set of protein–RNA complexes that can be used to develop and test protein–RNA docking methods. Currently, the protein–RNA docking benchmark is composed of 106 test cases, with 71 cases in which the coordinates for the interacting subunits were taken from crystallography or NMR experimental structures (Table I), and 35 cases in which at least one of the interacting subunits was built by homology-based modeling (Table II). All cases in the experimental set have available unbound protein structure, and include five cases with available unbound RNA structure, four cases with a pseudo-unbound RNA structure, and 62 cases with the bound RNA form (Table I). The additional set of modeling cases comprises five unbound-model, eight model-unbound, 19 model-bound, and three model-model protein–RNA cases (Table II). The models generated for the unbound protein or RNA molecules were similarly close to the bound structures in terms of RMSD as the templates used to build them, except some cases in which the models were closer to the bound structures than their templates (Supporting Information Table S2). The set of homology modeling cases is an addition to the benchmark to extend the number of cases that could be used for docking development. We provided here specific models to help automatic calculations and comparisons, but the ones included here are not by any means the only possible models. Besides, docking developers should be aware that using models for docking will increase uncertainty. But on the other side, they represent a real challenge that need to be solved, perhaps by developing new strategies to treat ensembles of modeled conformers, or by improving the treatment of flexibility to overcome possible inaccuracies of the models. The benchmark covers all major functional categories of RNA.[26] Of the 25 cases with available unbound, pseudo-unbound or modeled RNA structure, the majority are of protein-transfer RNA type (15 cases). However, among the 81 cases in which only the bound RNA structure is available, a significant number of cases are genetic regulatory elements (21 cases) or synthetic RNA ligands (20 cases), usually formed by ssRNAs with no defined structure in free solution.

Conformational changes in protein–RNA docking test cases

Regarding protein and RNA flexibility, we can classify the cases into those with small, medium, and large con-

Table I
Protein–RNA Docking Benchmark: Extended Set From Experimental Cases

Complex			Protein			RNA			
PDB[a]	Protein/RNA description	Type[b]	PDB[c]	RMSD[d]	BSA[e]	PDB[c]	RMSD[d]	BSA[e]	I-RMSD[f]
Unbound-unbound (five cases)									
1ASY_a,b:r/s	Saccharomyces cerevisiae *aspartyl-tRNA synthetase/Asp-tRNA*	1	1EQV_a,b	1.4	2088	3TRA_a	5.0	2201	2.7
1DFU_p:m,n	Escherichia coli *ribosomal protein L25/loop E of 5S rRNA*	2	1B75_a	4.3	815	364D_b,c	3.3	841	4.7
1OB2_a:b	Escherichia coli *elongation factor EF-TU/Phe-tRNA*	1	1EFC_a	11.2	1241	1EHZ_a	2.7	1435	10.6
1R3E_a:c,d,e	Thermotoga Maritima *tRNA pseudouridine synthase TruB/RNA substrate*	1	1ZE1_a	5.2	1349	1EHZ_a	2.3	1537	7.2
2FMT_a:c	Escherichia coli *methionyl-tRNAfMet formyltransferase/ formyl-methionyl-tRNAfMet*	1	1FMT_a	1.2	1573	3CW5_a	2.9	1590	2.2
Unbound-pseudo-unbound (four cases)									
1B23_p:r	Thermus Aquaticus *elongation factor EF-TU:GTP/E. Coli cysteinyl-tRNA*	1	1TUI_a	10.0	1286	1U0B_a	6.6	1457	10.4
1MFQ_c:a	Homc sapiens *SRP 54kDa protein/7S RNA of human SRP*	5	1QB2_b	3.1	538	1L9A_b	5.1	539	2.5
1QTQ_a:b	Escherichia coli *glutaminyl-tRNA synthetase/Gln-tRNA*	1	1NYL_a	1.6	2360	3KNH_y	4.6	2695	2.7
1U0B_b:a	Escherichia coli *cysteinyl-tRNA synthetase/Cys-tRNA*	1	1LI7_a	1.0	2099	1B23_r	6.6	2302	3.6
Unbound-bound (62 cases)[g]									
1B7F_a:p	Drosophila melanogaster *sxl-lethal protein/RNA*	9	3SXL_a	6.7	1348	1B7F_p	0.0	1567	6.9
1C9S_J,m,n,o,p,q,r,s,t,u,v:w	Geobacillus stearothermophilus *trp RNA-binding attenuation protein/ssRNA*	9	1OAW_a,b,c,d,e,f,g,h,i,j,k	0.4	16466	1C9S_w	0.0	29294	0.4
1DK1_a:b	Thermus thermophilus *ribosomal protein S15/16S rRNA fragment*	2	2FKX_a	2.8	1226	1DK1_b	0.0	1234	2.4
1E7K_a:c	Homc sapiens *spliceosomal 15.5kd protein/U4 snRNA fragment*	4	2JNB_a	3.2	610	1E7K_c	0.0	659	1.8
1EC6_a:d	Homc sapiens *Nova-2 KH3 K-homology RNA-binding domain/RNA hairpin*	8	1DTJ_a	1.6	873	1EC6_d	0.0	906	0.6
1EIY_a,b,d,e:c/f[h]	Thermus thermophilus *phenylalanyl-tRNA synthetase/Phe-tRNA*	1	1JJC_a,b,d,e[h]	1.3	2143	1EIY_c	0.0	2272	1.8
1EKZ_a:b	Drosophila melanogaster *Staufen dsRBD/RNA hairpin*	6	1STU_a	5.2	596	1EKZ_b	0.0	548	3.4
1F7U_a:b	Saccharomyces cerevisiae *arginyl-tRNA synthetase/Arg-tRNA*	1	1BS2_a	3.4	2591	1F7U_b	0.0	2992	2.5
1G1X_a:d	Thermus thermophilus *ribosomal protein S6/16S rRNA fragment*	2	1RIS_a	1.9	179	1G1X_d	0.0	176	2.2
1H3E_a:b	Thermus thermophilus *tyrosyl-tRNA synthetase/Tyr- tRNA(gua) and ATP and tyrosinol*	1	1H3F_a	9.4	1223	1H3E_b	0.0	1297	0.2
1H4S_a:b,t	Thermus thermophilus *prolyl-tRNA synthetase/Pro-tRNA(cgg) and prolyl-adenylate analogue*	1	1HC7_a,b	1.4	1123	1H4S_t	0.0	1252	0.9
1HC8_a:c	Bacillus stearothermophilus *ribosomal protein L11/23S rRNA fragment*	2	1FOY_a	2.9	990	1HC8_c	0.0	984	2.5
1HVU_a,b:c	Human immunodeficiency virus 1 reverse transcriptase/RNA pseudoknot	9	2VG5_a,b	5.4	621	1HVU_c	0.0	607	6.4
1JBR_b:d	Aspergillus restrictus *ribotoxin restrictocin/23S rRNA saicin-ricin domain*	2	1AQZ_a	0.6	702	1JBR_d	0,0	754	0.5
1K8W_a:b	Escherichia coli *pseudouridine synthase TruB/T stem-loop RNA*	1	1R3F_a	2.2	1407	1K8W_b	0.0	1649	1.7
1KOG_a:i	Escherichia coli *threonyl-tRNA synthetase/essential domain of its mRNA operator*	6	1EVL_a	0.6	851	1KOG_i	0.0	862	0.5
1KQ2_a,b,h,i,k,m:r	Staphylococcus aureus *Hfq protein/RNA*	9	1KQ1_a,b,h,i,k,m	1.4	1358	1KQ2_r	0.0	1642	1.1
1M5O_c:a,b	Homc sapiens *U1 small nuclear ribonucleoprotein A/RNA hairpin ribozyme and RNA substrate*	3	1NU4_a	1.6	869	1M5O_a,b	0.0	950	1.8

(Continued)

Table I
(Continued)

Complex			Protein			RNA			
PDB[a]	Protein/RNA description	Type[b]	PDB[c]	RMSD[d]	BSA[e]	PDB[c]	RMSD[d]	BSA[e]	I-RMSD[f]
1M8V_a,b,c,d,e,f,g:o/p/q/r/s/t/u	Pyrococcus abyssii *sm protein/uridine heptamer*	9	1H64_a,b,c,d,e,f,g	0.6	309	1M8V_o	0.0	365	0.4
1M8W_a:c	Homo sapiens *pumilio-homology domain/NRE1-19 RNA*	6	1M8Z_a	1.2	941	1M8W_c,e	0.0	1159	0.7
1MMS_a:c	Thermotoga maritima *ribosomal protein L11/23S rRNA fragment*	2	2K3F_a	4.2	1200	1MMS_c	0.0	1239	3.6
1N78_a:c	Thermus thermophilus *glutamyl-tRNA synthetase/Glu-tRNA and glutamol-AMP*	1	1J09_a	1.9	2064	1N78_c	0.0	2308	1.5
1Q2R_a:e	Zymomonas mobilis *catalytic tRNA guanine transglycosylase/RNA substrate*	1	1R5Y_a	0.8	1232	1Q2R_e	0.0	1362	1.1
1SER_a,b:t	Thermus thermophilus *seryl-tRNA synthetase/Ser-tRNA*	1	1SES_a,b	1.9	1095	1SER_t	0.0	1085	2.3
1T0K_a,b:c,d	Escherichia coli *MBP*-Saccharomyces cerevisiae *L30e fusion protein/pre-mRNA*	6	1NMU_a,b	1.5	466	1T0K_c,d	0.0	507	3.3
1T4L_b:a	Saccharomyces cerevisiae *dsRBD of Rnt1p RNase III/5′ terminal RNA hairpin of snR47 precursor*	4	1T4O_a	2.2	924	1T4L_a	0.0	795	1.7
1U63_a:b	Methanocaldococcus jannaschii *ribosomal protein L1/mRNA fragment*	2	1I2A_a	1.3	1135	1U63_b	0.0	1075	1.0
1WNE_a:b,c	Foot and Mouth Disease Virus RNA-dependent RNA polymerase/template-primer RNA	6	1U09_a	0.7	1426	1WNE_b,c	0.0	1597	0.7
1WPU_a:c	Bacillus subtilis *Hut operon positive regulatory protein/Hut mRNA*	6	1WPV_a	0.2	608	1WPU_c	0.0	700	0.2
1WSU_a:e	Moorella thermoacetica C-terminal domain of elongation factor selB/SECIS RNA	6	1LVA_a	0.7	444	1WSU_e	0.0	471	0.6
1YVP_a:c,d	Xenopus laevis *Ro autoantigen /RNA*	9	1YVR_a	1.3	1072	1YVP_c,d	0.0	1125	1.3
2AD9_a:b	Homo sapiens *polypyrimidine Tract Binding protein RBD1/CUCUCU RNA*	9	1SJQ_a	2.9	617	2AD9_b	0.0	760	2.3
2ADB_a:b	Homo sapiens *polypyrimidine Tract Binding protein RBD2/CUCUCU RNA*	9	1SJR_a	3.7	594	2ADB_b	0.0	792	2.4
2ADC_a:b/c	Homo sapiens *polypyrimidine Tract Binding protein RBD34/CUCUCU RNA*	9	2EVZ_a	4.2	1497	2ADC_b	0.0	1876	3.7
2ASB_a:b	Mycobacterium tuberculosis *NusA/BoxC stem-loop motif RNA fragment*	6	1K0R_a	1.1	1076	2ASB_b	0.0	1189	0.6
2AZ0_a,b:c,d	Flock House virus B2 protein/dsRNA	9	2B9Z_a,b	1.4	1090	2AZ0_c,d	0.0	1114	1.1
2AZX_a,b:c/d[h]	Homo sapiens *tryptophanyl-tRNA synthetase/Trp-tRNA*	1	1R6T_a,b	1.0	1958	2AZX_c	0.0	2130	0.6
2BGG_a:p,q	Archaeoglobus fulgidus *piwi protein/16nt siRNA duplex*	6	1W9H_a	1.0	1008	2BGG_p,q	0.0	1179	0.9
2BH2_a:c	Escherichia coli *5-methyluridine methyltransferase ruma/rRNA and s-adenosylhomocysteine*	2	1UWV_a	1.4	2067	2BH2_c	0.0	2304	1.3
2BTE_a:b	Thermus thermophilus *leu-tRNA synthetase/Leu-tRNA and a substrate analogue*	1	1H3N_a	4.1	1628	2BTE_b	0.0	1673	2.4
2BU1_a:r	Bacteriophage ms2 coat protein/RNA harpin	7	2MS2_a	0.2	397	2BU1_r	0.0	445	0.1
2C0B_a,b,c,d:e/f/g/h[h]	Escherichia coli *RNase E/RNA*	9	2VMK_a,b,c,d	19.3	989	2C0B_e	0.0	1034	17.9
2CZJ_a:b/d	Thermus thermophilus *hb8 SsrA-binding protein/tmRNA*	6	1WJX_a	1.4	1799	2CZJ_b	0.0	1715	1.1
2ERR_a:b	Homo sapiens *RBD of Fox-1/UGCAUGU RNA*	9	2CQ3_a	3.4	840	2ERR_b	0.0	1101	0.1
2F8K_a:b	SAM domain of Saccharomyces cerevisiae *Vts1/RNA ligand*	9	2D3D_a	0.5	429	2F8K_b	0.0	432	0.5
2FK6_a,b:r/s[h]	Bacillus subtilis *Rnase Z/Thr-tRNA*	1	1Y44_a,b	1.7	1095	2FK6_r	0.0	1162	2.1
2GIC_a,b,c,d,e,f,g,h,i,j:r,k[h]	Vesicular stomatitis *virus nucleocapsid protein/RNA*	7	3PTO_a,b,c,d,e,f,g,h,i,j[h]	0.8	10013	2GIC_r,k	0.0	10210	0.7

(Continued)

Table I
(Continued)

Complex			Protein			RNA			
PDB[a]	Protein/RNA description	Type[b]	PDB[c]	RMSD[d]	BSA[e]	PDB[c]	RMSD[d]	BSA[e]	I-RMSD[f]
2GJE_a,d:r,s	Trypanosoma brucei *treu927 guideRNA-binding protein/gRNA*	6	2GIA_a,b	4.2	859	2GJE_r,s	0.0	836	4.2
2GJW_a,b:e,r,h	Archaeoglobus fulgidus *thNA-splicing endonuclease/tNA*	9	1H0V_a,b	1.6	1508	2GJW_e,r,h	0.0	1648	1.6
2HGH_a:b	Xenopus laevs *transcription factor IIIA zinc fingers 4-6/5S rRNA 55mer*	2	2J7J_a	12.0	1314	2HGH_b	0.0	1201	9.0
2HW8_a:b	Thermus thermophilus *ribosomal protein L1/Methanococcus vannielii mRNA fragment*	6	1AD2_a	6.7	1170	2HW8_b	0.0	1207	5.7
2I91_b:e,f	Xenopus laevs *Ro autoantigen/misfolded pre-5S rRNA fragment*	2	1YVR_a	1.2	1832	2I91_e,f	0.0	1924	1.4
2IX1_a:b	Escherichia coli *RNASE III/RNA*	10	2I00_a	1.4	1877	2IX1_b	0.0	2106	1.0
2PY9_a,b:f	KH1 domain of Homo sapiens *poly(rC)-binding protein 2/*Homo sapiens *telomeric RNA fragment*	6	2JZX_a	2.5	492	2PY9_f	0.0	520	2.1
2QUX_a,b:c	Pseudomonas phage pp7 coat protein/RNA harpin	7	2QUD_a,b	0.7	780	2QUX_c	0.0	900	0.3
2R7R_a:x	Rotavirus RNA-dependent RNA polymerase VP1/RNA	9	2R7Q_a	0.6	890	2R7R_x	0.0	1004	1.5
3BO2_a:b,c,d,e	Homo sapiens *U1 small nuclear ribonucleoprotein A/Group I intron p9*	3	1NU4_a	1.7	860	3BO2_b,c,d,e	0.0	938	1.7
3BSB_b:c	Homo sapiens *Pumilio1 protein/CyclinB reverse RNA*	6	1M8Z_a	1.8	1203	3BSB_c	0.0	1447	0.9
3BSO_a:p,t	Norwalk Virus polymerase/primer-template RNA and CTP	9	1SH0_a	1.3	1513	3BSO_p,t	0.0	1670	1.5
3BSX_a:c	Homo sapiens *Pumilio 1 protein/Puf5 RNA*	6	1M8Z_a	1.5	1205	3BSX_c	0.0	1478	0.6
3BX2_a:c	Saccharomyces cerevisiae *Puf4 RBD/HO endonuclease RNA 3′ UTR recognition sequence*	6	1M8Z_a	3.5	1100	3BX2_c	0.0	1317	1.8
3CIY_a:c,d	Mus musculus Toll-like receptor 3 ectodomain/dsRNA	9	3CIG_a	1.2	1133	3CIY_c,d	0.0	1102	1.4

[a]PDB code of the protein-RNA complex. Protein and RNA chains are separated by colon. Alternative binding modes are separated by a slash ("/").
[b]Functional classification of RNA according to SCOR 1.2 database[36]: 1 = tRNA, 2 = rRNA, 3 = ribozyme, 4 = snRNA, 5 = SRP RNA, 6 = Genetic control elements, 7 = vRNA, 8 = SELEX RNA, 9 = synthetic RNA, and 10 = other.
[c]PDB code of the protein or RNA subunits.
[d]RMSD (Å) between Cα atoms (for proteins) or phosphorous atoms (for RNA) of individual bound reference and unbound structures.
[e]Buried Surface Area (Å^2) of the unbound molecule upon complex formation [see Eq. (1) in Methods].
[f]RMSD (Å) of interface Cα and phosphorus atoms of unbound protein and RNA structures, and the equivalent atoms in the complex.
[g]Unbound RNA is not available or cannot be modeled, so the bound form is used.
[h]The biological assembly from the PDB has been considered, therefore some of the chains in the complex have been renamed.

Table II
Protein–RNA Docking Benchmark: Extended Set From Homology Modeling Cases

Complex			Protein			RNA			
PDB[a]	Protein/RNA description	Type[b]	PDB[c]	RMSD[d]	BSA[e]	PDB[c]	RMSD[d]	BSA[e]	I-RMSD[f]
Unbound-model (five cases)									
1C0A_a:b	Escherichia Coli *aspartyl-tRNA synthetase/Asp-tRNA*	1	1EQR_a	1.6	2140	1EFW_c	2.6	2272	1.9
1EFW_a:c	Thermus thermophilus *aspartyl-tRNA synthetase/E. Coli Asp-tRNA*	1	1L0W_a	1.3	1201	1C0A_b	1.6	1166	1.1
1J1U_a,c:b/d	Methanocadococcus jannaschii *tyrosyl-tRNA synthetase/Tyr-tRNA*	1	1U7D_a,b	2.4	1049	2XUY_v	2.2	1125	1.9
1J2B_a,b:c	Pyrococcus horikoshii tRNA-Guanine Transglycosylase/ lambda-form Val-tRNA	1	1IQ8_a,b	0.9	3254	2ZUF_b	10.4	3602	5.5
2DRA_a:b	Archaeoglobus fulgidus *CCA-adding enzyme with tRNAminiDCC and ATP*	1	1R89_a	1.9	1363	1VFG_d	3.0	1613	2.1
Model-unbound (eight cases)									
1FEU_a:b,c	Thermus Thermophilus *ribosomal protein L25/fragment of 5S rRNA*	2	1NJP_t	4.4	789	1A4D_a,b	5.0	786	3.6
1HQ1_a:b	M domain of Escherichia coli *SRP Ffh protein/domain IV of 4.5S SRP RNA*	5	2FFH_a	1.2	669	1CQL_a	9.5	660	4.7
1LNG_a:b	Methanocaldococcus jannaschii *SRP 19kDa protein/7S.S SRP RNA*	5	1KVV_a	4.1	1099	1Z43_a	2.0	1185	3.6
1OOA_a:c	Mus musculus nuclear factor NF-kappa-B p105 subunit/high-affinity RNA aptamer	8	3DO7_b	6.4	951	2JWV_a	6.0	859	4.7
1RKJ_a:b	N-terminal RNA-binding domains of Mesocricetus auratus nucleolin/pre-rRNA target	2	2KRR_a	10.1	1056	1QWA_a	4.9	1183	8.3
2R8S_l,h:r	Mus musculus specific synthetic FAB/P4-P6 RNA ribozyme domain	3	2FJF_l,h	1.9	1181	1HR2_a	4.0	1279	2.2
2V3C_c:m	Methanocadococcus jannaschii *SRP54 protein/7S.S SRP RNA*	5	3DM5_a	13.0	1371	1Z43_a	1.7	1371	12.5
2ZKO_a,b:c,d	NS1 protein of human influenza virus A/dsRNA	9	2Z0A_a,b	1.0	1197	2ZI0_c,d	4.4	1182	2.6
Model-bound (19 cases)[g]									
1DDL_a,b,c:d	Desmodium yellow mottle tymovirus coat protein/vRNA fragment	7	1AUY_a,b,c	1.1[h]	572	1DDL_d,e	0.0	657	0.6[h]
1E8O_a,b:e/c,d:e	Homo sapiens *SRP 9KDa and 16 Kda proteins/Alu RNA 5' domain*	5	1914_a,b	1.8	519	1E8O_e	0.0	507	0.6
1FXL_a:b	Homo sapiens *antigen HUD RRM12 domains/fragment of the class I c-fos AU-rich element*	6	3SXL_a	7.4	1139	1FXL_b	0.0	1257	7.6
1K1G_a:b	Homo sapiens *SF1-Bo isoform/yeast and mammalian pre-mRNA transcript intron BPS*	6	2BL5_a	10.7	1142	1K1G_b	0.0	1343	11.2
1MJI_a:d/c	Thermus thermophilus *ribosomal protein L5/5S rRNA fragment*	2	1IQ4_a	3.2	862	1MJI_d	0.0	828	1.1
1RLG_a:c	Archaeoglobus fulgidus *50S ribosomal protein L7Ae/box C/D RNA fragment*	10	1XBI_a	2.4	631	1RLG_c	0.0	625	1.6
1S03_h:a	Escherichia coli *ribosomal Protein S8/spc Operon mRNA fragment*	6	1SEI_a	1.6	833	1S03_a	0.0	883	1.5
2B3J_a,b:e/f	Staphylococcus aureus *tRNA Adenosine Deaminase/ anticodon stem-loop of Arg-tRNA*	1	1WWR_a,b	2.2	1011	2B3J_e	0.0	1051	2.8
2CJK_a:b	Saccharomyces cerevisiae *nuclear polyadenylated RBP4/RNA*	9	1L3K_a	12.8	1165	2CJK_b	0.0	1450	13.1

(Continued)

Table II
(Continued)

Complex			Protein			RNA			
PDB[a]	Protein/RNA description	Type[b]	PDB[c]	RMSD[d]	BSA[e]	PDB[c]	RMSD[d]	BSA[e]	I-RMSD[f]
2CSX_a:c	Aquifex aeolicus *methionyl-tRNA synthetase/Met-tRNA*	1	2D5B_a	2.2	1023	2CSX_c	0.0	1093	2.1
2D6F_a,b,c,d:e/f	Methanothermobacter thermautotrophicus Glutamyl-tRNA amidotransferase/Glu-tRNA	1	1ZQ1_a,b,c,d	2.9	616	2D6F_e	0.0	602	3.7
2DB3_a:e	Drosophila melanogaster *DEAD-box protein Vasa/ssRNA*	9	2I4I_a	15.8	492	2DB3_e	0.0	678	16.5
2DER_b:d	Escherichia coli *tRNA-specific 2-thiouridylase mnmA/Glu-tRNA*	1	2HMA_a	2.8	1125	2DER_d	0.0	1281	3.3
2DLC_a,b:y/z[i]	Saccharomyces cerevisiae *tyrosyl-tRNA synthetase/Tyr-tRNA*	1	1N3L_a,b[i]	6.1	999	2DLC_y	0.0	1052	4.4
2FY1_a:b	Homo sapiens *RNA-binding motif protein Y/S1A stem-loop RNA*	8	1X5S_a	5.5	1072	2FY1_b	0.0	1117	3.1
2I82_a:e	Escherichia coli *pseudouridine synthetase/anticodon stem loop of Phe-tRNA*	1	1XPI_a	4.9	1476	2I82_e	0.0	1666	5.4
2IPY_a:c	Oryctolagus cuniculus *Iron-responsive element-binding protein 1/ferritin IRE-RNA*	6	2B3Y_a	11.8	1331	2IPY_c	0.0	1411	8.4
2JPP_a,b:c/d	Pseudomonas fluorescens *RsmE/Shine-Dalgarno sequence of hcnA mRNA*	6	1Y00_a,b	4.9	970	2JPP_c	0.0	1039	4.2
2NUG_a:c,d,e,f	RNase III from Aquifex aeolicus/*dsRNA*	9	100W_a	14.2	1479	2NUG_c,d,e,f	0.0	1473	12.0
Model-model (three cases)									
1QF6_a:b	Escherichia coli *threonyl-trna synthetase complexed with its cognate tRNA*	1	1NYQ_a	4.1	2231	3FDZ_c	3.9	2269	3.2
1VFG_a:c	Aquifex aeolicus *tRNA nucleotidyltransferase/primer tRNA and an incoming ATP analog*	1	3H38_a	4.0	422	1EIY_c	3.1	397	4.7
2DU3_a,b,c,d:e/f	Archaeoglobus fulgidus *O-phosphoseryl-tRNA synthetase/Cys-tRNA and O-phosphoserine*	1	2DU7_a,b,c,d	4.5	631	2ZZN_d	3.1	709	3.7

[a]PDB code of the protein-RNA complex. Protein and RNA chains are separated by colon. Alternative binding modes are separated by a slash ("/").
[b]Functional classification of RNA according to SCOR 1.2 database[6]; 1 = tRNA, 2 = rRNA, 3 = ribozyme, 4 = snRNA, 5 = SRP RNA, 6 = Genetic control elements, 7 = vRNA, 8 = SELEX RNA, 9 = synthetic RNA and 10 = other.
[c]PDB code of the protein or RNA subunits. In cases in which the protein or RNA has been modeled, the PDB id of the template is shown. Models have the same chain IDs as their templates. In italics are shown cases with pseudo-unbound templates, i.e. bound to a different protein or RNA molecule (see main text).
[d]RMSD (Å) between $C\alpha$ atoms (for proteins) or phosphorous atoms (for RNA) of individual bound reference and unbound structures.
[e]Buried Surface Area (Å^2) of the unbound molecule upon complex formation [see Eq. (1) in Methods].
[f]RMSD (Å) of interface $C\alpha$ and phosphorus atoms of unbound protein and RNA structures, and the equivalent atoms in the complex.
[g]Unbound RNA is not available or cannot be modeled, so the bound form is used.
[h]The 24 Nt residues have not been considered for the RMSD calculation as they are swapped between target and template structures and give high RMSDs that do not reflect the true overall quality.
[i]The biological assembly from the PDB has been considered, therefore some of the chains in the complex have been renamed.

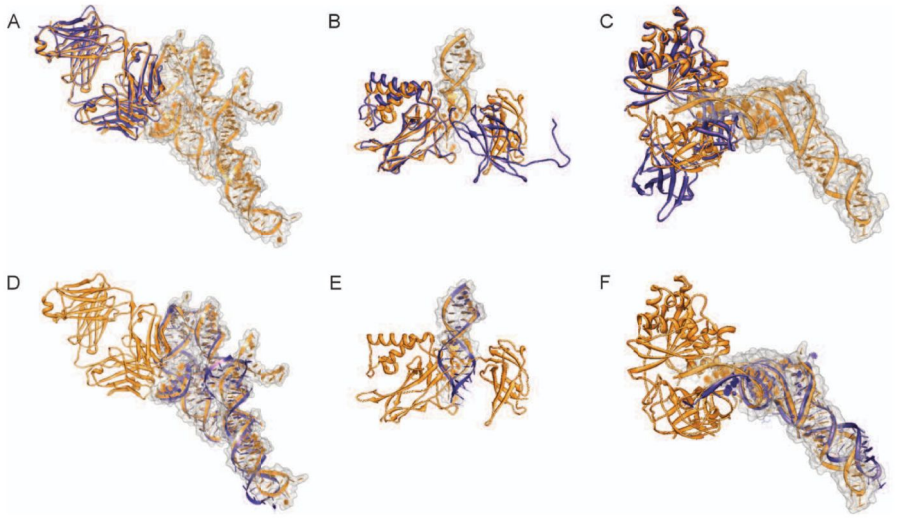


Figure 1

Examples of easy (interface RMSD < 2.5Å), intermediate (2.5Å ≤ interface RMSD ≥ 5Å) and difficult cases (interface RMSD > 5Å). Complex structures are colored in orange while unbound (or modeled) protein or RNA structures are shown in blue. (**A**,**D**) An easy case: the modeled protein and unbound RNA structures, superimposed onto a specific synthetic FAB bound to P4–P6 RNA ribozyme domain (PDB code 2R8S) (interface RMSD 2.2Å). (**B**,**E**) An intermediate case: the modeled protein and unbound RNA structures, superimposed onto the *Mus musculus* NF-KB(P50)2 bound to a high-affinity RNA aptamer (PDB code 1OOA) (interface RMSD 4.7Å). (**C**,**F**) A difficult case: the unbound protein and RNA structures, superimposed onto *E. Coli* elongation factor EF-TU bound to Phe-tRNA (PDB code 1OB2) (interface RMSD 10.6Å).

formational changes upon binding (interface RMSD below 2.5 Å, between 2.5 and 5.0 Å, and above 5.0 Å, respectively). These cases could be defined as easy, intermediate and difficult cases for docking predictions, respectively, following the definition used for protein–DNA complexes in a previous study.[25] From the point of view of RNA flexibility, the 25 cases with available unbound, pseudo-unbound or modeled RNA structure form a quite realistic docking test set, with six easy, 13 intermediate, and six difficult cases. The 81 cases in which only the bound structure for RNA is available represent a less challenging, albeit useful, docking test set, with 58 easy, 11 intermediate, and 12 difficult cases.

It is interesting to note that the typically most flexible regions in RNAs, such as the unpaired 5′ or 3′ ends or the RNA loops, are not involved in the interface in cases with small conformational changes [Fig. 1(A,D)].

In contrast, in cases with medium conformational rearrangement upon complex formation [Fig. 1(B,E)], these typically flexible regions are mostly located at interfaces, adopting an optimal conformation for the specific interaction. Furthermore, in some cases general RNA backbone flexibility is also significant when unbound and bound structures are compared. On the protein side, these cases are often proteins with flexible loops or linkers that generate protein motif or domain reorientations.

Finally, in protein–RNA docking test cases with large conformational changes upon binding, protein domain rearrangement due to interdomain linker or backbone flexibility critically affects protein–RNA interfaces [see Fig. 1(C,F)]. As a consequence, these are highly challenging cases for protein–RNA docking.

Other important challenges for protein–RNA docking

In addition to protein and RNA flexibility, the benchmark shows other key issues for docking prediction. One important difficulty is related to the complex size, which represents a bottleneck for docking sampling and scoring. In many cases, the RNA-binding protein is a large catalytic machine (with >600 residues). Some examples are tRNA synthetases (cases with PDB code 1ASY, 1EIY, 1F7U, 2DLC, 2DU3, 1H4S, 1SER, 2AZX, and 2BTE), RNases and endonucleases (cases with PDB code 2C0B, 2FK6, 2IX1, and 2GJW), as well as other types of RNA-

binding proteins (cases with PDB code 1J2B, 1HVU, 2GIC, 2R7R, 3CIY, 1C9S, 2D6F, and 2IPY). On the other hand, in most of these cases, proteins do not have a globular-like shape, which can also affect sampling. Additional difficulties are found in cases in which the RNA binds to a deep protein cavity (cases with PDB code 3BSO, 2IX1, 2R7R, and 2GIC), which may also represents a challenge for docking predictions.

Post-transcriptionally modified nucleotides may represent another important challenge for protein–RNA docking. In some cases, these nucleotides are located at the interface, which may significantly affect the predictions. Docking methods should consider this as an additional variable in the quest for successful procedures.

DISCUSSION

The benchmark that we present here, composed of all possible test cases that can be found from unbound and complex structures deposited in the PDB,[29] is to our knowledge the largest collection of protein–RNA docking cases reported so far. Despite the small number of currently available test cases with available or modeled unbound RNA structure (25 of 106), this set represents a useful benchmark for developing and testing protein–RNA docking algorithms, and more importantly, it provides guidelines to further extend and analyze this dataset. Comparing this protein–RNA test set with respect to other protein–protein and protein–DNA benchmarks,[24,25] the difficulties of the protein–RNA docking field are more apparent. In general, while the mean interface RMSD between the unbound and bound forms in protein–protein benchmark 4.0 is 1.41 Å,[24] the difficulty of the protein–DNA[25] and protein–RNA benchmarks is higher, with mean interface RMSD of 3.49 and 4.51 Å, respectively. Thus, the commonly used rigid-body docking approaches for protein–protein interactions seem of limited use for the modeling of protein-nucleic acid interactions, and therefore more efforts should focus onto the development of sampling methods capable of facing the flexibility problem, as well as new scoring functions for protein–RNA interaction. While benchmark cases without available unbound RNA structure do not represent a truly realistic benchmark for the development of sampling methods, they could still be useful to optimize scoring functions, since they include some protein–RNA complex types that could not have been considered otherwise. Other large data sets of protein–RNA complexes have been reported with the purpose of developing tools for predicting RNA-contacting residues,[33] which could be useful for development of new protein–RNA docking and scoring methods.

In summary, we believe this extended protein–RNA benchmark presented here can contribute to the development and optimization of protein–RNA docking methods, including approaches for exploring the conformational flexibility of RNA in the context of protein–RNA interactions. The benchmark is available on the website http://life.bsc.es/pid/protein-rna-benchmark.

ACKNOWLEDGMENTS

LP-C is recipient of an FPU fellowship from the Spanish Ministry of Science. The authors are grateful to Joel Janin for useful comments and suggestions, and to Carles Fenollosa and the Spanish Bioinformatics Institute (INB; www.inab.org) for their help in compiling the benchmark.

REFERENCES

1. 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.
2. Andrusier N, Mashiach E, Nussinov R, Wolfson HJ. Principles of flexible protein-protein docking. Proteins 2008;73:271–289.
3. Smith GR, Sternberg MJ. Prediction of protein-protein interactions by docking methods. Curr Opin Struct Biol 2002;12:28–35.
4. 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.
5. Janin J. The targets of CAPRI Rounds 13–19. Proteins 2010;78: 3067–3072.
6. Lensink MF, Wodak SJ. Docking and scoring protein interactions: CAPRI 2009. Proteins 2010;78:3073–3084.
7. Janin J. Protein-protein docking tested in blind predictions: the CAPRI experiment. Mol Biosyst 2010;6:2351–2362.
8. Janin J, Henrick K, Moult J, Eyck LT, Sternberg MJ, Vajda S, Vakser I, Wodak SJ. CAPRI: a critical assessment of PRedicted interactions. Proteins 2003;52:2–9.
9. Wimberly BT, Brodersen DE, Clemons WM Jr, Morgan-Warren RJ, Carter AP, Vonrhein C, Hartsch T, Ramakrishnan V. Structure of the 30S ribosomal subunit. Nature 2000;407:327–339.
10. Ban N, Nissen P, Hansen J, Moore PB, Steitz TA. The complete atomic structure of the large ribosomal subunit at 2.4 A resolution. Science 2000;289:905–920.
11. Pérez-Cano L, Fernández-Recio J. Dissection and prediction of RNA-binding sites on proteins. BioMol Concept 2010;1:345–355.
12. Tuszynska I, Bujnicki JM. DARS-RNP and QUASI-RNP: new statistical potentials for protein-RNA docking. BMC Bioinform 2011;12:348.
13. Perez-Cano L, Solernou A, Pons C, Fernandez-Recio J. Structural prediction of protein-RNA interaction by computational docking with propensity-based statistical potentials. Pac Symp Biocomput 2010;2010:293–301.
14. Zheng S, Robertson TA, Varani G. A knowledge-based potential function predicts the specificity and relative binding energy of RNA-binding proteins. FEBS J 2007;274:6378–6391.
15. Chen Y, Kortemme T, Robertson T, Baker D, Varani G. A new hydrogen-bonding potential for the design of protein-RNA interactions predicts specific contacts and discriminates decoys. Nucleic Acids Res 2004;32:5147–5162.
16. Das R, Baker D. Macromolecular modeling with rosetta. Annu Rev Biochem 2008;77:363–382.
17. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 2003;331:281–299.
18. Fleishman SJ, Corn JE, Strauch EM, Whitehead TA, Andre I, Thompson J, Havranek JJ, Das R, Bradley P, Baker D. Rosetta in CAPRI rounds 13–19. Proteins 2010;78:3212–3218.

19. Huang SY, Zou X. MDockPP: a hierarchical approach for protein-protein docking and its application to CAPRI rounds 15-19. Proteins 2010;78:3096–3103.
20. Cheng TM, Blundell TL, Fernández-Recio J. pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. Proteins 2007;68:503–515.
21. Grosdidier S, Pons C, Solernou A, Fernández-Recio J. Prediction and scoring of docking poses with pyDock. Proteins 2007;69:852–858.
22. Pons C, Solernou A, Pérez-Cano L, Grosdidier S, Fernández-Recio J. Optimization of pyDock for the new CAPRI challenges: docking of homology-based models, domain-domain assembly and protein-RNA binding. Proteins 2010;78:3182–3188.
23. Hyeon C, Dima RI, Thirumalai D. Size, shape, and flexibility of RNA structures. J Chem Phys 2006;125:194905.
24. Hwang H, Vreven T, Janin J, Weng Z. Protein-protein docking benchmark version 4.0. Proteins 2010;78:3111–3114.
25. van Dijk M, Bonvin AM. A protein-DNA docking benchmark. Nucleic Acids Res 2008;36:e88.
26. Klosterman PS, Tamura M, Holbrook SR, Brenner SE. SCOR: a structural classification of RNA database. Nucleic Acids Res 2002; 30:392–394.
27. Pérez-Cano L, Fernández-Recio J. Optimal protein-RNA area, OPRA: a propensity-based method to identify RNA-binding sites on proteins. Proteins 2010;78:25–35.
28. Lewis BA, Walia RR, Terribilini M, Ferguson J, Zheng C, Honavar V, Dobbs D. PRIDB: a protein-RNA interface database. Nucleic Acids Res 2011;39:D277–D282.
29. Dutta S, Burkhardt K, Young J, Swaminathan GJ, Matsuura T, Henrick K, Nakamura H, Berman HM. Data deposition and annotation at the worldwide protein data bank. Mol Biotechnol 2009;42:1–13.
30. Eswar N, Eramian D, Webb B, Shen MY, Sali A. Protein structure modeling with MODELLER. Methods Mol Biol 2008;426:145–159.
31. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
32. Rother M, Rother K, Puton T, Bujnicki JM. ModeRNA: a tool for comparative modeling of RNA 3D structure. Nucleic Acids Res 2011;39:4007–4022.
33. Tjong H, Zhou HX. DISPLAR: an accurate method for predicting DNA-binding sites on protein surfaces. Nucleic Acids Res 2007; 35:1465–1477.

Table S1. Percentage of interface residues and nucleotides contained in the unbound protein and RNA structures

Complex PDB	Unbound Interface Residues[a]	Bound Interface Residues[b]	Interface Residue Coverage[c]	Unbound Interface Nucleotide[a]	Bound Interface Nucleotide[b]	Interface Nucleotide Coverage[c]
1ASY	188	188	100.0	42	45	93.3
1B23	125	128	97.7	23	25	92.0
1B7F	98	102	96.1	12	12	100.0
1C0A	200	200	100.0	45	47	95.7
1C9S	359	359	100.0	55	55	100.0
1DDL	51	57	89.5	7	7	100.0
1DFU	52	52	100.0	27	27	100.0
1DK1	73	73	100.0	42	42	100.0
1E7K	52	52	100.0	15	15	100.0
1E8O	40	40	100.0	13	13	100.0
1EC6	44	57	77.2	17	17	100.0
1EFW	107	107	100.0	30	33	90.9
1EIY	204	262	77.9	57	57	100.0
1EKZ	38	41	92.7	16	16	100.0
1F7U	255	255	100.0	59	59	100.0
1FEU	66	66	100.0	26	29	89.7
1FXL	99	99	100.0	9	9	100.0
1G1X	34	34	100.0	13	13	100.0
1H3E	117	124	94.4	38	38	100.0
1H4S	101	101	100.0	30	30	100.0
1HC8	57	57	100.0	22	22	100.0
1HQ1	41	41	100.0	19	19	100.0
1HVU	140	140	100.0	23	23	100.0
1J1U	110	120	91.7	34	34	100.0
1J2B	292	292	100.0	60	60	100.0
1JBR	72	72	100.0	17	17	100.0
1K1G	89	89	100.0	11	11	100.0
1K8W	100	125	80.0	18	18	100.0
1KOG	71	71	100.0	24	24	100.0
1KQ2	134	134	100.0	7	7	100.0
1LNG	58	58	100.0	35	35	100.0
1M5O	56	57	98.2	19	19	100.0
1M8V	28	35	80.0	6	6	100.0
1M8W	113	113	100.0	8	8	100.0
1MFQ	40	40	100.0	18	18	100.0

1MJI	48	48	100.0	20	20	100.0
1MMS	81	81	100.0	34	34	100.0
1N78	207	207	100.0	43	43	100.0
1OB2	128	131	97.7	24	25	96.0
1OOA	73	73	100.0	24	24	100.0
1Q2R	120	129	93.0	19	19	100.0
1QF6	205	205	100.0	42	44	95.5
1QTQ	212	217	97.7	48	49	98.0
1R3E	124	124	100.0	14	16	87.5
1RKJ	93	94	98.9	13	13	100.0
1RLG	49	49	100.0	19	19	100.0
1S03	63	63	100.0	33	33	100.0
1SER	120	120	100.0	41	42	97.6
1T0K	45	45	100.0	17	17	100.0
1T4L	56	57	98.2	28	28	100.0
1U0B	167	216	77.3	44	49	89.8
1U63	85	85	100.0	30	30	100.0
1VFG	66	66	100.0	23	23	100.0
1WNE	166	166	100.0	13	13	100.0
1WPU	56	56	100.0	7	7	100.0
1WSU	43	43	100.0	11	11	100.0
1YVP	76	83	91.6	18	19	94.7
2AD9	52	57	91.2	6	6	100.0
2ADB	55	56	98.2	6	6	100.0
2ADC	74	74	100.0	6	6	100.0
2ASB	90	90	100.0	11	11	100.0
2AZ0	74	74	100.0	25	25	100.0
2AZX	121	127	95.3	30	30	100.0
2B3J	102	102	100.0	12	12	100.0
2BGG	98	103	95.1	15	15	100.0
2BH2	188	188	100.0	28	28	100.0
2BTE	143	179	79.9	55	55	100.0
2BU1	52	52	100.0	13	13	100.0
2C0B	127	127	100.0	10	10	100.0
2CJK	107	110	97.3	8	8	100.0
2CSX	113	113	100.0	38	38	100.0
2CZJ	56	66	84.8	33	33	100.0
2D6F	63	66	95.5	25	25	100.0
2DB3	80	84	95.2	7	7	100.0

2DER	118	118	100.0	28	28	100.0
2DLC	111	111	100.0	29	29	100.0
2DRA	152	152	100.0	23	23	100.0
2DU3	156	156	100.0	50	50	100.0
2ERR	54	54	100.0	7	7	100.0
2F8K	30	31	96.8	14	14	100.0
2FK6	93	120	77.5	28	28	100.0
2FMT	100	106	94.3	35	35	100.0
2FY1	62	64	96.9	21	21	100.0
2GIC	1136	1136	100.0	90	90	100.0
2GJE	104	105	99.0	18	18	100.0
2GJW	154	154	100.0	29	29	100.0
2HGH	64	65	98.5	46	46	100.0
2HW8	80	84	95.2	31	31	100.0
2I82	112	112	100.0	16	16	100.0
2I91	144	149	96.6	21	21	100.0
2IPY	167	167	100.0	25	25	100.0
2IX1	233	233	100.0	13	13	100.0
2JPP	61	61	100.0	17	17	100.0
2NUG	112	112	100.0	35	35	100.0
2PY9	41	41	100.0	11	11	100.0
2QUX	89	89	100.0	15	15	100.0
2R7R	122	123	99.2	7	7	100.0
2R8S	91	91	100.0	39	39	100.0
2V3C	124	124	100.0	51	51	100.0
2ZKO	72	74	97.3	31	31	100.0
3BO2	56	59	94.9	20	20	100.0
3BSB	124	124	100.0	9	9	100.0
3BSO	182	182	100.0	16	16	100.0
3BSX	125	125	100.0	10	10	100.0
3BX2	120	121	99.2	9	9	100.0
3CIY	99	102	97.1	34	34	100.0

[a] Number of covered interface residues or nucleotides in the unbound protein or RNA structures.

[b] Number of interface residues or nucleotides in the complex structure.

[c] Percentage of covered interface residues or nucleotides in the unbound protein or RNA structures.

Table S2. Homology modeling of protein and RNA molecules with no available unbound coordinates

Target	Template[a]	Sequence identity[b]	RMSD target-template[c]	RMSD model-template[c]	RMSD model-target[c]
Protein					
1FEU_a	*1NJP_t*	32%	4.1	1.6	4.4
1HQ1_a	2FFH_a	44%	2.8	1.5	1.2
1LNG_a	1KVV_a	35%	4.1	0.9	4.1
1OOA_a	*3DO7_b*	55%	6.4	1.6	6.4
1RKJ_a:b*	2KRR_a	79%	10.1	0.9	10.1
2R8S_l,h	2FJF_l,h	93%	1.9	0.3	1.9
1QF6_a	1NYQ_a	43%	3.0	2.1	4.1
1VFG_a	3H38_a	36%	6.0	5.1	4.0
2DU3_a,b,c,d	2DU7_a,b,c,d	53%	3.8	3.2	4.5
1DDL_a,b,c[#]	1AUY_a,b,c	41%	0.9	0.7	1.1
1E8O_a,b	1914_a	79%	1.4	2.8	1.8
1FXL_a	3SXL_a	48%	6.9	2.5	7.4
1K1G_a*	2BL5_a	36%	11.3	3.9	10.7
1MJI_a	1IQ4_a	60%	3.0	0.4	3.2
1RLG_a	1XBI_a	53%	1.6	2.6	2.4
1S03_h	1SEI_a	50%	1.7	0.8	1.6
2B3J_a,b	1WWR_a,b	46%	1.4	2.1	2.2
2CJK_a*	1L3K_a	40%	12.7	0.9	12.8
2CSX_a	2D5B_a	44%	2.9	2.3	2.2
2D6F_a,b,c,d	1ZQ1_a,b,c,d	49%	3.8	1.4	2.9
2DB3_a*	2I4I_a	45%	18.7	2.6	15.8
2DER_a	2HMA_a	56%	2.4	1.5	2.8
2DLC_a,b	1N3L_a,b	49%	3.6	4.5	6.1
2FY1_a	1X5S_a	43%	6.4	2.5	5.5
2I82_a	1XPI_a	35%	4.4	3.3	4.9
2IPY_a*	2B3Y_a	94%	11.7	0.1	11.8

2JPP_a,b	1Y00_a,b	57%	4.9	0.9	4.9
2NUG_a*	1O0W_a	35%	14.4	2.4	14.2
2V3C_c*	3DM5_a	56%	12.8	2.6	13.0
2ZKO_a,b:c,d	2Z0A_a,b	91%	0.8	0.6	1.0
RNA					
1C0A_b	*1EFW_c*	100%	1.7	0.0	2.4
1EFW_c	*1C0A_b*	100%	1.7	0.0	1.7
1J1U_b	*2XUY_v*	70%	2.8	0.3	2.2
1J2B_c	*2ZUF_b*	72%	10.9	1.1	10.4
2DRA_b	*1VFG_d*	91%	3.0	0.5	3.0
1QF6_b	*3FOZ_c*	70%	3.8	0.0	3.9
1VFG_c	*1EIY_c*	83%	3.2	0.4	3.1
2DU3_e	*2ZZN_d*	72%	3.1	0.0	3.1

[a] PDB code of the protein or RNA template use to build the model. Cases with pseudo-unbound template, i.e. bound to a different protein or RNA molecule (see main text) are shown in italics.

[b] Sequence identity from the global alignments used for modeling (it might differ from the sequence identity based on BLAST used for the selection of templates).

[c] RMSD (Å) considering Cα atoms (for proteins) or phosphorous atoms (for RNA).

[*] Protein targets with flexible inter-domain linkers, especially difficult for modeling.

[#] The 24 Nt residues have not been considered for the RMSD calculation as they are swapped between target and template structures and give high RMSDs that do not reflect the true overall quality.

3.2.3. Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2

Thom Vreven[1#], Iain H. Moal[2#], Anna Vangone[3#], Brian G. Pierce[1], Panagiotis L. Kastritis[3], Mieczyslaw Torchala[4], Raphael Chaleil[4], Brian Jiménez-García[2], Paul A. Bates[4*], Juan Fernandez-Recio[2*], Alexander M. Bonvin[3*] and Zhiping Weng[1*]

[1]*Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, Worcester, MA 01605, USA*

[2]*Joint BSC-CRG-IRB Research Program in Computational Biology, Life Sciences Department, Barcelona Supercomputing Center, C/Jordi Girona 29, 08034 Barcelona, Spain*

[3]*Bijvoet Center for Biomolecular Research, Faculty of Science, Utrecht University, 3584CH Utrecht, The Netherlands*

[4]*Biomolecular Modelling Laboratory, The Francis Crick Institute, Lincoln's Inn Fields Laboratory, London WC2A 3LY, United Kingdom*

*Corresponding author

[#]Equal contribution

Database/
Web Servers

Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2

Thom Vreven[1,†], Iain H. Moal[2,†], Anna Vangone[3,†], Brian G. Pierce[1], Panagiotis L. Kastritis[3], Mieczyslaw Torchala[4], Raphael Chaleil[4], Brian Jiménez-García[2], Paul A. Bates[4], Juan Fernandez-Recio[2], Alexandre M.J.J. Bonvin[3] and Zhiping Weng[1]

1 - Program in Bioinformatics and Integrative Biology, *University of Massachusetts Medical School, Worcester, MA 01605, USA*
2 - Joint BSC-CRG-IRB Research Program in Computational Biology, *Life Sciences Department, Barcelona Supercomputing Center, C/Jordi Girona 29, 08034 Barcelona, Spain*
3 - Bijvoet Center for Biomolecular Research, *Faculty of Science, Utrecht University, 3584CH Utrecht, The Netherlands*
4 - Biomolecular Modelling Laboratory, *The Francis Crick Institute, Lincoln's Inn Fields Laboratory, London WC2A 3LY, United Kingdom*

Correspondence to Paul A. Bates, Juan Fernandez-Recio, Alexandre M.J.J. Bonvin and Zhiping Weng:
paul.bates@crick.ac.uk; juan.fernandez@bsc.es; a.m.j.j.bonvin@uu.nl; zhiping.weng@umassmed.edu
http://dx.doi.org/10.1016/j.jmb.2015.07.016
Edited by M. Sternberg

Abstract

We present an updated and integrated version of our widely used protein–protein docking and binding affinity benchmarks. The benchmarks consist of non-redundant, high-quality structures of protein–protein complexes along with the unbound structures of their components. Fifty-five new complexes were added to the docking benchmark, 35 of which have experimentally measured binding affinities. These updated docking and affinity benchmarks now contain 230 and 179 entries, respectively. In particular, the number of antibody–antigen complexes has increased significantly, by 67% and 74% in the docking and affinity benchmarks, respectively. We tested previously developed docking and affinity prediction algorithms on the new cases. Considering only the top 10 docking predictions per benchmark case, a prediction accuracy of 38% is achieved on all 55 cases and up to 50% for the 32 rigid-body cases only. Predicted affinity scores are found to correlate with experimental binding energies up to $r = 0.52$ overall and $r = 0.72$ for the rigid complexes.

Introduction

Protein–protein interactions are among the most important processes in biology, playing fundamental roles in the immune system, signaling pathways, and enzyme inhibition. Proteome-wide studies have revealed that most proteins interact with other proteins [1]. The experimental characterization of the structure of a protein–protein complex is, however, difficult and not always successful. To complement experimental approaches, we have developed computational techniques for the prediction of protein complexes over the years, stimulated by the CAPRI (*C*ritical *A*ssessment of *PR*edicted *I*nteractions) experiment [2]. Computational approaches for modeling protein–protein complex structures include *ab initio* docking methods [3,4], homology-based methods based on the experimental structures of similar complexes [5–11], and integrative, information-driven methods [12]. These approaches typically attempt to predict the most likely structure of a complex but are not designed to predict how strongly the proteins bind or whether they bind at all. Thus, a more complete computational description of protein–protein interaction also requires algorithms that can predict binding affinities. Although energy functions for affinity prediction and the ranking of docking poses are related, they are often developed specifically for their respective purposes and so far have shown varying and rather limited performance [13]. Example areas where scoring functions can be

Table 1. New cases in the docking benchmark 5 and affinity benchmark 2.

	Category[a]	PDB ID 1[b]	Protein 1	PDB ID 2[b]	Protein 2	I-RMSD (Å)	ΔASA^c ($Å^2$)	K_d (M)	ΔG^d (kcal/mol)	T (°C)
Rigid body										
2VXT_HL:I	A	2VXU_HL	Murine reference antibody 125-2H Fab	1JIS_A(6)	Interleukin-18	1.33	2163	5.33e-10	−12.65	
2W9E_HL:A	A	2W9D_HL	ICSM 18 Fab fragment	1OM1_A	Prion protein fragment	1.13	1677	1.3e-10	−13.49	
3EOA_LH:I	A	3EC9_LH	Efalizumab Fab fragment	3F74_A	Integrin alpha-L I domain	0.39	1272	2.2e-9	−11.81	25
3HMX_LH:AB	A	3HMW_LH	Ustekinumab Fab	1F45_AB	Interleukin-12	0.73	1841			
3MXW_LH:A	A	3MXV_LH	Anti-Shh 5E1 chimera Fab fragment	3M1N_A	Sonic Hedgehog N-terminal domain	0.48	1666	7e-9	−11.31	30
3RVW_CD:A	A	3RVT_CD	4C1 Fab	3F5V_A	DER P 1 allergen	0.50	1383	1.9e-8	−10.53	25
4DN4_LH:M	A	4DN3_LH	CNTO888 Fab	1DOL_A	MCP-1	0.81	1317	3.8e-11	−14.22	25
4FQI_HL:ABEFCD	A	4FCH_HL	CR9114 Fab	2FK0_ABCDEF	H5N1 influenza virus hemagglutinin	1.08	1459	9e-10	−12.55	30
4G6J_HL:A	A	4G5Z_HL	Canakinumab antibody fragment	4I1B_A	Interleukin-1 beta	0.61	1863	4.1e-9	−11.44	25
4G6M_HL:A	A	4G6K_HL	Gevokizumab antibody fragment	4I1B_A	Interleukin-1 beta	0.49	1673	2.9e-10	−13.01	25
4GXU_MN:ABEFCD	A	4GXV_HL	1F1 antibody	1RUZ_HIJKLM	1918H1 hemagglutinin	0.78	1830	6.2e-9	−11.2	
1JTD_B:A	EI	3QI0_A	BLIP-II	1BTL_A	TEM-1 beta-lactamase	0.44	2160	2.72e-11	−14.41	25
2A1A_B:A	ES	3UU_A	Eukaryotic translation initiation factor 2-alpha-kinase 2	1046_A	eIF2 alpha-subunit	1.35	1166			
2GAF_D:A	ER	3OWG_A	Poly(A) polymerase VP55	1VPT_A	Vaccinia protein VP39	0.69	3368	1.2e-9	−12.17	
2YVJ_A:B	ER	2YYF_A	Ferredoxin reductase BPHA4	2E4P_A	Biphenyl dioxygenase ferredoxin subunit	0.60	1377			
3A4S_A:D	EI	1A3S_A	SUMO-conjugating enzyme UBC9	3A4R_A	NFATC2-interacting protein SLD2 ubiquitin-like domain	0.72	1116	2.81e-6	−7.57	25
3K75_D:B	ER	1BPB_A	DNA polymerase beta	3K77_A	Reduced XRCC1, N-terminal domain	0.64	1195	1.1e-7	−9.49	
3LVK_AC:B	ER	3LVJ_AB	Cysteine desulfurase IscS	1DCJ_A(12)	Sulfur transferase tusA	0.81	1609	3.04e-7	−8.89	25
3PC8_A:C	ER	3PC6_A	DNA repair protein XRCC1	3PC7_A	DNA ligase III-alpha-BRCT domain	0.50	1240	1.02e-7	−9.54	
3VLB_A:B	EI	3VLA_A	EDGP	3VL8_A	Xyloglucan-specific *endo*-beta-1,4-glucanase A	0.51	2020			
4HX3_BD:A	EI	4HWX_AB	Neutral proteinase inhibitor ScNPI	1C7K_A	Zinc endoprotease	0.90	2066	6e-6	−7.41	37
4H03_A:B	ES	1CIQ_A	Iota toxin component IA	1JU_A	Alpha actin	0.68	1474			
1EXB_ABDC:EGFH	OX	1QRC_ABCD	KV beta2 protein beta-subunit	1QDV_ABCD	KV1.2 potassium channel N-terminal domain	0.62	3558			
1M27_AB:C	OX	1D4T_AB	SAP-SLAM complex	3IIA6_A	Fyn kinase SH3 domain	1.22	759	3.45e-6	−7.45	25
2GTP_A:D	OG	1GFI_A	Alpha-1 subunit guanine nucleotide-binding protein G(I)	2BV1_A	RGS1	0.54	1442			
2X9A_D:C	OR	1S62_A(8)	TolA C-terminal domain	2X9B_A	G3P TolA binding domain	1.33	1571	4.4e-6	−7.31	25
3BIW_A:E	OX	3BX_A	Neuroligin-1	2R1D_A	Neuroligin-1-beta	0.39	1151	9.7e-8	−9.41	20
3H2V_A:E	OX	3MI1_A	Vinculin tail domain	1W6_A(8)	Raver1 RRM1 domain	0.80	1263	2.21e-5	−6.31	23
3P57_AB:P	OX	3KCV_AB	MEF2A	3IO2_A	p300 TAZ2 domain	0.53	1251			
3P57_CD:P	OX	3KCV_AB	MEF2A	3IO2_A	p300 TAZ2 domain	0.74	1177			
3P57_IJ:P	OX	3KCV_AB	MEF2A	3IO2_A	p300 TAZ2 domain	0.91	1126			
4M76_A:B	OR	1C3D_A	C3D	1M1U_A	Integrin alpha-M CD11B A-domain	0.43	1046	4.5e-7	−8.66	25

Medium												
3EO1_AB:CF	A	3E00_AB	GC-1008 Fab fragment	1TGJ_AB	Transforming growth factor beta-3	1.37	1630					
3G6D_LH:A	A	3G6A_LH	CNTO607 Fab	1IKO_A(10)	Interleukin-13	1.86	1793	1.84e-11	−14.65	25		
3HI6_XY:B	A	3HI5_HL	AL-57 Fab fragment	1MJN_A	Integrin alpha-L I domain	1.65	1871	4.7e-6	−7.27			
3L5W_LH:I	A	3L7E_LH	C836 Fab	1IKO_A(11)	Interleukin-13	0.48	1138	5.4e-11	−14.01	25		
3V6Z_AB:F	A	3V6F_AB	Fab E6	3KXS_F	Capsid protein assembly domain	1.83	1922	3.3e-9	−11.57			
4FZA_A:B	ER	1UPL_A	MO25 alpha	3GGF_A	Serine/threonine-protein kinase MST4	2.04	1685					
4IZ7_A:B	EI	1ERK_A	Non-phosphorylated ERK	2LS7_A(1)	PEA-15 Death Effector Domain	1.56	1202	1.33e-7	−9.44	27		
4LW4_AB:C	ES	4LW2_AB	Cysteine desulfurase CsdA	1N7_A(8)	Cysteine desulfuration protein CsdE	1.60	1610					
3AAA_AB:C	OX	3AA7_AB	Actin capping protein	1MYO_A(30)	Myotrophin	1.78	1686	2.1e-8	−10.3	20		
3AAD_A:D	OX	1EQF_A	Double bromodomain	1TEY_A(13)	Histone chaperone ASF1	2.00	1461					
3BX7_A:C	OX	3BK8_A	Lipocalin 2	3OSK_A	CTLA-4 extracellular domain	1.63	2349	9e-9	−10.98	25		
3DAW_A:B	OX	1IJJ_A	Alpha-actin	2HD7_A(5)	Twinfilin-1 C-terminal domain	1.49	2323	2e-5	−6.41			
3R9A_AC:B	OR	1HCC_AB	Alanine-glyoxylate aminotransferase	2C0M_A	PEX5P TPR repeat domain	1.91	1926	3.5e-6	−7.44	25		
3SZK_DE:F	OX	3OCQ_AB	Met hemoglobin	2H3K_A	ISDH-N1	2.10	1263	9.01e-8	−9.45	20		
3S9D_B:A	OR	1N6U_A(15)	IFNAR2	1ITF_A(9)	IFNa2	1.69	1841	3e-9	−11.63			
4JCV_ADBC:E	OX	1VDC_ABCD	Recombinational repair protein RecR	1W3S_A	DNA repair protein RecO	1.62	1949					
Difficult												
3FN1_B:A	ER	2EDI_A(5)	UQ_con domain from NEDD8-conjugating enzyme UBE2F	2LQ7_A	NEDD8-activating enzyme E1 catalytic subunit	3.65	1867					
3H11_BC:A	ER	4JJ7_AB	Caspase-8	3H13_A	c-FLIPL protease-like domain	3.79	3169					
4GAM_AFBGCH:D	ER	1XVB_ABCDEF	Methane monooxygenase hydroxylase	1CKV_A(9)	Methane monooxygenase regulatory protein B	5.79	6671					
1RKE_A:B	OX	1SYQ_A	Vinculin head	3MY1_A	Vinculin tail	4.25	2614					
3AAD_A:B	OX	1EQF_A	Double bromodomain	1TEY_A(4)	Histone chaperone ASF1	4.37	1654					
3F1P_A:B	OX	1P97_A(9)	HIF2 alpha-C-terminal PAS domain	1X0O_A(5)	ARNT C-terminal PAS domain	2.52	1919	1.4e-6	−7.85	20		
3L89_ABC:M	OR	3L88_ABC	Ad21 fiber knob	1CKL_A	CD46 SCR1 and SCR2 domains	2.51	2167	2.84e-7	−8.93	25		

[a] Categories: antibody–antigen (A); enzyme–inhibitor (EI); enzyme–substrate (ES); enzyme complex with a regulatory or accessory chain (ER); others, G-protein containing (OG); others, receptor containing (OR); others, miscellaneous (CX).
[b] Numbers in parentheses denote the NMR model that was chosen as the unbound structure.
[c] Change in solvent-accessible surface area upon complex formation, calculated using the NACCESS program (see the materials and methods section).
[d] Calculated using $\Delta G = RT\ln K_d$, where R is the gas constant and T is the absolute temperature, with T set to 298.15 K when unknown.

improved are entropic contributions [14], solvent effects [15], and the optimal combination of terms [16].

Essential for the development of computational algorithms are training and test sets that are reliable and sufficiently large. It is computationally daunting to sift the Protein Data Bank (PDB) for structures of protein–protein complexes; the experimental conditions and accuracies of these structures vary widely and are not always straightforward to assess, and neither is the definition of the biological unit. Recognizing this, various benchmarks that attempt to collect a reliable and well-understood set of data were developed. Our docking benchmark, which after its initial development [17] has seen three updates [18–20], is widely used for developing and assessing docking methods. Key features are the availability of both the complex structure and the unbound structures of the component proteins, non-redundancy, and reliability of the data. Other benchmarks include DOCKGROUND [21], which also focuses on protein–protein interactions, and benchmarks that contain complexes of proteins with nucleic acids [22,23].

More recently, we used our protein–protein docking benchmark as a starting point for developing a structure-based affinity benchmark [24,25], which includes the entries from our docking benchmark for which experimental binding affinities were available. The affinity benchmark has been used for the development of algorithms for predicting protein–protein binding free energies, with a typical correlation coefficient of $r = 0.6$ with experimentally measured binding free energies [26–28].

In this paper, we present updates to our docking and affinity benchmarks, of which the development is tightly integrated. We added 55 new protein–protein complexes to the docking benchmark, for 35 of which experimental affinities could be found that were added to the affinity benchmark. These new additions to both benchmarks were then used, as an independent test set, to assess the performance of four docking algorithms and a large panel of affinity prediction algorithms that had been previously developed without seeing any of the new cases. This allowed us to assess the performance of docking and affinity predictions, both of which remained limited due to conformational changes, with an indication that low-affinity complexes were also more challenging to dock.

Results and Discussion

Composition

We added 55 cases to the docking benchmark (Table 1). PDB entries 3AAD and 3P57 show two and three distinct binding modes, respectively. As in the previous versions of the benchmark, the complexes that display multiple binding modes were split into different cases. This represents an increase of 31% over the previous 175 cases. We could find binding affinity data for 35 of the cases, which brought the total number of cases in the affinity benchmark to 179, a 24% increase. In Table 2, we show the composition of the updated benchmarks compared with the previous versions. The most noticeable increase is for antibody–antigen complexes: from 24 cases to 40 cases in the docking benchmark and from 19 cases to 33 cases in the affinity benchmark, which reflects a surging interest in antibody-based therapeutics.

In the previous versions of the benchmarks, some categories are underrepresented, most notably the antibody–antigen cases (14%) and difficult cases (15%), while rigid-body cases are overrepresented (68%). Although there still is overrepresentation and underrepresentation in the updated benchmark, the newly added cases do not worsen the representation of any category and achieve a more balanced composition for most categories. We examined the new cases on various properties related to size and flexibility of the component proteins, but we only found the total solvent-accessible surface area of the component

Table 2. Composition of the updated docking and affinity benchmarks (in parentheses are values for the previous versions of the benchmarks, docking version 4 and affinity version 1).

	Docking		Affinity	
	N	%	*N*	%
All	230 (175)		179 (144)	
Enzyme containing	88 (71)	38% (41%)	69 (61)	39% (42%)
Antibody–antigen	40 (24)	17% (14%)	33 (19)	18% (13%)
Others	102 (80)	45% (45%)	77 (64)	43% (45%)
Rigid body[a]	151 (119)	65% (68%)		
Medium[a]	45 (29)	20% (17%)		
Difficult[a]	34 (27)	15% (15%)		
Rigid (I-RMSD < 1.0 Å)[a]			93 (75)	52% (52%)
Flexible (I-RMSD > 1.0 Å)[a]			86 (69)	48% (48%)

[a] See the materials and methods section for definition.

proteins to be significantly smaller in docking benchmark 4 than the 55 new cases ($p = 0.05$; Kolmogorov–Smirnov test), with average total surface areas of ~24,000 Å^2 and ~29,000 Å^2, respectively. It is not clear, however, to what extent this difference reflects changes in the content of the PDB. Finally, the cases in the docking benchmark that involve nuclear magnetic resonance (NMR) structures increased from 16 cases (9%) in version 4 to 32 cases (14%) in version 5.

Performance of docking algorithms

We applied four docking algorithms (see Materials and Methods) to the new cases and their results are shown in Fig. 1a. SwarmDock [29,30], pyDock [31], and ZDOCK [32,33] are *ab initio* methods, whereas HADDOCK (*H*igh *A*mbiguity *D*riven *DOCK*ing) uses bioinformatics predictions to drive the docking [34]; in this particular case, it uses CPORT to predict interface residues [35] and Paratome [36] to identify complementarity-determining region loops of antibodies (see the materials and methods section). Overall the success rates (at least one acceptable prediction for a benchmark case) ranged between 5% and 16% for the top prediction, 20–38% for the top 10 predictions, and 40–67% for the top 100 predictions, comparable to the success rates on version 4 of the docking benchmark using SwarmDock and ZDOCK [37,38]. As expected, the success rate was much higher for the rigid-body category, with the success rates for the top 10 predictions at 31–50%, compared to 4–22% for the medium and difficult cases. The success rates also varied according to biological category, highest for enzyme containing complexes (29–41%) followed by the antibody/antigen complexes (13–38%) and finally the other complexes (5–36%).

We observed that the performances of the different docking algorithms were correlated; for 25% of the rigid-body cases, no: a single acceptable solution was found in the top 10 predictions by any of the algorithms, and for 22% cases, all four methods succeeded. These

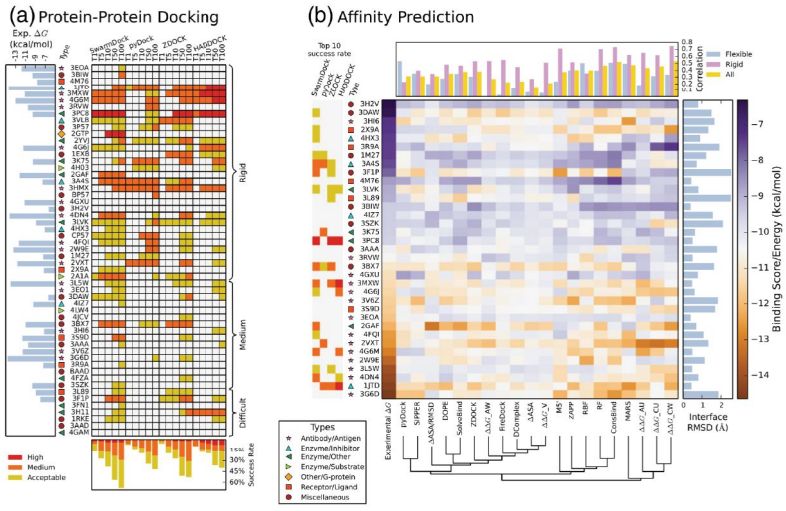


Fig. 1. (a) Performance of four docking algorithms on the new cases in the benchmarks, showing whether acceptable/medium/high-quality structures evaluated using the CAPRI criteria were present in the top 1/5/10/50/100 predictions for each case (denoted by T1, T5, T10, T50, and T100, respectively). Also shown are the overall success rates (bottom), complex type (left), and binding energy where available (far left). The complexes are ordered first by the difficulty category, then by I-RMSD. (b) Evaluation of affinity prediction methods. Complexes are ordered by increasing experimental affinities, to which the predicted affinities were fitted using linear regression in order to compare the performance of various prediction methods. The performances are grouped using a weighted average linkage agglomerative clustering algorithm (bottom). Correlations against the experimental data are shown at the top, for all the new benchmark cases and for the flexible complexes (I-RMSD $\geq$ 1.0 Å) only or for the rigid complexes (I-RMSD $<$ 1.0 Å) only. Also shown are the I-RMSD values (right), complex type (left), and the docking success rate at top 10 predictions (far left).

figures are much higher than would be expected if the complexes with correct predictions were randomly distributed among the rigid-body cases (16% and 2%, respectively). Some insight into why some interactions were inherently easier to dock than others, even within the rigid-body category, can be gleaned by focusing on the cases for which affinities are available. When all the docking algorithms failed to find an acceptable solution in the top 10 predictions, the affinity predictors also predicted weak binding energies (3EOA, 3RIW, 4M76, 3RVW, 4GXU, and 3H2V). This is either because the complexes are indeed of low affinity or due to deficiencies in the energy functions used in both docking and affinity prediction. The success rates were higher for enzyme containing and antibody–antigen complexes than for other complexes, as the latter tend to form weaker interactions.

We searched for features indicative of a successful docking outcome. We define a successful run as a benchmark case for which at least three out of four docking algorithms yielded an acceptable or better prediction in the top 100 predictions, while an unsuccessful docking run had at most one algorithm with an acceptable prediction in the top 100 predictions. We asked which features could separate the cases with successful docking runs from the cases with unsuccessful docking runs. Because a major driving force in many protein–protein docking algorithms is the desolvation of the protein components [28], we computed the buried interface area (ΔASA) upon complex formation, which is a good measure for desolvation. We further hypothesized that strong binders were easier to dock than weak binders. Indeed, ΔASA and experimentally measured binding free energy achieved a good separation of the two sets of cases with successful and unsuccessful docking runs (Fig. 2). Note that the correlation between ΔASA and the experimental binding energy is low, as reported in Fig. 1b and discussed below. These two features were individually mildly predictive of docking success (e.g., the seven strongest binders all resulted in successful docking runs), the combination of them could almost cleanly separate the successful and unsuccessful docking runs. Below the separating line, 79% docking runs were successful, and above the line, the docking performance drops to 31%. The outlier 2GAF [39] has the largest interface area of all the cases and a binding energy stronger than any of the other cases with unsuccessful docking runs. Below, we discuss this complex in more detail.

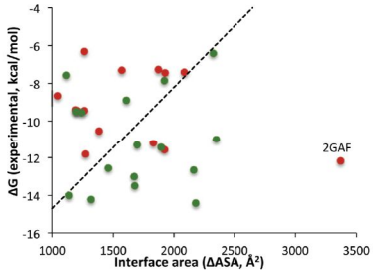


Fig. 2. Interface area *versus* experimental binding energy of the benchmark cases with successful docking runs (green; at least three docking protocols yielding acceptable predictions in the top 100) or unsuccessful docking runs (red; at most one docking protocol yielding acceptable predictions in the top 100).

Performance of affinity prediction algorithms

The change in buried surface area, ΔASA, does not correlate well with binding energy ($r = -0.16$), even for the rigid complexes [interface root-mean-square deviation (I-RMSD) < 1.0 Å, $r = -0.28$], due to complexes with large ΔASA but low affinity, such as the snpA protease/inhibitor complex (4HX3), as well as high-affinity complexes with low surface area such as the C836 (3L5W) and carlumab (4DN4) antibodies, which are highly optimized for cytokine binding. Similarly, the binding energy does not correlate highly with I-RMSD ($r = -0.24$), and only a small improvement is found using a minimal linear model combining ΔASA and I-RMSD ($r = 0.31$) [40]. We further evaluated a number of prediction methods that include the specific geometry and composition of the interaction (Fig. 1b). This yielded overall correlations of up to $r = 0.53$, with a predictive power much higher for rigid complexes, up to $r = 0.75$, than for the flexible cases, up to $r = 0.53$. The best performing methods were trained either using the first version of the affinity benchmark [25] or using changes in affinity upon mutation [41], yet these functions yielded lower correlations on the new benchmark cases than the best correlation of $r = 0.63$ previously reported for the original affinity benchmark [26,27,42]. The correlations were lower for the statistical potentials and docking scores.

For some of the complexes, the predictions were consistently poor across all methods. All methods underestimated the affinities for the antibody/hemagglutinin complex (4GXU), which features a glycosylated asparagine at the periphery of the interface; the C3D/integrin α-M complex (4M76), for which the interaction is mediated via a Ca^{2+} ion at the core of the interface; and the efalizumab/integrin α-L complex (3EOA), which is the most rigid interaction in the benchmark update (I-RMSD = 0.39 Å). On the other hand, all methods overestimated the affinities for the actin/twinfilin (3DAW), AL-57/integrin α-L (3HI6), TolA/G3P (2X9A), and HIF2/ARNT (3F1P) complexes, all of which have high flexibility, for which the energy penalty of conformational rearrangement may not be well estimated.

Highlighted case: Poly(A) polymerase VP55/ vaccinia protein VP39 (2GAF)

Figure 2 shows that the combination of experimentally measured binding energy and buried surface area forms a good indicator for a successful docking run. The complex of poly(A) polymerase VP55 and vaccinia protein VP39 (2GAF) [39], however, is a striking outlier. Only a single docking protocol was successful despite 2GAF having the largest buried surface area of all complexes and stronger binding than any of the other complexes that had at most one successful docking run. Furthermore, this complex belongs to the rigid-body category, with an I-RMSD of 0.69 Å, and we did not find co-factors or other aspects that might complicate the docking. We studied 2GAF in more detail to understand the poor docking performance. Inspection of the structure (Fig. 3) suggests that the difficulty may be related to the deep cavity of the receptor being completely filled by the ligand. To quantify this, we calculated the degree of encapsulation of a protein by its binding partner using C^{α} atoms and performed the same calculation for all the benchmark cases in Fig. 2. We found that 39 residues of the vaccinia protein VP39 are within the cavity of the poly(A) polymerase VP55 (indicated in blue in Fig. 3). This is the highest number observed in the set of proteins considered for Fig. 2; 4FQI and 3BX7 have 25 and 12 residues encapsulated, respectively, while all other proteins have fewer than 10 residues within the cavities (39 proteins show zero resides). Presumably, the tight fit seen in 2GAF renders the mouth of the energy funnel narrow, which may impact the ability of docking algorithms to find and enter the energy funnel. In addition, the tight fit may cause difficulty for grid-based methods (ZDOCK and pyDock) because even small deviations from the ideal position, resulting from the discreet rigid-body conformational parameters, may cause clashes that prevent favorable scores. Indeed, for a run with a finer rotational sampling (6° *versus* the default of 15°), ZDOCK found a high-accuracy prediction at rank 23. SwarmDock was able to find a solution in the top 5. Small conformational changes allowed by SwarmDock, which may have alleviated steric clashes at the funnel entrance, could have facilitated a smoother entry to the binding funnel. Indeed, the lowest-frequency normal mode corresponds to the opening of the binding cavity, allowing ligand insertion. In the case of HADDOCK, it was the low quality of the bioinformatics predictions for the ligand binding site (recall of 7%) that prevented the sampling of near-native solutions. Docking with center of mass or random ambiguous interaction restraints (two *ab initio* docking modes of HADDOCK) does generate acceptable solutions in the top 50 (data not shown). In general, it appears that the poor performance of the docking algorithms for 2GAF is caused by the inability to correctly sample or find the native orientation of the ligand within the receptor cavity. This makes 2GAF an exception to the general consensus in the field that failures of docking protocols are caused either by inaccuracies of the scoring functions (including explicit solvation and entropy effects) or by the difficulty of modeling protein conformational changes [43,44].

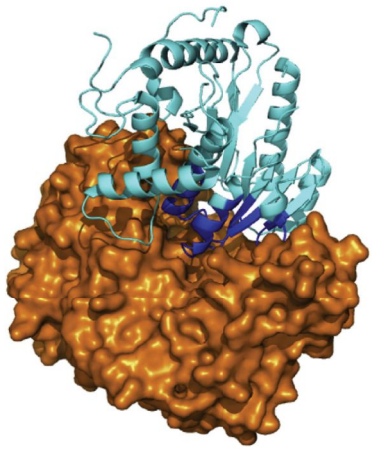


Fig. 3. Crystal structure (2GAF) of the complex of poly(A) polymerase (orange) VP55 and vaccinia protein VP39 (blue and cyan). Vaccinia protein VP39 residues that are within the poly(A) polymerase cavity are colored blue, while the residues outside the cavity are colored cyan.

Conclusions

We have presented updated versions to our widely used protein–protein docking and affinity benchmarks with 55 and 35 new entries, respectively. This represents relative increases of 31% and 24%, respectively, compared with the previous versions. The updated benchmarks have slightly improved the balance with respect to both complex types and the range of conformational changes between bound and unbound forms complete.

We analyzed the performance of four different docking methods and a comprehensive set of state-of-the-art protein–protein complex affinity prediction methods. We found that the newly added complexes provide a challenging test set for both docking and affinity prediction algorithms: Structure prediction success rates and correlations with experimentally obtained affinities are lower than reported using previous versions of the benchmark.

These updated benchmarks will aid the community in improving these algorithms and increasing our understanding of biomolecular interactions.

Materials and Methods

Benchmark construction

We collected new structures for our benchmarks from the PDB [45] using a semiautomatic pipeline. We first used the BLAST sequence homology search tool [46] to find protein–protein complexes for which the experimental structures of both the complex and the unbound component proteins were available. We also used the SACS resource [47] to collect a candidate list of antibody–antigen complexes. These complexes were then filtered using various quality criteria: (1) the complex structure needed to be determined by X-ray crystallography, the unbound structures by either X-ray crystallography or NMR; (2) the sequence identity between bound and unbound chains needed to be at least 96% with an alignment coverage larger than 80%; (3) the X-ray resolution needed to be 3.25 Å or better; and (4) chains needed to consist of at least 30 residues.

While constructing the previous versions of our docking benchmark [17–20], we deemed two complexes redundant when the pairs of interacting domains were the same at the SCOP [48] family level. Antibody–antigen complexes were considered redundant only when the SCOP families of the antigens were identical, and at least 80% of the antigen interface residues were shared between the two complexes. We used SCOPe 2.03 [49] (previously named SCOP 1.75C), which represented a limited update with respect to the 1.75 release used for the first four versions of the docking benchmark. To further compensate for the lack of SCOP coverage for the most recently solved PDB structures, we inferred their SCOP family-level assignments using the older PDB entries with identical sequences and known SCOP IDs.

We manually investigated the candidate complexes extensively, consulting the literature associated with the PDB entries. We checked whether any residues were missing or mutated in the interface (allowing such residues only if binding would not be affected) and whether co-factors that affect binding were present or compatible in both bound and unbound forms. The starting point for the manual step was the first biological assembly listed in the PDB, although in a number of cases, these were not accurate and an alternative assembly had to be used. When multiple entries were available for a complex or a component protein, we chose the entry that had the best overall structure quality. This was to some extent a subjective criterion, as we had to balance all the aforementioned features in the decision. For component proteins with NMR structures, we chose the model that had the lowest I-RMSD from the bound structure. Finally, we prepared structure files that included the fewest protein chains that correctly reflected the binding process, aligned the bound and unbound structures, and retained only those HETATM fields that we deemed biologically relevant.

We evaluated several properties from the structure files. The change in solvent-accessible surface area (ΔASA) upon complex formation was calculated using the NACCESS algorithm [50]. The I-RMSD was calculated by superposing the unbound component proteins onto their bound forms, using the C^{α} atoms for residues that had any atom within 10 Å of any atom of the binding partner. We also assessed the expected difficulty of a benchmark entry for protein–protein docking algorithms [17–20]. Complexes with I-RMSD > 2.2 Å were considered difficult, and complexes with I-RMSD < 1.5 Å were considered rigid body if their $f_{non\text{-}nat}$ [51] were <0.40. All other complexes were considered to be of medium docking difficulty.

We then used the set of complexes as a starting point for extending the structural affinity benchmark. For many entries, affinities were reported multiple times either by different groups or by using different techniques. These measurements were mostly in mutual accordance with one another, typically within 1 order of magnitude in terms of equilibrium constant. When selecting the value to include in the benchmark, priority was given to affinities reported for samples matching the sequences of the reported structures of the complexes. When this criterion could not be met or still resulted in multiple values, preference was based on sequence similarity and the measurement method. As in the first version of the affinity benchmark, most affinities were measured using surface plasmon resonance, isothermal titration calorimetry, or spectroscopic methods. The affinities of four new cases were measured using the more recent thermophoresis and bio-layer interferometry technologies. We also collected experimental conditions and additional thermodynamic and kinetic data whenever available. Affinities were measured at a pH in the 7–8 range, typically within the 20–25 °C temperature range and with an ionic strength of around 150 mM. In the context of affinity prediction, we consider complexes with I-RMSD < 1.0 Å as rigid body and the remaining complexes as flexible.

Docking algorithms

ZDOCK is an FFT-based rigid-body docking algorithm that performs a grid-based exhaustive search with a 15° or 6° rotational sampling in three-dimensional (3D) rotational space and a 1.2 Å sampling in the 3D translational space [32,33,38,52]. For each combination of the three rotational angles, the best scoring prediction in the translational space is retained, yielding 3600 or 54,000 predictions for the 15° and the 6° sampling, respectively. Here we report results obtained using the 15° sampling. We used ZDOCK version 3.0.2 that uses the IFACE [53] scoring function and the advanced 3D convolution library [54].

SwarmDock is a flexible docking method employing a population-based memetic algorithm that combines a modified particle swarm optimization global search with an adaptive random local search [29,30]. Elastic network normal mode analysis is used to model flexibility, and the algorithm simultaneously optimizes translational, quaternion, and normal coordinates, using the DComplex statistical potential as objective function [55]. The algorithm was run at the SwarmDock server [37]; swarms are initialized around ca 120 points surrounding the receptor and the algorithm was run four times from each starting point for 600 iterations. The lowest energy solutions found in each run were ranked using the centroid potential of Tobi [56] and clustered, retaining only the lowest energy member of each cluster.

pyDock [31] is a protein–protein docking protocol built upon FTDock [57], an FFT-based method that searches for

geometrically complementary rigid-body poses in the translational and rotational space. FTDock predicts 10,000 poses that are then scored using an empirical potential composed of electrostatic interaction (coulombic energy with a distance-dependent dielectric constant ε = 4.0r and charges specified by the AMBER94 force field [58], truncated to be in between 1.0 and −1.0 kcal/mol), desolvation (based on atomic solvation parameters optimized for rigid-body docking), and a limited (10%) contribution from the van der Waals energy (6–12 Lennard-Jones potential with atomic parameters from the AMBER94 force field, truncated to be below 1.0 kcal/mol).

HADDOCK [34] is a semiflexible docking protocol that uses bioinformatics predictions and biochemical/biophysical interaction data to drive the docking process. It uses CNS (*C*rystallography and *N*MR *S*ystem) [59] as its structure calculation engine. The protocol consists of three steps: (i) randomization of orientation and rigid-body docking via energy minimization driven by interaction restraints (it0), (ii) semiflexible refinement in the torsional angle space in which side-chain and backbone atoms of the interface residues are allowed to move (it1), and (iii) Cartesian dynamics refinement in explicit solvent, typically water. The final structures are clustered using the pairwise backbone ligand interface RMSD and the resulting clusters ranked according to the HADDOCK score (weighted sum of the restraint energy, the van der Waals and electrostatic energies based on OPLS parameters [60], and a desolvation energy term [61]). Note that, in the docking performance analysis presented here, no clustering was performed and individual models were selected based on their HADDOCK score.

We used the HADDOCK Web server [62], outputting 10,000/400/400 models for the three stages of the protocol. Restraints to drive the docking were derived from bioinformatics predictions by CPORT [35], except for the antibody–antigen complexes for which complementarity-determining regions identified with Paratome [36] were defined as active, and all solvent-accessible residues of the antigen were used as passive residues to define ambiguous interaction restraints to drive the docking. The predicted interfaces (and their recall and precision) used for docking are available at the SBGRid Data Bank, along with all docking decoys and HADDOCK input files from the deposited HADDOCK docking set [63].

Affinity prediction algorithms

ZAPP predicts protein–protein binding free energies using a linear combination of nine energy terms and a constant [26]. Only one term uses the unbound structures in addition to the complex structures, while the other eight terms only require the complex structure.

ConsBind is an affinity prediction method based on machine learning in which the predicted affinity is a consensus of four learners [42]: multivariate adaptive regression splines, random forest regression, radial basis function interpolation, and an M5′ regression tree. The learners were trained using 143 of the 144 affinities in the previous affinity benchmark [25] with all 108 features extracted from the bound structures using the CCharPPI Web server [64]. Information from the unbound structures was not used. The final consensus score is the arithmetic mean of the four learners.

SolveBind is a binding affinity prediction method based on the global surface model of Kastritis *et al.* [27], combining the number of atoms in the interface ($N_{AtomsINT}$) and the percentages of charged and polar residues in the non-interacting surface ($\%AA_{char}^{NIS}$ and $\%AA_{pol}^{NIS}$):

$$-\log K_d = \alpha \cdot \%AA_{pol}{}^{NS} + \beta \cdot \%AA_{char}{}^{NIS} + \gamma \cdot N_{AtomsINT} + \delta$$

with α = 0.0857, β = −0.0685, γ = 0.0262, and δ = 3.0125 (obtained after 4-fold cross-validation based on the rigid-body complexes of the previous affinity benchmark [25]). Properties of the non-interacting surface were found to correlate with affinity [13,27] and may regulate solvation and electrostatic contributions to binding affinity [27,65].

Besides the aforementioned binding affinity prediction methods developed in our groups, we also assessed the minimal affinity model of Janin (ΔASA/RMSD) [40], buried surface area (ΔASA), the DOPE [66] and DComplex [55] statistical potentials, the pyDock [31], SIPPER [67], ZDOCK [68], and FireDock [69] docking scores, as well as contact potentials (ΔΔ*G*_AW, ΔΔ*G*_AU, ΔΔ*G*_CW, and ΔΔ*G*_CU) [41] and a surface energy model (ΔΔ*G*_V) [70] derived from mutation data.

Appendix A. Supplementary data

Supplementary data (CDR definition used for docking antibody-antigen complexes with HADDOCK, predicted affinities listed by benchmark entry, experimental conditions of the affinities measurements, and the full references to the experimentally measured affinities) to this article can be found online at http://dx.doi.org/10.1016/j.jmb.2015.07.016.

The complete docking benchmark is hosted at http://zlab.umassmed.edu/benchmark, and the complete affinity benchmark at http://bmm.cancerresearchuk.org/~bmmadmin/Affinity.

Received 11 May 2015;
Received in revised form 17 July 2015;
Accepted 17 July 2015
Available online 29 July 2015

Keywords:
protein–protein complex structure;
antibody–antigen;
conformational change;
protein–protein interface;
binding free energy

†T.V., I.H.M., and A.V. contributed equally to this work.

Present address: I. H. Moal, European Molecular Biology Laboratory European Bioinformatics Institute, Hinxton, Cambridge CB10 1SD, United Kingdom.

Present address: B. G. Pierce, Institute for Bioscience and Biotechnology Research, University of Maryland, Rockville, MD 20850, USA.

Present address: P. L. Kastritis, European Molecular Biology Laboratory, Meyerhofstraße 1, 69117 Heidelberg, Germany.

Abbreviations used:
PDB, Protein Data Bank; I-RMSD, interface root-mean-square deviation; 3D, three-dimensional.

References

[1] S.J. Wodak, J. Vlasblom, A.L. Turinsky, S. Pu, Protein–protein interaction networks: The puzzling riches, Curr. Opin. Struct. Biol. 23 (2013) 941–953.
[2] J. Janin, K. Henrick, J. Moult, L.T. Eyck, M.J.E. Sternberg, S. Vajda, et al., CAPRI: A Critical Assessment of PRedicted Interactions, Proteins 52 (2003) 2–9.
[3] D.W. Ritchie, Recent progress and future directions in protein–protein docking, Curr. Protein Pept. Sci. 9 (2008) 1–15.
[4] G.R. Smith, M.J.E. Sternberg, Prediction of protein–protein interactions by docking methods, Curr. Opin. Struct. Biol. 12 (2002) 28–35.
[5] L. Lu, H. Lu, J. Skolnick, MULTIPROSPECTOR: An algorithm for the prediction of protein–protein interactions by multimeric threading, Proteins 49 (2002) 350–364.
[6] S. Mukherjee, Y. Zhang, Protein–protein complex structure predictions by multimeric threading and template recombination, Structure 19 (2011) 955–966.
[7] A. Szilagyi, Y. Zhang, Template-based structure modeling of protein–protein interactions, Curr. Opin. Struct. Biol. 24 (2014) 10–23.
[8] U. Ogmen, O. Keskin, A.S. Aytuna, R. Nussinov, A. Gursoy, PRISM: Protein interactions by structural matching, Nucleic Acids Res. 33 (2005) W331–W336.
[9] N. Tuncbag, A. Gursoy, R. Nussinov, O. Keskin, Predicting protein–protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM, Nat. Protoc. 6 (2011) 1341–1354.
[10] R. Sinha, P.J. Kundrotas, I.A. Vakser, Docking by structural similarity at protein–protein interfaces, Proteins 78 (2010) 3235–3241.
[11] T. Vreven, H. Hwang, B.G. Pierce, Z. Weng, Evaluating template-based and template-free protein–protein complex structure prediction, Brief. Bioinform. 15 (2014) 169–176.
[12] J.P.G.L.M. Rodrigues, A.M.J.J. Bonvin, Integrative computational modeling of protein interactions, FEBS J. 281 (2014) 1988–2003.
[13] P.L. Kastritis, A.M.J.J. Bonvin, Molecular origins of binding affinity: Seeking the Archimedean point, Curr. Opin. Struct. Biol. 23 (2013) 868–877.
[14] P.L. Kastritis, A.M.J.J. Bonvin, On the binding affinity of macromolecular interactions: Daring to ask why proteins interact, J. R. Soc. Interface 10 (2013) 20120835.
[15] P.L. Kastritis, K.M. Visscher, A.D.J. van Dijk, A.M.J.J. Bonvin, Solvated protein–protein docking using Kyte-Doolittle-based water preferences, Proteins 81 (2013) 510–518.
[16] I.H. Moal, M. Torchala, P.A. Bates, J. Fernandez-Recio, The scoring of poses in protein–protein docking: Current capabilities and future directions, BMC Bioinformatics 14 (2013) 286.
[17] R. Chen, J. Mintseris, J. Janin, Z. Weng, A protein–protein docking benchmark, Proteins 52 (2003) 88–91.
[18] J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin, et al., Protein–protein docking benchmark 2.0: An update, Proteins 60 (2005) 214–216.
[19] H. Hwang, B. Pierce, J. Mintseris, J. Janin, Z. Weng, Protein–protein docking benchmark version 3.0, Proteins 73 (2008) 705–709.
[20] H. Hwang, T. Vreven, J. Janin, Z. Weng, Protein–protein docking benchmark version 4.0, Proteins 78 (2010) 3111–3114.
[21] D. Douguet, H.-C. Chen, A. Tovchigrechko, I.A. Vakser, DOCKGROUND resource for studying protein–protein interfaces, Bioinformatics 22 (2006) 2612–2618.
[22] M. van Dijk, A.M.J.J. Bonvin, A protein–DNA docking benchmark, Nucleic Acids Res. 36 (2008) e88.
[23] L. Perez-Cano, B. Jiménez-García, J. Fernandez-Recio, A protein–RNA docking benchmark (II): Extended set from experimental and homology modeling data, Proteins 80 (2012) 1872–1882.
[24] P.L. Kastritis, A.M.J.J. Bonvin, Are scoring functions in protein–protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark, J. Proteome Res. 9 (2010) 2216–2225.
[25] P.L. Kastritis, I.H. Moal, H. Hwang, Z. Weng, P.A. Bates, A.M.J.J. Bonvin, et al., A structure-based benchmark for protein–protein binding affinity, Protein Sci. 20 (2011) 482–491.
[26] T. Vreven, H. Hwang, B.G. Pierce, Z. Weng, Prediction of protein–protein binding free energies, Protein Sci. 21 (2012) 396–404.
[27] P.L. Kastritis, J.P.G.L.M. Rodrigues, G.E. Folkers, R. Boelens, A.M.J.J. Bonvin, Proteins feel more than they see: Fine-tuning of binding affinity by properties of the non-interacting surface, J. Mol. Biol. 426 (2014) 2632–2652.
[28] I.H. Moal, R. Moretti, D. Baker, J. Fernandez-Recio, Scoring functions for protein–protein interactions, Curr. Opin. Struct. Biol. 23 (2013) 862–867.
[29] I.H. Moal, P.A. Bates, SwarmDock and the use of normal modes in protein–protein docking, Int. J. Mol. Sci. 11 (2010) 3623–3648.
[30] X. Li, I.H. Moal, P.A. Bates, Detection and refinement of encounter complexes for protein–protein docking: Taking account of macromolecular crowding, Proteins 78 (2010) 3189–3196.
[31] T.M.-K. Cheng, T.L. Blundell, J. Fernandez-Recio, pyDock: Electrostatics and desolvation for effective scoring of rigid-body protein–protein docking, Proteins 68 (2007) 503–515.
[32] R. Chen, L. Li, Z. Weng, ZDOCK: An initial-stage protein-docking algorithm, Proteins 52 (2003) 80–87.
[33] R. Chen, Z. Weng, A novel shape complementarity scoring function for protein–protein docking, Proteins 51 (2003) 397–408.
[34] C. Dominguez, R. Boelens, A. Bonvin, HADDOCK: A protein–protein docking approach based on biochemical or biophysical information, J. Am. Chem. Soc. 125 (2003) 1731–1737.
[35] S.J. De Vries, A.M.J.J. Bonvin, CPORT: A consensus interface predictor and its performance in prediction-driven docking with HADDOCK, PLoS ONE 6 (2011) e17695.
[36] V. Kunik, S. Ashkenazi, Y. Ofran, Paratome: An online tool for systematic identification of antigen-binding regions in antibodies based on sequence or structure, Nucleic Acids Res. 40 (2012) W521–W524.
[37] M. Torchala, I.H. Moal, R.A.G. Chaleil, J. Fernandez-Recio, P.A. Bates, SwarmDock: A server for flexible protein–protein docking, Bioinformatics 29 (2013) 807–809.

[38] B.G. Pierce, K. Wiehe, H. Hwang, B.H. Kim, T. Vreven, Z. Weng, ZDOCK server: Interactive docking prediction of protein–protein complexes and symmetric multimers, Bioinformatics 30 (2014) 1771–1773.
[39] C.M. Moure, B.R. Bowman, P.D. Gershon, F.A. Quiocho, Crystal structures of the vaccinia virus polyadenylate polymerase heterodimer: Insights into ATP selectivity and processivity, Mol. Cell 22 (2006) 339–349.
[40] J. Janin, A minimal model of protein–protein binding affinities, Protein Sci. 23 (2014) 1813–1817.
[41] I.H. Moal, J. Fernandez-Recio, Intermolecular contact potentials for protein–protein interactions extracted from binding free energy changes upon mutation, J. Chem. Theory Comput. 9 (2013) 3715–3727.
[42] I.H. Moal, R. Agius, P.A. Bates, Protein–protein binding affinity prediction on a diverse set of structures, Bioinformatics 27 (2011) 3002–3009.
[43] A. Bonvin, Flexible protein–protein docking, Curr. Opin. Struct. Biol. 16 (2006) 194–200.
[44] M. Zacharias, Accounting for conformational changes during protein–protein docking, Curr. Opin. Struct. Biol. 20 (2010) 180–186.
[45] H.M. Berman, The Protein Data Bank, Nucleic Acids Res. 28 (2000) 235–242.
[46] S. Altschul, Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.
[47] L.C. Allcorn, A.C.R. Martin, SACS—Self-maintaining database of antibody crystal structure information, Bioinformatics 18 (2002) 175–181.
[48] A.G. Murzin, S.E. Brenner, T. Hubbard, C. Chothia, SCOP: A structural classification of proteins database for the investigation of sequences and structures, J. Mol. Biol. 247 (1995) 536–540.
[49] N.K. Fox, S.E. Brenner, J.-M. Chandonia, SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures, Nucleic Acids Res. 42 (2014) D304–D309.
[50] S.J. Hubbard, J.M. Thornton, "NACCESS", Computer Program, Department of Biochemistry and Molecular Biology, University College London, 1993.
[51] R. Méndez, R. Leplae, L. De Maria, S.J. Wodak, Assessment of blind predictions of protein–protein interactions: Current status of docking methods, Proteins 52 (2003) 51–67.
[52] T. Vreven, B.G. Pierce, H. Hwang, Z. Weng, Performance of ZDOCK in CAPRI rounds 20–26, Proteins 81 (2013) 2175–2182.
[53] J. Mintseris, B. Pierce, K. Wiehe, R. Anderson, R. Chen, Z. Weng, Integrating statistical pair potentials into protein complex prediction, Proteins 69 (2007) 511–520.
[54] B.G. Pierce, Y. Hourai, Z. Weng, Accelerating protein docking in ZDOCK using an advanced 3D convolution library, PLoS ONE 6 (2011) e24657.
[55] S. Liu, C. Zhang, H. Zhou, Y. Zhou, A physical reference state unifies the structure-derived potential of mean force for protein folding and binding, Proteins 56 (2004) 93–101.
[56] D. Tobi, Designing coarse grained-and atom based-potentials for protein–protein docking, BMC Struct. Biol. 10 (2010) 40.
[57] H.A. Gabb, R.M. Jackson, M.J. Sternberg, Modelling protein docking using shape complementarity, electrostatics and biochemical information, J. Mol. Biol. 272 (1997) 106–120.
[58] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, et al., A second generation force field for the simulation of proteins, nucleic acids, and organic molecules, J. Am. Chem. Soc. 117 (1995) 5179–5197.
[59] A.T. Brünger, P.D. Adams, G.M. Clore, W.L. DeLano, P. Gros, R.W. Grosse-Kunstleve, et al., Crystallography & NMR system: A new software suite for macromolecular structure determination, Acta Crystallogr. D Biol. Crystallogr. 54 (1998) 905–921.
[60] W.L. Jorgensen, J. Tirado-Rives, The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin, J. Am. Chem. Soc. 110 (1988) 1657–1666.
[61] J. Fernandez-Recio, M. Totrov, R. Abagyan, Identification of protein–protein interaction sites from docking energy landscapes, J. Mol. Biol. 335 (2004) 843–865.
[62] S.J. De Vries, M. van Dijk, A.M.J.J. Bonvin, The HADDOCK Web server for data-driven biomolecular docking, Nat. Protoc. 5 (2010) 883–897.
[63] A. Vangone, A.M.J.J. Bonvin, HADDOCK decoys for 55 new entries in docking benchmark 5, SBGRid Data Bank, 12015, http://dx.doi.org/10.15785/SBGRID/131.
[64] I.H. Moal, B. Jiménez-García, J. Fernandez-Recio, CCharPPI Web server: Computational characterization of protein–protein interactions from structure, Bioinformatics 31 (2015) 123–125.
[65] K.M. Visscher, P.L. Kastritis, A.M.J.J. Bonvin, Non-interacting surface solvation and dynamics in protein–protein interactions, Proteins 83 (2015) 445–458.
[66] M.-Y. Shen, A. Sali, Statistical potential for assessment and prediction of protein structures, Protein Sci. 15 (2006) 2507–2524.
[67] C. Pons, D. Talavera, X. la Cruz de, M. Orozco, J. Fernandez-Recio, Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): A new efficient potential for protein–protein docking, J. Chem. Inf. Model. 51 (2011) 370–377.
[68] B. Pierce, Z. Weng, A combination of rescoring and refinement significantly improves protein docking performance, Proteins 72 (2008) 270–279.
[69] N. Andrusier, R. Nussinov, H.J. Wolfson, FireDock: Fast interaction refinement in molecular docking, Proteins 69 (2007) 139–159.
[70] I.H. Moal, J. Dapkūnas, J. Fernandez-Recio, Inferring the microscopic surface energy of protein–protein interfaces from mutation data, Proteins 83 (2015) 640–650.

Complex PDB (a)	Class (b)	Unbound PDB 1	Component 1	Unbound PDB 2	Component 2	iRMSD (Å) (c)	ΔASA (Å^2) (d)	Kd (M)	ΔG (kcal/mol) (e)	Temp. (K)	pH	Method (f)	Ref. (g)
3H2V_A:E	OX	3MYI_A	Vinculin tail domain	1WI6_A(8)	Raver1 RRM1 domain	0.8	1263	2.21e-5	-6.31	296.0	8	ITC	[1]
3DAW_A:B	OX	1IJJ_A	Alpha actin	2HD7_A(5)	Twinfilin-1 C-terminal domain	1.49	2323	2e-5	-6.41			SA	[2]
4HX3_BD:A	EI	4HWX_AB	Neutral proteinase inhibitor ScNPI	1C7K_A	Zinc endoprotease	0.9	2086	6e-6	-7.41	310.0	7.4	SA	[3]
3HI6_XY:B	A	3HI5_HL	AL-57 Fab fragment	1MJN_A	Integrin alpha-L I domain	1.65	1871	4.7e-6	-7.27		7.4	SPR	[4]
2X9A_D:C	OR	1S62_A(8)	TolA C-terminal domain	2X9B_A	G3P TolA binding domain	1.33	1571	4.4e-6	-7.31	298.0	7	SA	[5]
3R9A_AC:B	OR	1H0C_AB	Alanine-glyoxylate aminotransferase	2C0M_A	PEX5P TPR repeat domain	1.91	1926	3.5e-6	-7.44	298.0	7.5	ITC	[6]
1M27_AB:C	OX	1D4T_AB	SAP-SLAM Complex	3UA6_A	Fyn kinase SH3 domain	1.22	799	3.45e-6	-7.45	298.0	8	ITC	[7]
3A4S_A:D	EI	1A3S_A	SUMO-conjugating enzyme UBC9	3A4R_A	NFATC2-interacting protein SLD2 ubiquitin-like domain	0.72	1116	2.81e-6	-7.57	298.0	7.5	ITC	[8]
3F1P_A:B	OX	1P97_A(9)	HIF2 alpha C-terminal PAS domain	1X0O_A(5)	ARNT C-terminal PAS domain	2.52	1919	1.4e-6	-7.85	293.0	7.5	ITC	[9]
4M76_A:B	OR	1C3D_A	C3D	1M1U_A	Integrin alpha-M CD11B A-domain	0.43	1046	4.5e-7	-8.66	298.0	7.5	ITC	[10]
3LVK_AC:B	E	3LVM_AB	Cysteine desulfurase IscS	1DCJ_A(12)	Sulfurtransferase tusA	0.81	1609	3.04e-7	-8.89	298.0	7.4	SPR	[11]
3L89_ABC:M	OR	3L88_ABC	Ad21 fiber knob	1CKL_A	CD46 SCR1 and SCR2 domains	2.51	2167	2.84e-7	-8.93	298.0	7.4	SPR	[12]
4IZ7_A:B	EI	1ERK_A	Non-phosphorylated ERK	2LS7_A(1)	PEA-15 Death Effector Domain	1.56	1202	1.33e-7	-9.44	300.0	7.5	SA	[13]
3K75_D:B	ER	1BPB_A	DNA polymerase beta	3K77_A	Reduced XRCC1, N-terminal domain	0.64	1195	1.1e-7	-9.49		7.5	SA	[14]
3PC8_A:C	ER	3PC6_A	DNA repair protein XRCC1	3PC7_A	DNA ligase III-alpha BRCT domain	0.5	1240	1.02e-7	-9.54		7.5	SPR	[15]
3BIW_A:E	OX	3BIX_A	Neuroligin-1	2R1D_A	Neuroligin-1-beta	0.39	1191	9.7e-8	-9.41	293.0	7.2	ITC	[16]
3SZK_DE:F	OX	3ODQ_AB	MetHaemoglobin	2H3K_A	ISDH-N1	2.1	1263	9.01e-8	-9.45	293.0	7.5	ITC	[17]
3AAA_AB:C	OX	3AA7_AB	Actin capping protein	1MYO_A(30)	Myotrophin	1.78	1686	2.1e-8	-10.30	293.0	7	SPR	[18]
3RVW_CD:A	A	3RVT_CD	4C1 Fab	3F5V_A	DER P 1 allergen	0.5	1383	1.9e-8	-10.53	298.0	7.5	ITC	[19]
3BX7_A:C	OX	3BX8_A	Lipocalin 2	3OSK_A	CTLA-4 extracellular domain	1.63	2349	9e-9	-10.98	298.0	7.4	SPR	[20]
3MXW_LH:A	A	3MXV_LH	Anti-Shh 5E1 chimera Fab fragment	3M1N_A	Sonic Hedgehog N-terminal domain	0.48	1696	7e-9	-11.31	303.0	7.2	ITC	[21]
4GXU_MN:ABEFCD	A	4GXV_HL	1F1 antibody	1RUZ_HIJKLM	1918 H1 Hemagglutinin	0.78	1830	6.2e-9	-11.20			BLI	[22]
4G6J_HL:A	A	4G8Z_HL	Canakinumab antibody fragment	4I1B_A	Interleukin-1 beta	0.61	1893	4.1e-9	-11.44	298.0	7.4	TP	[23]
3V6Z_AB:F	A	3V6F_AB	Fab E6	3KXS_F	Capsid protein assembly domain	1.83	1922	3.3e-9	-11.57		7.4	SPR	[24]
3S9D_B:A	OR	1N6U_A(15)	IFNAR2	1ITF_A(9)	IFNa2	1.69	1841	3e-9	-11.63		7.5	SA	[25]
3EOA_LH:I	A	3EO9_LH	Efalizumab Fab fragment	3F74_A	Integrin alpha-L I domain	0.39	1272	2.2e-9	-11.81	298.0	7.4	SPR	[26]
2GAF_D:A	ER	3OWG_A	Poly(A) polymerase VP55	1VPT_A	Vaccinia protein VP39	0.69	3368	1.2e-9	-12.17		7	SPR	[27]
4FQI_HL:ABEFCD	A	4FQH_HL	CR9114 Fab	2FK0_ABCDEF	H5N1 influenza virus hemagglutinin	1.08	1459	9e-10	-12.55	303.0	7.4	BLI	[28]
2VXT_HL:I	A	2VXU_HL	Murine reference antibody 125-2H Fab	1J0S_A(6)	Interleukin-18	1.33	2163	5.33e-10	-12.65			SPR	[29]
4G6M_HL:A	A	4G6K_HL	Gevokizumab antibody fragment	4I1B_A	Interleukin-1 beta	0.49	1673	2.9e-10	-13.01	298.0	7.4	TP	[23]
2W9E_HL:A	A	2W9D_HL	ICSM 18 Fab fragment	1QM1_A	Prion protein fragment	1.13	1677	1.3e-10	-13.49		9.6	ELISA	[30]
3L5W_LH:I	A	3L7E_LH	C836 Fab	1IK0_A(11)	Interleukin-13	0.48	1138	5.4e-11	-14.01	298.0	7.4	SPR	[31]
4DN4_LH:M	A	4DN3_LH	CNTO888 Fab	1DOL_A	MCP-1	0.81	1317	3.8e-11	-14.22	298.0	7.1	SPR	[32]
1JTD_B:A	EI	3QI0_A	BLIP-II	1BTL_A	TEM-1 beta-lactamase	0.44	2180	2.72e-11	-14.41	298.0	7	SA	[33]
3G6D_LH:A	A	3G6A_LH	CNTO607 Fab	1IK0_A(10)	Interleukin-13	1.86	1793	1.84e-11	-14.65	298.0	7.3	SPR	[34]

Supporting Table 1: New interactions in version 2 of the structural affinity benchmark. Notes: (a) PDB entry with the chain codes noted AB:C to represent a complex where chains A and B make up component 1, chain C, component 2. Some of the unbound components have NMR structures; the number in parentheses refers to a model in the NMR ensemble. The processed coordinate files may be downloaded from `http://zlab.umassmed.edu/benchmark/` (b) Functional classes: A antigen/antibody; EI enzyme/inhibitor; ER enzyme complex with a regulatory or accessory chain; OR receptor containing; OX miscellaneous. (c) Root-mean-square displacement of the Cα atoms of interface residues of the two partners after the unbound and the bound interfaces have been superimposed by least-square. (d) Change in accessible surface area (ASA) between the complex and its components in bound conformation. (e) $\Delta G = -RT \ln K$; temperatures are as reported, with "ambient" or "not stated" set to 298 K. Additional kinetic and thermodynamic are available online at `http://bmm.cancerresearchuk.org/~bmmadmin/Affinity` (f) Method used to determine affinity: ITC isothermal titration calorimetry; SA spectroscopic assay; SPR surface plasmon resonance; BLI biolayer interferometry; TP thermophoresis; ELISA enzyme-linked immunosorbent assay. (g) Primary citation as listed below. Additional references and corroborating data are available online at `http://bmm.cancerresearchuk.org/~bmmadmin/Affinity`

Complex	ΔG (kcal/mol)	ΔASA	DComplex	DOPE	pyDock	SIPPER	FireDock	DDG_CU	DDG_CW	DDG_AU	DDG_AW	DDG_V	MARS	RF	RBF	M5'	ConsBind	SolveBind	ZAPP	IRAD	ZRANK	ZDOCK	ΔASA/RMSD
1M27	−7.45	799	−9.27	−499	−36.20	2.72	−21.15	−9.56	−8.72	−11.29	−10.84	−10.56	−8.61	−8.01	−6.25	−7.75	−7.65	6.30	−8.23	−138.84	−82.13	644.2	6.61
3PC8	−9.54	1240	−14.67	−770	−28.96	6.09	−60.48	−3.87	−6.28	−9.05	−15.47	−20.36	−9.62	−8.82	−9.33	−7.91	−8.92	5.55	−9.46	−215.42	−103.27	1108.4	10.39
3BIW	−9.41	1191	−9.66	−681	−18.71	0.66	−18.34	−0.13	−3.14	−3.21	−7.69	−12.41	−7.87	−7.83	−6.42	−10.72	−8.21	7.25	−9.69	−120.07	−69.44	516.8	10.37
3F1P	−7.85	1919	−18.28	−1521	−44.79	14.04	−50.20	−2.11	−4.61	−9.28	−18.02	−29.53	−8.87	−10.64	−14.22	−17.52	−12.82	8.93	−10.89	−233.74	−87.55	1299.3	9.57
3A4S	−7.57	1116	−11.30	−643	−37.00	2.85	−29.43	−10.00	−11.97	−9.54	−14.03	−11.81	−10.28	−7.99	−6.82	−6.08	−7.79	5.58	−8.03	−177.25	−109.72	902.0	9.46
2X9A	−7.31	1571	−17.03	−945	−21.02	1.64	−82.03	−10.36	−18.63	−16.73	−27.46	−22.05	−11.31	−12.30	−11.89	−14.39	−12.47	7.24	−10.15	−222.07	−116.17	1302.4	9.06
3L89	−8.93	2167	−13.68	−1446	−21.95	8.68	−45.87	−3.72	−6.21	−10.31	−19.61	−27.60	−9.57	−9.93	−8.90	−10.35	−9.69	10.19	−12.05	−218.08	−127.60	1238.1	10.51
4HX3	−7.41	2086	−16.70	−1372	−29.60	4.68	−41.56	−7.70	−3.84	−7.07	−18.38	−23.96	−8.61	−10.73	−13.63	−10.35	−10.83	10.03	−12.80	−221.45	−118.88	1235.4	12.65
3H2V	−6.31	1263	−9.53	−712	−13.64	−0.29	2.30	−1.86	−3.44	−6.85	−9.37	−13.35	−8.58	−8.22	−8.89	−7.75	−8.36	5.99	−7.45	−114.67	−58.88	672.5	9.81
3AAA	−10.30	1686	−13.84	−1123	−43.31	0.45	−60.77	−6.54	−7.19	−15.89	−20.28	−19.73	−11.36	−11.28	−10.45	−8.67	−10.44	7.01	−9.90	−244.97	−131.67	1257.6	8.68
4IZ7	−9.44	1202	−10.75	−717	−26.59	5.03	−37.98	−7.02	−5.37	−10.06	−13.26	−19.02	−8.74	−8.82	−9.29	−9.36	−9.05	6.76	−9.98	−160.50	−81.94	881.3	6.84
3SZK	−9.45	1263	−11.16	−1128	−19.30	1.27	−7.58	−12.35	−13.07	−9.10	−15.60	−16.95	−9.16	−9.36	−11.27	−11.47	−10.31	6.34	−9.53	−135.20	−69.74	704.4	7.07
1JTD	−14.41	2180	−19.50	−1910	−48.40	12.04	−53.71	−8.29	−17.27	−19.10	−29.19	−29.32	−13.64	−12.42	−12.65	−11.34	−12.51	8.88	−13.51	−234.03	−99.64	1729.0	14.05
3S9D	−11.63	1841	−15.52	−1082	−32.77	10.16	−56.36	−11.61	−14.45	−10.30	−16.87	−25.54	−11.23	−10.68	−12.58	−11.34	−11.46	8.14	−11.41	−186.75	−92.79	1262.9	9.27
2GAF	−12.17	3368	−19.44	−1949	−41.41	7.73	−48.72	−13.37	−18.50	−16.22	−36.96	−39.10	−9.00	−12.29	−11.42	−8.67	−10.34	11.58	−10.01	−292.91	−150.15	1622.1	18.09
3K75	−9.49	1195	−12.43	−1011	−36.98	2.73	−51.09	−3.08	−10.34	−5.05	−12.58	−18.86	−9.77	−8.73	−11.27	−7.75	−9.38	6.10	−10.46	−142.06	−69.75	817.7	9.94
3R9A	−7.44	1926	−15.28	−959	−17.93	−2.26	−30.09	2.71	1.23	−8.52	−24.01	−25.80	−7.54	−10.59	−7.36	−10.22	−8.92	7.47	−9.05	−180.61	−105.48	1007.7	9.59
3LVK	−8.89	1609	−11.46	−902	−43.02	11.12	−43.88	−4.24	−5.90	−13.58	−25.16	−17.49	−9.85	−9.23	−10.09	−7.75	−9.23	6.25	−7.27	−235.72	−122.70	1524.6	11.10
3BX7	−10.98	2349	−21.48	−1627	−44.46	10.31	−75.21	−7.48	−10.34	−7.30	−19.53	−37.80	−10.80	−11.75	−12.00	−14.39	−12.24	9.57	−11.00	−268.18	−129.98	1697.1	11.20
3DAW	−6.41	2323	−19.23	−1519	−28.45	9.75	−107.16	−9.62	−11.75	−16.94	−25.66	−33.67	−10.66	−12.17	−12.60	−10.42	−11.46	8.47	−11.28	−319.66	−161.42	1650.0	11.15
4M76	−8.66	1046	−9.68	−499	−8.15	0.39	−10.81	−4.46	−7.06	−6.59	−14.49	−8.76	−7.93	−7.18	−5.88	−6.08	−6.77	5.18	−7.31	−94.73	−44.33	577.8	9.76
3G6D	−14.65	1793	−16.16	−1238	−57.10	15.33	−61.34	−8.25	−12.70	−16.31	−19.21	−25.67	−12.58	−12.60	−12.90	−16.35	−13.61	9.14	−11.30	−278.12	−141.04	1423.9	9.09
3L5W	−14.01	1138	−11.06	−895	−31.50	4.88	−48.21	−12.59	−14.66	−14.92	−17.09	−18.31	−10.50	−10.15	−9.53	−13.83	−11.00	7.87	−9.67	−174.44	−78.23	1098.7	10.03
3V6Z	−11.57	1922	−15.10	−1746	−43.74	16.11	−62.69	−4.34	−10.90	−13.36	−19.06	−29.90	−13.08	−12.28	−10.79	−14.39	−12.63	9.61	−14.09	−223.07	−111.87	1344.4	9.58
4G6J	−11.44	1893	−16.10	−1463	−27.70	2.01	−60.16	−13.75	−15.62	−19.27	−26.18	−23.01	−11.21	−10.90	−13.65	−8.67	−11.11	9.18	−12.03	−215.20	−104.43	1282.9	12.66
4G6M	−13.01	1673	−14.66	−1488	−18.48	1.07	−45.44	−12.94	−10.05	−22.64	−22.54	−21.79	−9.71	−10.78	−13.89	−12.39	−11.69	8.78	−12.98	−232.79	−117.86	1249.9	12.05
2VXT	−12.65	2163	−21.28	−1457	−46.26	4.89	−76.01	−19.35	−20.28	−25.99	−35.82	−31.97	−13.03	−12.44	−11.99	−12.59	−12.51	9.30	−12.55	−318.30	−160.11	1687.4	11.31
3EOA	−11.81	1272	−12.46	−892	−33.64	5.73	−43.08	−6.65	−8.06	−9.68	−13.72	−17.16	−10.31	−9.57	−9.54	−9.36	−9.69	7.36	−9.66	−167.36	−82.97	1148.2	10.68
3HI6	−7.27	1871	−16.41	−1391	−51.66	8.12	−51.64	−3.84	−9.37	−11.10	−22.10	−26.79	−13.50	−10.39	−9.57	−10.35	−10.95	9.15	−11.88	−235.06	−113.86	1455.8	9.38
3RVW	−10.53	1383	−13.96	−1174	−29.63	7.10	−9.45	−6.13	−7.48	−7.18	−8.46	−18.17	−8.68	−9.94	−13.56	−9.36	−10.39	9.05	−10.61	−169.67	−74.46	1233.7	10.93
2W9E	−13.49	1677	−16.05	−1471	−37.82	5.58	−35.00	−5.81	−9.27	−9.17	−14.17	−18.96	−10.54	−10.90	−15.97	−12.39	−12.45	9.38	−10.82	−234.13	−98.88	1353.8	10.30
3MXW	−11.31	1696	−18.63	−1405	−44.13	5.25	−68.89	−11.60	−10.56	−20.70	−15.80	−20.11	−11.06	−11.22	−12.15	−13.83	−12.06	8.79	−10.17	−300.39	−150.62	1503.3	12.15
4DN4	−14.22	1317	−13.77	−779	−23.25	8.64	−74.64	−11.24	−15.81	−15.04	−24.80	−20.32	−11.03	−11.01	−10.02	−13.83	−11.47	7.57	−10.96	−235.64	−112.60	1236.6	9.99
4FQI	−12.55	1459	−13.65	−1220	−46.91	8.34	−63.13	−8.36	−11.33	−11.22	−23.76	−22.07	−12.64	−11.43	−12.32	−7.88	−11.07	7.78	−11.25	−201.69	−93.54	1405.9	9.66
4GXU	−11.20	1830	−11.56	−1089	−5.61	−2.46	−6.52	−6.36	−9.91	−5.61	−20.73	−21.64	−8.88	−10.18	−8.83	−10.22	−9.53	9.00	−7.83	−105.82	−48.78	798.7	12.02

Supporting Table 2: Predicted affinities. See main text for descriptions of the scoring functions.

PDB code*	Light Chain L1	Light Chain L2	Light Chain L3	Heavy Chain H1	Heavy Chain H2	Heavy Chain H3
3G6D_LH	26, 27, 28, 29, 30, 31, 32, 33	45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55	88, 89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111
3L5W_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37	49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61	99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
3V6Z_BA	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40	52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62	95, 96, 97, 98, 99, 100, 101, 102, 103	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112
4G6J_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107
4G6M_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37	49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61	99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
2VXT_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 52A, 53, 54, 55, 56, 57, 58, 59	94, 95, 96, 101, 102
3EOA_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
3HI6_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
3RVW_CD	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95	27, 28, 29, 30, 31, 32, 33, 34, 35, 36	48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
2W9E_LH	27, 28, 29, 30, 31, 32, 33	45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55	88, 89, 90, 91, 92, 93, 94, 95	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105
3MXW_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 55A, 56, 57, 58, 59	94, 95, 96, 97, 98, 99, 100, 100A, 100K, 101, 102
4DN4_LH	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57	90, 91, 92, 93, 94, 95, 96, 97, 98	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108
3EO1_AB	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57	90, 91, 92, 93, 94, 95, 96, 97	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
3HMX_LH	27, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108
4FQI_LH	27, 27A, 27B, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 95A, 95B, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 52A, 53, 54, 55, 56, 57, 58, 59	94, 95, 96, 97, 98, 99, 100, 100A, 100B, 100C, 100D, 101, 102
4GXU_NM	27, 27A, 27B, 28, 29, 30, 31, 32, 33, 34	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56	89, 90, 91, 92, 93, 94, 95, 95A, 95B, 95C, 96	27, 28, 29, 30, 31, 32, 33, 34, 35	47, 48, 49, 50, 51, 52, 52A, 53, 54, 55, 56, 57, 58, 59	94, 95, 96, 97, 98, 99, 100, 100A, 100B, 100C, 100D, 100E, 100F, 100G, 100H, 100I, 101, 102

* PDB code and chain identifiers are reported, in the order light/heavy chain.

Supporting Table 3: List of antibodies complementarity-determining regions (CDRs) in antibody/antigen complexes used to drive docking simulation in HADDOCK; data calculated by PARATOME webserver. For the other type of complexes, restraints to drive the docking were derived from bioinformatics predictions by CPORT (`http://haddock.chem.uu.nl/services/CPORT/`). The full list of the CPORT predicted residues is reported in the online set of HADDOCK decoys provided.

References

[1] Lee, J. H., Rangarajan, E. S., Yogesha, S. D. & Izard, T. (2009). Raver1 interactions with vinculin and RNA suggest a feed-forward pathway in directing mRNA to focal adhesions. Structure 17, 833–842.

[2] Paavilainen, V. O., Oksanen, E., Goldman, A. & Lappalainen, P. (2008). Structure of the actin-depolymerizing factor homology domain in complex with actin. J. Cell Biol. 182, 51–59.

[3] Trillo-Muyo, S., Martinez-Rodriguez, S., Arolas, J. L. & Gomis-Ruth, F. X. (2013). Mechanism of action of a janus-faced single-domain protein inhibitor simultaneously targeting two peptidase classes. Chem. Sci. 4, 791–797.

[4] Shimaoka, M., Kim, M., Cohen, E. H., Yang, W., Astrof, N., Peer, D., Salas, A., Ferrand, A. & Springer, T. A. (2006). AL-57, a ligand-mimetic antibody to integrin LFA-1, reveals chemokine-induced affinity up-regulation in lymphocytes. Proc. Natl. Acad. Sci. U.S.A. 103, 13991–13996.

[5] Lorenz, S. H., Jakob, R. P., Weininger, U., Balbach, J., Dobbek, H. & Schmid, F. X. (2011). The filamentous phages fd and IF1 use different mechanisms to infect Escherichia coli. J. Mol. Biol. 405, 989–1003.

[6] Fodor, K., Wolf, J., Erdmann, R., Schliebs, W. & Wilmanns, M. (2012). Molecular requirements for peroxisomal targeting of alanine-glyoxylate aminotransferase as an essential determinant in primary hyperoxaluria type 1. PLoS Biol. 10, e1001309.

[7] Chan, B., Lanyi, A., Song, H. K., Griesbach, J., Simarro-Grande, M., Poy, F., Howie, D., Sumegi, J., Terhorst, C. & Eck, M. J. (2003). SAP couples Fyn to SLAM immune receptors. Nat. Cell Biol. 5, 155–160.

[8] Sekiyama, N., Arita, K., Ikeda, Y., Hashiguchi, K., Ariyoshi, M., Tochio, H., Saitoh, H. & Shirakawa, M. (2010). Structural basis for regulation of poly-SUMO chain by a SUMO-like domain of Nip45. Proteins 78, 1491–1502.

[9] Cardoso, R., Love, R., Nilsson, C. L., Bergqvist, S., Nowlin, D., Yan, J., Liu, K. K., Zhu, J., Chen, P., Deng, Y. L., Dyson, H. J., Greig, M. J. & Brooun, A. (2012). Identification of Cys255 in HIF-1 as a novel site for development of covalent inhibitors of HIF-1/ARNT PasB domain protein-protein interaction. Protein Sci. 21, 1885–1896.

[10] Bajic, G., Yatime, L., Sim, R. B., Vorup-Jensen, T. & Andersen, G. R. (2013). Structural insight on the recognition of surface-bound opsonins by the integrin I domain of complement receptor 3. Proc. Natl. Acad. Sci. U.S.A. 110, 16426–16431.

[11] Dahl, J. U., Radon, C., Buhning, M., Nimtz, M., Leichert, L. I., Denis, Y., Jourlin-Castelli, C., Iobbi-Nivol, C., Mejean, V. & Leimkuhler, S. (2013). The sulfur carrier protein TusA has a pleiotropic role in Escherichia coli that also affects molybdenum cofactor biosynthesis. J. Biol. Chem. 288, 5426–5442.

[12] Cupelli, K., Muller, S., Persson, B. D., Jost, M., Arnberg, N. & Stehle, T. (2010). Structure of adenovirus type 21 knob in complex with CD46 reveals key differences in receptor contacts among species B adenoviruses. J. Virol. 84, 3189–3200.

[13] Callaway, K., Abramczyk, O., Martin, L. & Dalby, K. N. (2007). The anti-apoptotic protein PEA-15 is a tight binding inhibitor of ERK1 and ERK2, which blocks docking interactions at the D-recruitment site. Biochemistry 46, 9187–9198.

[14] Cuneo, M. J. & London, R. E. (2010). Oxidation state of the XRCC1 N-terminal domain regulates DNA polymerase beta binding affinity. Proc. Natl. Acad. Sci. U.S.A. 107, 6805–6810.

[15] Beernink, P. T., Hwang, M., Ramirez, M., Murphy, M. B., Doyle, S. A. & Thelen, M. P. (2005). Specificity of protein interactions mediated by BRCT domains of the XRCC1 DNA repair protein. J. Biol. Chem. 280, 30206–30213.

[16] Arac, D., Boucard, A. A., Ozkan, E., Strop, P., Newell, E., Sudhof, T. C. & Brunger, A. T. (2007). Structures of neuroligin-1 and the neuroligin-1/neurexin-1 beta complex reveal specific protein-protein and protein-Ca2+ interactions. Neuron 56, 992–1003.

[17] Krishna Kumar, K., Jacques, D. A., Pishchany, G., Caradoc-Davies, T., Spirig, T., Malmirchegini, G. R., Langley, D. B., Dickson, C. F., Mackay, J. P., Clubb, R. T., Skaar, E. P., Guss, J. M. & Gell, D. A. (2011). Structural basis for hemoglobin capture by Staphylococcus aureus cell-surface protein, IsdH. J. Biol. Chem. 286, 38439–38447.

[18] Takeda, S., Minakata, S., Koike, R., Kawahata, I., Narita, A., Kitazawa, M., Ota, M., Yamakuni, T., Maeda, Y. & Nitanai, Y. (2010). Two distinct mechanisms for actin capping protein regulation–steric and allosteric inhibition. PLoS Biol. 8, e1000416.

[19] Chruszcz, M., Pomes, A., Glesner, J., Vailes, L. D., Osinski, T., Porebski, P. J., Majorek, K. A., Heymann, P. W., Platts-Mills, T. A., Minor, W. & Chapman, M. D. (2012). Molecular determinants for antibody binding on group 1 house dust mite allergens. J. Biol. Chem. 287, 7388–7398.

[20] Schonfeld, D., Matschiner, G., Chatwell, L., Trentmann, S., Gille, H., Hulsmeyer, M., Brown, N., Kaye, P. M., Schlehuber, S., Hohlbaum, A. M. & Skerra, A. (2009). An engineered lipocalin specific for CTLA-4 reveals a combining site with structural and conformational features similar to antibodies. Proc. Natl. Acad. Sci. U.S.A. 106, 8198–8203.

[21] Maun, H. R., Wen, X., Lingel, A., de Sauvage, F. J., Lazarus, R. A., Scales, S. J. & Hymowitz, S. G. (2010). Hedgehog pathway antagonist 5E1 binds hedgehog at the pseudo-active site. J. Biol. Chem. 285, 26570–26580.

[22] Tsibane, T., Ekiert, D. C., Krause, J. C., Martinez, O., Crowe, J. E., Wilson, I. A. & Basler, C. F. (2012). Influenza human monoclonal antibody 1F1 interacts with three major antigenic sites and residues mediating human receptor specificity in H1N1 viruses. PLoS Pathog. 8, e1003067.

[23] Blech, M., Peter, D., Fischer, P., Bauer, M. M., Hafner, M., Zeeb, M. & Nar, H. (2013). One target-two different binding modes: structural insights into gevokizumab and canakinumab interactions to interleukin-1. J. Mol. Biol. 425, 94–111.

[24] Watts, N. R., Vethanayagam, J. G., Ferns, R. B., Tedder, R. S., Harris, A., Stahl, S. J., Steven, A. C. & Wingfield, P. T. (2010). Molecular basis for the high degree of antigenic cross-reactivity between hepatitis B virus capsids (HBcAg) and dimeric capsid-related protein (HBeAg): insights into the enigmatic nature of the e-antigen. J. Mol. Biol. 398, 530–541.

[25] Piehler, J., Roisman, L. C. & Schreiber, G. (2000). New structural and functional aspects of the type I interferon-receptor interaction revealed by comprehensive mutational analysis of the binding interface. J. Biol. Chem. 275, 40425–40433.

[26] Li, S., Wang, H., Peng, B., Zhang, M., Zhang, D., Hou, S., Guo, Y. & Ding, J. (2009). Efalizumab binding to the LFA-1 alphaL I domain blocks ICAM-1 binding via steric hindrance. Proc. Natl. Acad. Sci. U.S.A. 106, 4349–4354.

[27] Gershon, P. D. & Khilko, S. (1995). Stable chelating linkage for reversible immobilization of oligohistidine tagged proteins in the BIAcore surface plasmon resonance detector. J. Immunol. Methods 183, 65–76.

[28] Dreyfus, C., Laursen, N. S., Kwaks, T., Zuijdgeest, D., Khayat, R., Ekiert, D. C., Lee, J. H., Metlagel, Z., Bujny, M. V., Jongeneelen, M., van der Vlugt, R., Lamrani, M., Korse, H. J., Geelen, E., Sahin, O., Sieuwerts, M., Brakenhoff, J. P., Vogels, R., Li, O. T., Poon, L. L., Peiris, M., Koudstaal, W., Ward, A. B., Wilson, I. A., Goudsmit, J. & Friesen, R. H. (2012). Highly conserved protective epitopes on influenza B viruses. Science 337, 1343–1348.

[29] Wu, C., Sakorafas, P., Miller, R., McCarthy, D., Scesney, S., Dixon, R. & Ghayur, T. (2003). IL-18 receptor beta-induced changes in the presentation of IL-18 binding sites affect ligand binding and signal transduction. J. Immunol. 170, 5571–5577.

[30] Antonyuk, S. V., Trevitt, C. R., Strange, R. W., Jackson, G. S., Sangar, D., Batchelor, M., Cooper, S., Fraser, C., Jones, S., Georgiou, T., Khalili-Shirazi, A., Clarke, A. R., Hasnain, S. S. & Collinge, J. (2009). Crystal structure of human prion protein bound to a therapeutic antibody. Proc. Natl. Acad. Sci. U.S.A. 106, 2554–2558.

[31] Fransson, J., Teplyakov, A., Raghunathan, G., Chi, E., Cordier, W., Dinh, T., Feng, Y., Giles-Komar, J., Gilliland, G., Lollo, B., Malia, T. J., Nishioka, W., Obmolova, G., Zhao, S., Zhao, Y., Swanson, R. V. & Almagro, J. C. (2010). Human framework adaptation of a mouse anti-human IL-13 antibody. J. Mol. Biol. 398, 214–231.

[32] Das, A., Sweet, R., Tsui, P., Bethea, D., Wu, S., Kang, J. & Baker, A. (2009). Anti- mcp-1 antibodies, compositions, methods and uses. WO Patent App. PCT/US2008/068,696 .

[33] Lim, D., Park, H. U., De Castro, L., Kang, S. G., Lee, H. S., Jensen, S., Lee, K. J. & Strynadka, N. C. (2001). Crystal structure and kinetic analysis of beta-lactamase inhibitor protein-II in complex with TEM-1 beta-lactamase. Nat. Struct. Biol. 8, 848–852.

[34] Wu, S. J., Luo, J., O'Neil, K. T., Kang, J., Lacy, E. R., Canziani, G., Baker, A., Huang, M., Tang, Q. M., Raju, T. S., Jacobs, S. A., Teplyakov, A., Gilliland, G. L. & Feng, Y. (2010). Structure-based engineering of a monoclonal antibody for improved solubility. Protein Eng. Des. Sel. 23, 643–651.

3.3 New methods for structural protein-protein complex prediction

The aim of protein-protein docking methods is to predict the complex structure starting from the structure of the unbound partners. The nature of this problem is very complex and intractable by more physically accurate methods such as molecular dynamics. From the late 70s of the past century, several docking methods have been proposed, with very promising results. But community-wide experiments such as the CAPRI international contest have demonstrated the limitations of the current methods. In order to overcome many of the limitations of the current protein-protein docking methods, a new method called LightDock is proposed in this thesis. LightDock has been developed with the purpose in mind of being an experimenting platform where current and future developments could easily be prototyped. LightDock is a scoring function-agnostic framework, i.e. users can incorporate their own function, which is written in Python and can make use of normal mode analysis, precomputed ensembles and local non-gradient minimization, in order to model the protein flexibility.

Manuscripts presented in this section:

Brian Jiménez-García, Jorge Roel-Touris, Miquel Vidal and Juan Fernández-Recio (2016, Manuscript) **"LightDock: A framework for multi-scoring function flexible protein-protein docking"**

LightDock: A framework for multi-scoring function flexible protein-protein docking

Brian Jiménez-García[1] <bjimenez@bsc.es>, Jorge Roel-Touris[1] <jroeltou@bsc.es>, Miquel Vidal <miquel.vidal@bsc.es>[1], Juan Fernández-Recio[1] <juanf@bsc.es>

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

Abstract

Here we present LightDock, a novel protein-protein docking framework based on the Glowworm Swarm Optimization (GSO) algorithm. The framework is written in the Python programming language and allows the users to incorporate their own scoring function. By using the swarm information provided by the different agents during the simulation, the algorithm tries to converge to the multiple energetic minima. A key point is that the user scoring function is encoded in the fitness function that describes the search space to be optimized by the algorithm. The framework can use either full-atom or coarse-grained representation, and includes the use of normal mode analysis to introduce backbone flexibility in the interacting molecules. Simulations using the framework can be chained using different force-fields at each independent simulation. We have tested the

usability and the performance of the framework using two different scoring functions, DFIRE and pyDock. The results of LightDock in the Protein-protein Benchmark 5.0 using these two scoring functions, and a *posteriori* rescoring step using pyDock scoring energy in order to combine both DFIRE and pyDock, show similar results compared with other state-of-the-art docking algorithms. The LightDock framework is highly versatile, with many options that can be further developed and optimized by the users: It can accept any user-defined scoring function, can use local gradient-free minimization, the simulation can be restrained from the beginning to focus on user-assigned interacting regions, and it has support for the use of pre-calculated conformers for both receptor and ligand. LightDock source code can be freely downloaded from http://life.bsc.es/pid/lightdock.

1. Introduction

Protein-protein interactions are fundamental to virtually every cellular process, such as protein expression regulation, cell-cycle control, or immune response, among others (Eisenberg et al. 2000). With the avalanche of genomic sequences and data on messenger RNA expression that scientists are dealing with in the post-genomic era, situating protein-protein interactions in their functional network context is of vital importance to understand the physiological processes performed within the cell context.

According to the most recent data in Interactome3D (http://interactome3d.irbbarcelona.org), there is available structural data at atomic resolution (experimentally determined or based on a close homologous complex) for only a small portion (between 1 and 7%) of the estimated number of protein-protein complexes in human (Stumpf et al. 2008; Venkatesan et al. 2009). This is mostly due to current technical limitations in the structural determination of protein-protein complexes by X-ray crystallography or Nuclear Magnetic Resonance (*NMR*). Although new advances in low-resolution structural techniques, such as as cryo-electron microscopy (cryo-EM) or SAXS, are very promising (Doerr and Allison 2015; Bai et al. 2015, Spilotros and Svergun 2014), it is unlikely that this huge gap between the number of estimated protein-protein interactions and the number of complexes deposited in the Protein Data Bank can be overcome based only on experimental methods. In recent years, many efforts have focused on the development of computational tools for protein-protein complex structure prediction that can complement experimental methods. This is the so-called docking problem.

From a technical point of view, the docking problem presents two main challenges: the efficient sampling of the conformational and orientation space in search of near-native structures (sampling), and the identification of such near-native structures among the many models generated (scoring) (Moal and Bates 2010). In many cases, the applicability of a given scoring function is strongly dependent on the sampling approaches used. The

widely used Fast-Fourier Transform (FFT) based methods can efficiently generate geometrically complementary rigid-body docking poses, and their appearance gave rise to a significant development in the protein-protein docking field. Pioneer methods were MOLFIT (Katchalski-Katzir et al. 1992) and FTDock (Gabb et al. 1997), which incorporated an extra grid for taking into account electrostatics contributions, but other methods have included desolvation based on atom-contacts as in ZDOCK (Chen and Weng 2002) or pairwise interaction potentials as in PIPER (Kozakov et al. 2006). The main advantage of FFT-based methods is their high computer speed, which can be even further accelerated by using graphics processing units (GPU) (Ritchie and Venkatraman 2010; Ritchie et al. 2010). However, one of the major limitations of this approach is the difficulty in the inclusion of new scoring schemes within the FFT approach, since any extra atomic pairwise scoring function needs to be defined as one or more additional 3D grids, which usually comes at the expense of a high computational cost. Thus very often, new developed scoring functions are more efficiently used as part of an additional scoring step outside the FFT framework. This is the case of pyDock scoring function, with ASA-based desolvation optimized for protein-protein association, van der Waals potential, and Coulombic electrostatics (Cheng et al. 2007). Another major limitation of the FFT grid-based methods is that they cannot explicitly consider conformational flexibility. Thus, soft-potentials have to be used instead due to the restrictions of the rigid-body docking paradigm. Other types of docking methods, like PatchDock (Duhovny et al. 2002), use surface

representation of the molecules and geometric hashing to find geometric complementarity, in a rigid-body docking framework. In any of these cases, the use of new scoring functions is strongly limited by the sampling method, and an explicit conformational flexibility search can only be applied as an additional step.

The alternative to the FFT grid-based approaches is the use of explicit representation of the interacting proteins, at atomic or coarser-grained level, in search of the global energy minimum in order to identify the native orientation. The explicit representation of the molecules facilitates the use of a larger variety of scoring functions, which can represent better the energy of the association process. However, the computational cost of conformational search in atomistic representation is high, so in practice, these methods are often used to perform a first search in which the molecules are rigid. Very often, the initial rigid-body docking search is followed by an additional flexible refinement step, within the same atomistic framework. The ICM-DISCO docking method pioneered the application of global energy Monte-Carlo optimization and side-chain refinement (Fernandez-Recio 2003). In RosettaDock refinement step, a side-chain minimization using a rotamer library is performed (Schueler-Furman et al. 2005), but new versions of the software include more refinements. In HADDOCK approach, several flexible refinement steps are performed using molecular dynamics, with increasing levels of flexibility. In order to lower the computational costs, the number of degrees of freedom of the conformational search is dramatically reduced by using distance restraints from

experimental data (Dominguez et al. 2003). These types of docking methods can also include flexibility during the entire search phase, usually by applying a coarser-grained representation of the interacting proteins. In ATTRACT docking method (Zacharias 2003; May and Zacharias 2008), a reduced protein model and the first non-trivial normal modes of the anisotropic network model (ANM) are used. SwarmDock (Li et al. 2010; Moal and Bates 2010) method is based in the particle swarm optimization (PSO) algorithm, which makes use of normal modes to simultaneously optimize docking poses with an electrostatics and van der Waals scoring function.

On the scoring side, the development of new functions that can be independently applied to different sets of docking models generated by a variety of docking methods is an active area of research (Moal, Moretti, et al. 2013). Recently reported approaches include an asymmetric potential designed specifically for antibody–antigen docking (Brenke et al. 2012) or the integration of bioinformatics and experimental information (Schneidman-Duhovny et al. 2012). A large benchmark of more than a hundred scoring functions on their capabilities of rescoring docking poses generated by the SwarmDock method was recently reported (Moal et al. 2013). However, as above shown, the use of new scoring functions in docking has been traditionally limited by the type of sampling method. On the one hand, grid-based docking search methods have difficulties in efficiently including energy-based scoring functions. On the other hand, molecular dynamics, minimization or Monte-Carlo sampling

methods usually are linked to a specific force-field and cannot easily accept new scoring schemes. It is thus necessary the development of new sampling schemes in docking that can use multi-scale representation of the proteins, accept flexibility at different degrees, and accommodate a large variety of new scoring functions.

In this context, Swarm intelligence (SI) is a family of the artificial intelligence algorithms inspired by emergent systems in nature, which can perform a more efficient search in a complex space, quite independently on the scoring function to optimize. Basically, those algorithms make use of simple agents that interact locally in a decentralized way, and whose interactions lead to complex emergent patterns or systems in nature, e.g. fish schooling or termite mounds. SI algorithms have many interesting properties: heuristics are generally simpler because there is no need of central control, they are inspired by nature metaphors which makes their parameters easy to understand by humans and, finally, they tend to be easily scalable as more agents can be added at any time. SI algorithms, such as PSO and some variants have been applied to protein-ligand docking, e.g. Tribe-PSO as implemented in AutoDock 3.05 (Chen et al. 2006) and PSO@AutoDock (Namasivayam and Günther 2007), as well as to protein-protein docking, e.g. SwarmDock (Li et al. 2010). A related algorithm is Glowworm Swarm Optimization (GSO) (Krishnanand and Ghose 2008), a bio-inspired algorithm from the SI family, which is based in the concept that in nature, glowworms are being attracted by other mates depending on the

quantity of emitted light. This metaphor is used by the GSO algorithm for simultaneously capturing multiple local optima in multimodal functions. Each agent in the algorithm, a glowworm, carries out a quantity of *luciferin* which encodes the actual fitness of the position of the agent in the explored search space. The algorithm has been applied to many different problems (Krishnanand and Ghose 2009; Liao et al. 2011; Huang and Zhou 2011), but not explicitly to the protein-protein docking. Here in this work we demonstrate that GSO is a good candidate as a global optimization mechanism for capturing the multiple local and global energetic minima of the docking energetic landscape, independently from the force-field used. GSO has some advantages over PSO (Krishnanand and Ghose 2008). First, GSO was initially designed for capturing multimodal local and global minima or maxima, while PSO was only intended for capturing global maxima or minima. This property is especially relevant when exploring the protein-protein docking energetic landscape, which tends to be very noisy. Moreover, in GSO the number of captured minima or maxima is proportional to the number of defined agents, while this is not true in PSO, which poses a major drawback in systems which are required to scale. On the contrary, the major drawback of GSO over PSO is the computation time, which tends to be one order of magnitude higher.

The development of a new protein-protein docking method based on the GSO algorithm is justified by its robust performance in very noisy environments, as well as by its scalability (an

interesting property in high-performance computing architectures). The new method has been devised as a protein-protein docking framework for fast-prototyping and testing of new scoring functions.

2. Theory: a new framework for protein-protein docking

2.1. LightDock: GSO algorithm applied to protein-protein docking

The agents in the GSO algorithm are defined as glowworms which carry a luminescent quantity called *luciferin*. At each step of the simulation, the quantity of luciferin l depends on the evaluation of the complex energy by the user-defined scoring S function in the actual search space x and the previous value of the luciferin based on the trajectory of the given glowworm (equation 1). Decay of the quantity of luciferin is controlled by the ρ variable, and γ represents the enhancement constant, i.e. how much affects the actual evaluation of the energy in the luciferin quantity.

$$l_i(t+1) = (1-\rho)\cdot l_i(t) + \gamma \cdot S\big(x_i(t+1)\big) \qquad (1)$$

In LightDock, these parameters are defined by default as: $\rho = 0.4$, $\gamma = 0.6$, initial luciferin $l(t=0)$ = 5.0. Each glowworm g_i initially represents a specific position in the translational and rotational space of the ligand (equation 2), where t_x, t_y and t_z

are the components of the vector $v_{origin-ligand_{center}}$ and q_w, q_x, q_y and q_z are the components of the quaternion that represents the ligand rotation in the four-dimensional quaternions space. The use of quaternions in the framework is justified by their smaller physical memory footprint compared to three-dimensional rotation matrices (4 float variables instead of 9, 3x3), and by the absence of the known gimbal lock of sampling based on Euler angles or polar coordinates, and renormalization problems (Shoemake 1985).

$$g_i = [t_x, t_y, t_z, q_w, q_x, q_y, q_z] \quad (2)$$

In addition, the framework has the capability of using the anisotropic network model (ANM) (Atilgan et al. 2001; Doruker et al. 2000) to introduce a certain degree of backbone flexibility during the protein-protein binding process. In this case, each glowworm agent represents, in addition to a translation/rotation ligand position, the extent of deformation along each normal mode for both receptor, nr, and the ligand, nl, in the optimization vector (3). The number or non-trivial normal modes is customizable for the receptor, R, and the ligand, L.

$$g_i = [t_x, t_y, t_z, q_w, q_x, q_y, q_z, nr_{1\ldots R}, nl_{1..L}] \quad (3)$$

ANM is implemented in the LightDock framework via the ProDy Python library (Bakan et al. 2011). The ANM model is calculated on the $C\alpha$ atoms of the backbone of both receptor and ligand and then extended to the rest of atoms for each residue. By default,

we considered the first ten non-trivial normal modes ($R = L = 10$) because of the good compromise between the percentage of recovery in the interface as seen in (Moal and Bates 2010a) (55% in ten normal modes versus 44% for the first five non-trivial normal modes) and the computation time required.

2.2. Initial receptor/ligand models (glowworms)

Each independent simulation in a LightDock run will contain a fixed number of receptor/ligand models (glowworm group) in which the randomly defined ligand positions will cover a given region around the receptor. The initial ligand positions can show a certain overlapping between some of the glowworm groups so that taking all together they will cover all regions around the receptor. The use of independent simulations from different glowworm groups has important advantages. First, only the glowworms within the same group can see each other. In this way, the agents can only sample a localized region of the receptor and thus can maximize the acquired information by the swarm in this specific region of the search space. Second, it makes the algorithm to be embarrassingly parallel, with no need of communication between parallel executions and facilitates the optimal execution of the algorithm in high-performance computing architectures or small clusters. Finally, it offers the opportunity to the users to avoid regions that are known in advance not to be likely involved in binding, i.e. transmembrane

domains, as opposed to many FFT-based methods where this filtering has to be performed *a posteriori*.

In order to guarantee reproducibility of the results, the random number generator used in LightDock always saves the initial seed. This seed can also be given to the framework as an argument. The initial conditions of the algorithm, receptor/ligand starting models and the position of the glowworm groups over the receptor surface, are calculated on basis of the random number generator seed.

The setup of the initial glowworm groups is as follows. Initially, a fixed number of initial group centers N_s (by default 400) are defined around the receptor, by using the spiral method (Rakhmanov et al. 1994), and are projected using a ray-tracing technique to find the closest atom from the receptor at a distance of the maximum radius of the ligand. Fig. 1 shows the initial disposition of the 400 initial group centers in the 1VFB complex (only receptor is shown).

For each initial group center, glowworms are defined by randomly positioning the ligands (by default 300) so that their center of coordinates are placed within a 10 Å sphere from the given group center. If NMA representation is considered, deformational extents for receptor and ligand are randomly generated from a Gaussian distribution with $\mu = 4.0$ and $\sigma = 3.0$.

The number of initial glowworm group centers N_s is given as an input of the application. To guarantee a correct sampling over the

surface, a certain density of these centers is needed. Our research shows that, ideally, this number should be higher than 1.5 when the radius of the sphere used to randomly position the ligands is 10 Å. The framework calculates the density automatically and if the density is not met, warns the user about the possible under-sampling.

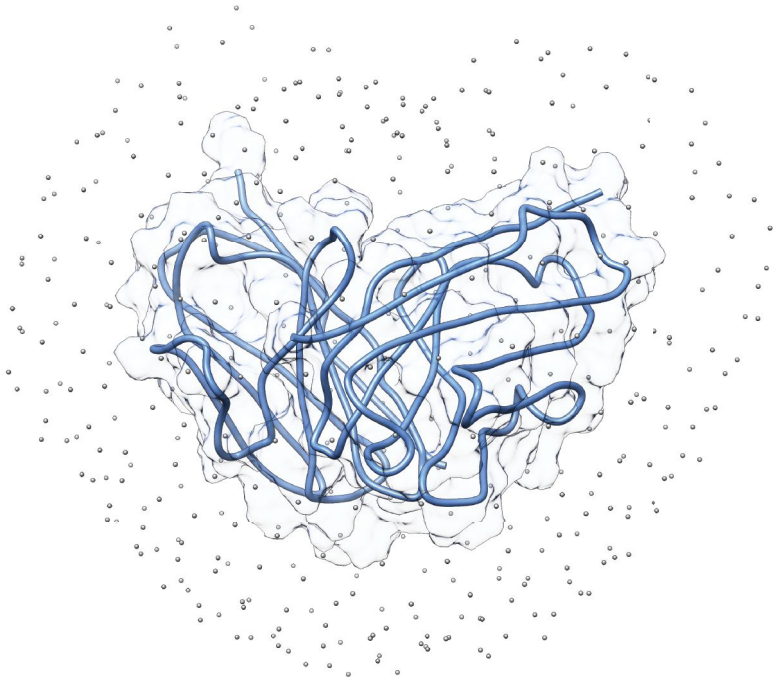


Figure 1. The Fv fragment of the anti-hen egg white lysozyme antibody D1.3 in its free and antigen-bound forms (PDB code 1VFB). The receptor (in blue) is surrounded by 400 initial glowworm group centers.

LightDock framework can also support the use of pre-calculated ligand poses generated by FTDock (Gabb et al. 1997). Each docking pose is represented in FTDock by a translation vector and three Euler angles. The top *N* FTDock solutions are classified based on their translation and rotation values into one (or more) LightDock glowworm group. If the number of solutions *N* provided by FTDock is not sufficient to populate the different LightDock glowworm groups, new starting positions are randomly generated within the selected glowworm group. However, in our tests the use of pre-calculated FTDock poses had a lower performance (data not shown), when compared to completely random poses. This lower performance could be explained by the bias introduced in the population of glowworm poses, which perhaps makes that many energetic wells are not correctly explored.

Regarding the ANM representation, the initial distribution of the normal modes for each glowworm agent is based on a Gaussian distribution ($\mu = 4.0$ and $\sigma = 3.0$) to assure an acceptable internal bonded geometry. To minimize over-fitting, these values were tested against a small set of only four complexes of the Protein-Protein Benchmark 3.0 (Hwang et al. 2008) that were classified as rigid in the mentioned benchmark. Intuitively, a relatively large value of σ is required to ensure some variability, but μ centered in 0.0 does not seem to be a good choice according to our tests (data not shown), since the range of the normal mode extents generated is not sufficient to recover unbound-bound conformational changes. Other methods as ATTRACT (de Vries

and Zacharias 2013) and SwarmDock (Moal and Bates 2010a) reported similar values for the deformational extents.

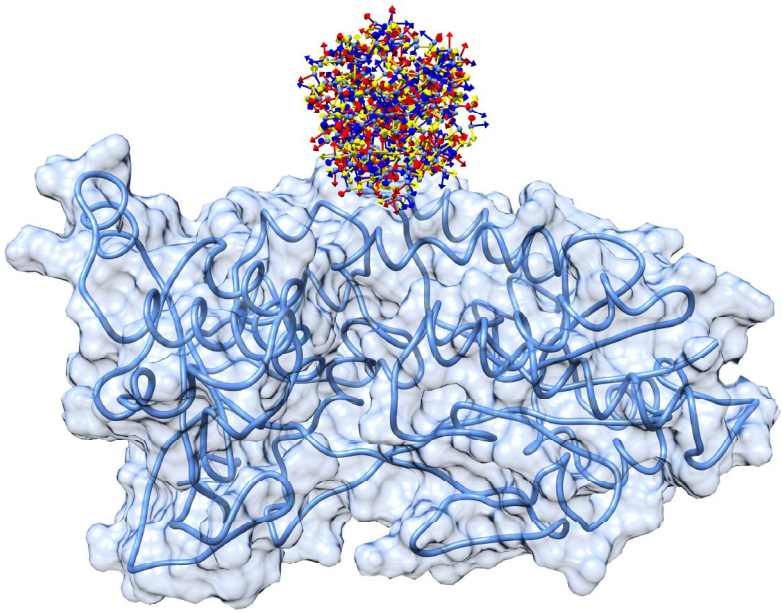


Figure 2. The Adrenoxin Reductase-Adrenoxin Complex (PDB code 1E6E). Receptor is shown in blue, 300 ligand positions for a given glowworm group are represented using a three-axis arrows model (red, yellow and blue represent the x, y and z orthogonal axis), showing their initial translation and rotation.

2.3. GSO sampling

As above described, sets of initial receptor/ligand putative models (glowworms) are defined for their use in independent simulations. Each of these glowworms will move towards the

best-scoring (luciferin) neighbor glowworm based on the probability calculated as described in Equation 4 (Krishnanand and Ghose 2008),

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (4)$$

where neighbor glowworms of glowworm g_i (N_i) are defined by the variables called *maximum number of neighbors* (by default 5) and *vision range* (by default 5.0 Å distance). The distance in the search space between two receptor/ligand models (glowworms) used to update this list of neighbors (N_i) is computed as that between the centers of the minimum ellipsoids of the ligands (translation and rotation of the receptors does not vary). We also tried a different definition of the distance between two ligands as the root mean squared distance (RMSD) computed for the seven representative points of each pose (using the six poles and the center of the minimum volume ellipsoid that contains a given ligand pose by its g_i position vector). However, this metric had a higher computational cost and was too sensitive to the pose rotation due to the RMSD calculation, and as a consequence, agents very far in the translational space were erroneously seen as neighbors. The neighborhood of each glowworm as well as the vision range are dynamically updated at each step depending on some constants as described elsewhere (Krishnanand and Ghose 2008), so that the vision range can vary up to a *maximum vision range* (by default 20.0). The parameter β, which limits the association of neighbors in the GSO algorithm, is set by default to 0.16.

The evolution from one ligand pose (initial glowworm g_1) towards another one (target glowworm g_2) is composed of two different movements: a translation in the translational space and a rotation in the space of the quaternions. Within the translational space, a new pose p_2 will be built from the initial pose p_1 by applying a number in the interval (0, 1) as defined in the *translation step* variable (by default 0.5) to the translation vector t_{12} between g_1 and g_2. As for the rotational movement, the movement in quaternion space is calculated using the spherical linear interpolation (SLERP) (Morrison and Jack 1992) between the quaternion components of g_1 and g_2 with a default step of 0.5. In the case of using the ANM representation, a simple interpolation in Euclidean space with a step of 0.5 will be included in the ligand movement. All of these step values can be changed by the users.

2.4. Scoring functions

The movement of the different agents though the search space is driven by the fitness of the function S of the quantity of luciferin. The GSO algorithm is able to optimize the function as long as the agents are uniformly distributed along the search space. In that sense, the optimization method is independent from the search space and makes the strategy valid for any scoring function used. LightDock framework offers the possibility to add new scoring functions abstracting the way of how molecules are considered. Thanks to a piece of software called adapter, the users can specify their own protein models (full atoms or coarse

grained). In the movement step, the model will be rotated and translated and there will be a new class coded by the user, the evaluation module, the one in charge of evaluating the fitness of the scoring function. To demonstrate the possibilities of the framework regarding further extension, two scoring functions have been implemented and tested: DFIRE (Liu et al. 2004) and a faster version of the pyDock scoring function (Cheng et al. 2007), which uses contact solvation from unbound precomputed ASA values (upcoming publication).

2.5. Clustering of final docking poses

The resulting models from each independent simulation (by default 300) are merged and clustered. Clustering plays an essential role in the final success rate independently of the scoring function applied, since it removes redundant models. Here we are presenting two different clustering approaches, which results in a huge decrease of the final number of solutions, while preserving the overall hits/solutions ratio.

The first approach consists in a hierarchical clustering (Ward 1963) based on the implementation of the algorithm in the SciPy library (Jones et al. 2001). After a sufficient number of steps, the algorithm converges to a set of clusters which clearly represent energetic minima (see Figure 3). But the degree of convergence varies from complex to complex, and isolated predicted poses can often appear. Moreover, for each of the identified minima, we

are only interested in the most energetically favorable receptor/ligand models (in Figure 3, the glowworms with more negative scoring for each of the groups). In this hierarchical clustering, only the groups of receptor/ligand models with more population (i.e. with a higher convergence degree) are selected. Then, for each of the identified clustered groups, the top *Z* ranked solutions by energy are selected where *Z* is proportional to the number of glowworms in the group found.

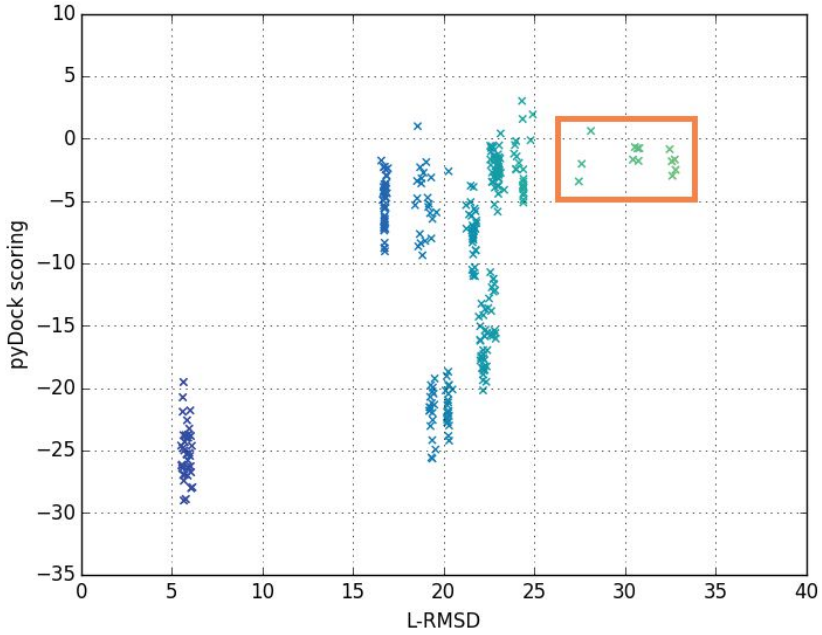


Figure 3. Ligand RMSD with respect to the reference vs. pyDock energy scoring of the different glowworms for a given cluster at step 100 of the simulation. Within the square are the poses that will be removed using the hierarchical clustering

approach because their group has no relevant population or they represent noisy receptor/ligand models.

We also applied a simple clustering procedure based on the Basic Sequential Algorithmic Scheme (BSAS) algorithm (Theodoridis and Koutroumbas 1999), which is devised to be able to discard redundant poses with a ligand RMSD below 4 Å. First, the best docking pose, in terms of energy, is identified establishing the first sub-cluster. Then, and sequentially, the rest of receptor-ligand complexes are structurally evaluated against the poses already clustered. If they are 4 Å below than any of the cluster representatives in terms of ligand RMSD, they will be included in that cluster, otherwise they will establish a new one.

After testing both procedures on the Protein-Protein Docking Benchmark version 5.0, the BSAS-based clustering method was established as the default clustering strategy due to a better ratio of near native solutions versus the number of total predictions.

2.6. Re-scoring and the use of multiple scoring functions

Simulations on the same complex using different scoring functions could be combined in order to capture different near-native predictions. One direct way of combining the results from any two or more different simulations is to merge the resulting models and score them by using the same function, in order to normalize the models. With this purpose, before evaluating the

combined results, we tested the predictive success of LightDock when the models (obtained either with pyDock or DFIRE scoring functions) were re-scored by pyDock scoring function. The main rationale behind this rescoring process is that pyDock is one of the top performing scoring functions, as found in the scorers round of the CAPRI community-wide experiment (Pallara et al. 2013; Lensink et al. 2016), and it is sufficiently fast to not become an overhead in the total computation time of LightDock.

2.7. Benchmarking and evaluation of results

The performance of the LightDock protocol was evaluated on the Protein-Protein Docking Benchmark 5.0 (Vreven et al. 2015) with a total of 230 cases. The predictive success rate was defined as the percentage of cases in which at least one near-native solution was found within the top *N* solutions (*N* = 10, 100), as ranked according to the corresponding scoring function. Near-native solutions were defined as those ones with a ligand RMSD < 10 Å with respect to the ligand position in the reference structure (when receptor molecules are superimposed).

3. Evaluation of LightDock performance

3.1. Overall predictive performance of LightDock

The predictive performance of LightDock was tested on the recently reported Protein-Protein Docking Benchmark 5.0, composed of a total of 230 complexes. LightDock was run using default parameters (see Theory section), and two scoring functions (DFIRE and pyDock) were independently tested. For each docking case, LightDock generated a total of 120,000 poses, which were clustered as described in the Theory section. The final number of docking models ranged between 600 (PDB 1CLV) and 6,387 (PDB 1DE4) when using pyDock scoring function, and between 748 (PDB 1CLV) and 6,713 (PDB 1AKJ), when using DFIRE.

As can be seen in Figure 4, the use of pyDock scoring function within LightDock showed better success rates for the top 10 docking solutions than when using the DFIRE scoring function. The performance of LightDock with pyDock scoring function is only slightly worse than that of pyDock applied on FTDock docking models, as in pyDock server (Jiménez-García et al. 2013). For the top 100 success rates, this difference in performance between pyDock and DFIRE scoring functions is higher, and interestingly, LightDock with pyDock is even slightly better than standard FTDock+pyDock (suppl. Figure S1).

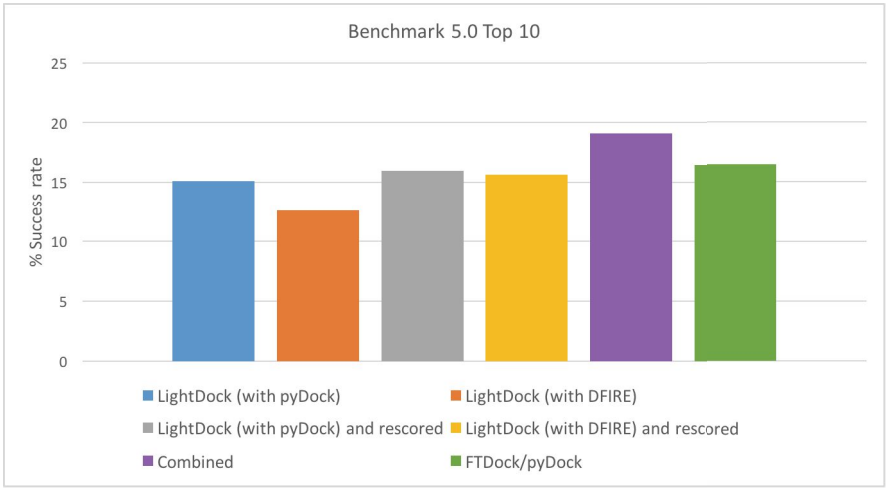


Figure 4 Predictive success rates for LightDock on the Protein-Protein Docking Benchmark 5.0, n = 230. Success rates for the top 10 docking models are shown for: LightDock with pyDock scoring function (blue), LightDock with DFIRE scoring function (orange), LightDock-pyDock after rescoring by pyDock (grey), LightDock-DFIRE after rescoring by pyDock (yellow), final combination of LightDock-pyDock and LightDock-DFIRE after rescoring with pyDock (purple). For comparison, the performance of the standard FTDock/pyDock protein-protein docking protocol is shown (green).

3.2. Combination of results from different scoring functions

Given that LightDock framework supports the use of a variety of scoring functions, we have evaluated the advantages of combining the results obtained by different scoring functions.

Figure 5A,B shows the number of successful cases obtained by LightDock on the protein-protein benchmark 5.0 when using independently either the pyDock or DFIRE scoring functions, considering the top 10 or top 100 docking models, and how many cases were predicted by both scoring functions. As Figure 5 shows, a significant number of cases were only successful with one of the scoring functions but not the other. This important degree of complementarity in the results suggested the possibility of combining the models from the two LightDock simulations as a way to increase the number of successful cases. Interestingly, the number of successful cases after pyDock rescoring increased for both methods. The improvement was more evident for LightDock/DFIRE models, which after re-scoring with pyDock, achieved success rates similar to LightDock/pyDock. This shows that the differences in the success rates when using pyDock or DFIRE as scoring function during the search mainly depended on the scoring of the resulting models, and not on the search algorithm itself, given that sampling, even with DFIRE, was able to provide good models that were later identified by pyDock re-scoring.

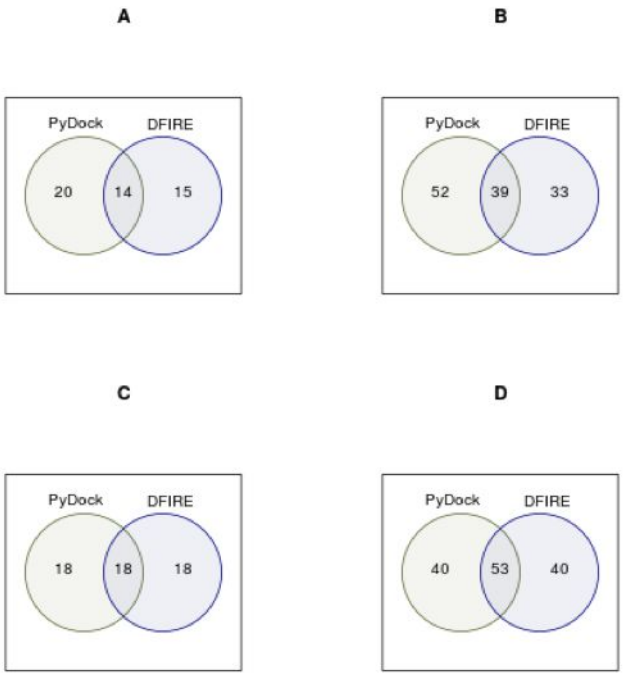


Figure 5. Venn diagrams showing the number of successful cases for LightDock with pyDock (green) or DFIRE2 (blue) scoring functions. A. Successful cases when considering the top 10 docking models. B. Successful cases when considering the top 100 docking models. C. Top 10 successful cases after rescoring with pyDock. D. Top 100 successful cases after rescoring with pyDock.

When combining the docking models obtained from the two LightDock versions, and subsequent re-scoring by pyDock,

global success rates (19.1% for top 10; 44.0% for top 100) slightly improved with respect to the individual simulations, and were even better than those of standard pyDock on FTDock models (Figure 4). However, we should note that this small improvement comes at the expense of doubling the computational cost, since two independent simulations are needed.

4. Discussion

4.1. The use of ANM significantly improves LightDock predictive success

The use of ANM-based flexibility is expected to provide better predicted models. To evaluate this, we tested a version of LightDock that did not use the ANM model, being thus completely rigid-body sampling, on a heterogeneous set of 30 complexes (6 rigid, 17 low-flexible, 5 medium-flexible and 2 flexible) from the Protein-Protein Benchmark 5.0. The success rates were much worse (10.0% for top 10; 20.0% for top 100; as compared to 16.7% and 26.7%, respectively, when using ANM). Interestingly, the analysis by category of flexibility shows that there is no difference between the use of ANM in the rigid-body class (16.7% for top 10 and top 100, using or not ANM), but the difference of success rate comes from an improvement in the low-flexible and medium-flexible categories for both top 10 and

top 100 results. This improvement provided by ANM is similar as that reported for other state of the art methods that use normal mode analysis.

4.2. LightDock is more efficient in flexible cases

It is interesting to analyze whether the performance of LightDock (with different scoring functions) depends on the flexibility of the interacting proteins. For that, we have classified the cases, according to the l-RMSDCα between the unbound and bound states (as defined in the Protein-Protein Docking Benchmark 5.0), in the following categories: rigid (l-RMSDCα < 0.5 Å), low-flexible (0.5 Å < l-RMSDCα < 1.0 Å), medium-flexible (1.0 Å < l-RMSDCα < 2.0 Å), flexible (2.0 Å < l-RMSDCα < 3.0 Å) and highly-flexible (l-RMSDCα > 3.0 Å). LightDock with pyDock performs better in the low-flexible cases (Figure 6), while the standard FTDock+pyDock protocol was more successful in the rigid cases. The introduction of the ANM representation is

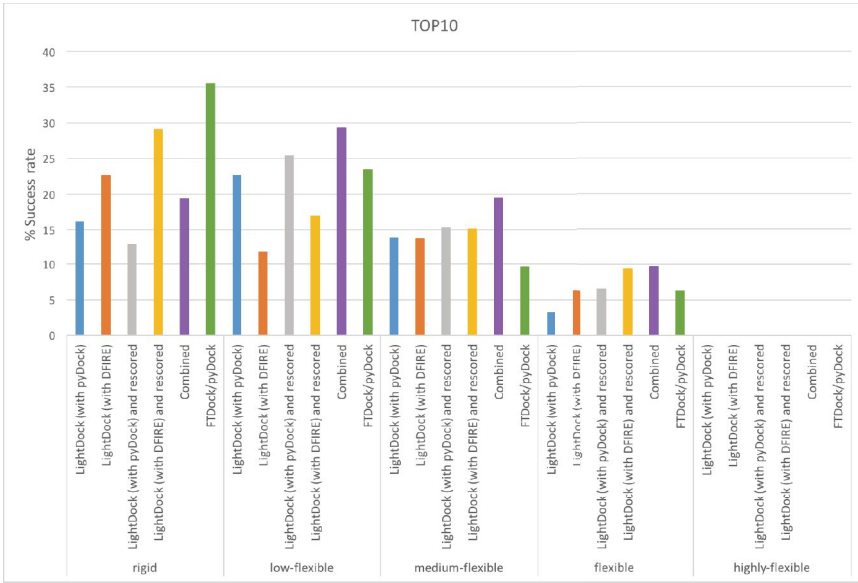


Figure 6. Predictive success rates for LightDock on Protein-Protein Docking Benchmark 5.0, n = 230, according to unbound-to-bound mobility. Top 10 success rates are shown for each LightDock strategy, in comparison to FTDock/pyDock standard protein-protein docking protocol.

probably improving the predictions in the more flexible cases, but at the expense of worsening the results in the rigid cases (due to the introduction of some noise in the already good geometries). Strikingly, LightDock with DFIRE showed its best results in the rigid cases, as in rigid-body FTDock+pyDock. It seems that the DFIRE scoring function cannot take advantage of the ANM model in the more flexible cases, perhaps due to the more coarse-grained character of the potentials. When both approaches are

rescored with pyDock, these tendencies remain, which suggests that the scoring function imposed some differences in the ANM-based conformational search. Results for top 100 show a similar fashion compared to top 10 (suppl. Figure S2).

4.3. Extending the framework

Four additional scoring functions have been implemented in the framework as a demonstration of the capabilities of LightDock for being extended with new scoring functions: MJ3h (Miyazawa and Jernigan 1999), PISA (Viswanath et al. 2013), TOBI (Tobi and Bahar 2006), and a faster version of the DFIRE scoring function used in this work, but implemented using the Python C extensions system. Implementation of user custom scoring function only requires the codification of two Python pieces of software: The Adapter and the Scorer classes. The Adapter is the piece of code in charge of translating the atomic structural information stored in PDB format into the model used in the Scorer class. For example, this model can be a set of atoms or a coarse-grained model, depending on the model that the custom scoring function needs. The Scorer class is the piece of code where the scoring function is implemented. The Scorer class receives a model from the Adapter class and uses the structural information into the coded scoring function. LightDock framework is agnostic to the model used as it will move into new receptor/ligand models independently from the structural representation used.

Several other options are supported by the framework. For instance, local energy optimization using a non-gradient algorithm has been implemented. For each glowworm groups and each step, the best glowworm in terms of scoring energy is minimized using this non-gradient algorithm. This strategy should help the algorithm to converge in fewer steps (data not shown).

On the other hand, the LightDock framework includes the option of using pre-calculated conformational ensembles, in which case each structure for receptor and ligand is identified by a unique id that is added to the optimization vector. For the future, a clearer strategy to define the distance between two conformers is needed so that it can be more efficiently used when one of the glowworms is moving towards the other one. The search could be optimized by maintaining a global list of the most successful or used conformers for receptor and ligand, and then use it to define a probability for selecting a given conformer.

Multi-scale or chained simulations are currently supported by the framework. One possible strategy is to perform a first run of the LightDock protocol using a given scoring function and then, after identifying the best energy wells, the predictions could be expanded by a new LightDock run, using the same scoring function or a different one, with finer sampling parameters for instance. In this way, a first quick run could be performed with a coarse-grained force-field, which can be followed by a more accurate refinement using a full-atom scoring function.

Finally, the framework includes more than 200 unit tests and more than 10 regression tests from point to point to guarantee a good testing coverage of the code, and additional usage examples to users who aim to extend the framework.

4.4. Computational cost

At the moment, one of the major drawbacks of the LightDock implementation is the computational cost compared to other protein-protein docking methods, especially compared to FFT-based methods. The average computation time for all the 230 complexes in the Protein-Protein Docking Benchmark 5.0 using DFIRE scoring function and 400 CPU cores is of 4.6 hours while for pyDock scoring function is of 7.0 hours in the same conditions. Our method is notably slower, although optimizations at the level of the scoring function (the most time-consuming part) could be performed, as shown with the faster version of the DFIRE scoring function using the Python C extensions mechanism. The speedup achieved with this faster implementation of the DFIRE scoring function is higher than 5x compared to the native Python implementation (Vidal 2015). In addition, LightDock is implemented using multicore and MPI Python libraries, and the algorithm is embarrassingly parallel which means that can ideally scale proportional to the number of CPU cores used. Native Python extension and Cython (www.cython.org) implementations, together with further

parallelization, open a promising way to optimize the LightDock framework in the future.

5. Conclusions

We have presented here a new protein-protein docking protocol called LightDock, which is based on GSO algorithm for sampling the translational and rotational space of protein-protein docking, and ANM representation for the inclusion of flexibility. LightDock aims to be a publicly available framework for testing and developing new scoring strategies for protein-protein docking. The use of pyDock scoring function during the search provides comparable success rates to state-of-the-art protocols, and the combination with additional functions, like DFIRE, can further improve the predictions. The docking framework has capabilities for the use of many different scoring functions and the inclusion of flexibility at different levels.

6. Acknowledgements

We give our thanks to Iain H. Moal for his invaluable help in many discussions on normal mode analysis, PSO and protein-protein docking in general, and to Miguel Romero for his help on implementing the pyDock contact-solvation function in LightDock.

Funding: B.J-G was supported by a FPI fellowship from the Spanish Ministry of Economy and Competitiveness. This work was supported by I+D+I Research Project grant number BIO2013-48213-R from the Spanish Ministry of Economy and Competitiveness.

7. References

Atilgan, A. R., S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar. 2001. "Anisotropy of Fluctuation Dynamics of Proteins with an Elastic Network Model." *Biophysical Journal* 80 (1): 505–15.

Bai, Xiao-Chen, Greg McMullan, and Sjors H. W. Scheres. 2015. "How Cryo-EM Is Revolutionizing Structural Biology." *Trends in Biochemical Sciences* 40 (1): 49–57.

Bakan, Ahmet, Lidio M. Meireles, and Ivet Bahar. 2011. "ProDy: Protein Dynamics Inferred from Theory and Experiments." *Bioinformatics* 27 (11): 1575–77.

Brenke, Ryan, David R. Hall, Gwo-Yu Chuang, Stephen R. Comeau, Tanggis Bohnuud, Dmitri Beglov, Ora Schueler-Furman, Sandor Vajda, and Dima Kozakov. 2012. "Application of Asymmetric Statistical Potentials to Antibody-Protein Docking." *Bioinformatics* 28 (20): 2608–14.

Chen, Kai, Li Tonghua, and Cao Tongcheng. 2006. "Tribe-PSO: A Novel Global Optimization Algorithm and Its Application in Molecular Docking." *Chemometrics and Intelligent Laboratory Systems* 82 (1-2): 248–59.

Chen, Rong, and Zhiping Weng. 2002. "Docking Unbound Proteins Using Shape Complementarity, Desolvation, and Electrostatics." *Proteins* 47 (3): 281–94.

Cheng, Tammy Man-Kuang, Tom L. Blundell, and Juan Fernandez-Recio. 2007. "pyDock: Electrostatics and Desolvation for Effective Scoring of Rigid-Body Protein-Protein Docking." *Proteins* 68 (2): 503–15.

de Vries, Sjoerd, and Martin Zacharias. 2013. "Flexible Docking and Refinement with a Coarse-Grained Protein Model Using ATTRACT." *Proteins* 81 (12): 2167–74.

Doerr, Allison, and Doerr Allison. 2015. "Structural Biology: Cryo-EM Strikes Gold." *Nature Methods* 12 (2): 102–3.

Dominguez, Cyril, Dominguez Cyril, Boelens Rolf, and Alexandre M J. 2003. "HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or Biophysical Information." *Journal of the American Chemical Society* 125 (7): 1731–37.

Doruker, P., A. R. Atilgan, and I. Bahar. 2000. "Dynamics of Proteins Predicted by Molecular Dynamics Simulations and Analytical Approaches: Application to Alpha-Amylase Inhibitor." *Proteins* 40 (3): 512–24.

Duhovny, Dina, Nussinov Ruth, and Haim J. Wolfson. 2002. "Efficient Unbound Docking of Rigid Molecules." In *Lecture Notes in Computer Science*, 185–200.

Eisenberg, D., E. M. Marcotte, I. Xenarios, and T. O. Yeates. 2000. "Protein Function in the Post-Genomic Era." *Nature* 405 (6788): 823–26.

Fernández-Recio, Juan, Maxim Totrov, and Ruben Abagyan. 2003. "ICM-DISCO Docking by Global Energy Optimization with Fully Flexible Side-Chains." *Proteins* 52 (1): 113–17.

Gabb, Henry A., Richard M. Jackson, and Michael J. E. Sternberg. 1997. "Modelling Protein Docking Using Shape Complementarity, Electrostatics and Biochemical Information." *Journal of Molecular Biology* 272 (1): 106–20.

Jimenez-Garcia B., Pons C., Fernandez-Recio J. 2013. "pyDockWEB: a web server for rigid-body protein-protein

docking using electrostatics and desolvation scoring." *Bioinformatics* 29:1698-1699.

Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2016-05-18].

Katchalski-Katzir, E., I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and I. A. Vakser. 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): 2195–99.

Kozakov, Dima, Ryan Brenke, Stephen R. Comeau, and Sandor Vajda. 2006. "PIPER: An FFT-Based Protein Docking Program with Pairwise Potentials." *Proteins* 65 (2): 392–406.

Krishnanand, K. N., and D. Ghose. 2008. "Glowworm Swarm Optimization for Simultaneous Capture of Multiple Local Optima of Multimodal Functions." *Swarm Intelligence* 3 (2): 87–124.

Krishnanand, K. N., and D. Ghose. 2009. "A Glowworm Swarm Optimization Based Multi-Robot System for Signal Source Localization." In *Studies in Computational Intelligence*, 49–68.

Lensink, Marc F., Sameer Velankar, Andriy Kryshtafovych, Shen-You Huang, Dina Schneidman-Duhovny, Andrej Sali, Joan Segura, et al. 2016. "Prediction of Homo- and Hetero-Protein Complexes by Protein Docking and Template-Based Modeling: A CASP-CAPRI Experiment." *Proteins*, April. doi:10.1002/prot.25007.

Liao, Wen-Hwa, Yucheng Kao, and Ying-Shan Li. 2011. "A Sensor Deployment Approach Using Glowworm Swarm Optimization Algorithm in Wireless Sensor Networks." *Expert Systems with Applications* 38 (10): 12180–88.

Liu, Song, Chi Zhang, Hongyi Zhou, and Yaoqi Zhou. 2004. "A Physical Reference State Unifies the Structure-Derived Potential of Mean Force for Protein Folding and Binding." *Proteins* 56 (1): 93–101.

Li, Xiaofan, Iain H. Moal, and Paul A. Bates. 2010. "Detection and Refinement of Encounter Complexes for Protein-Protein Docking: Taking Account of Macromolecular Crowding." *Proteins* 78 (15): 3189–96.

May, Andreas, and Martin Zacharias. 2008. "Energy Minimization in Low-Frequency Normal Modes to Efficiently Allow for Global Flexibility during Systematic Protein-Protein Docking." *Proteins* 70 (3): 794–809.

Miyazawa, Sanzo and Robert L. Jernigan. 1999. "Self-Consistent Estimation of Inter-Residue Protein Contact Energies Based on an Equilibrium Mixture Approximation of Residues." *Proteins: Structure, Function, and Genetics* 34 (1): 49–68.

Moal, Iain H., and Paul A. Bates. 2010. "SwarmDock and the Use of Normal Modes in Protein-Protein Docking." *International Journal of Molecular Sciences* 11 (10): 3623–48.

Moal, Iain H., Rocco Moretti, David Baker, and Juan Fernández-Recio. 2013. "Scoring Functions for Protein-Protein Interactions." *Current Opinion in Structural Biology* 23 (6): 862–67.

Moal, Iain H., Mieczyslaw Torchala, Paul A. Bates, and Juan Fernández-Recio. 2013. "The Scoring of Poses in Protein-Protein Docking: Current Capabilities and Future Directions." *BMC Bioinformatics* 14 (October): 286.

Morrison, Jack, and Morrison Jack. 1992. "QUATERNION INTERPOLATION WITH EXTRA SPINS." In *Graphics Gems III (IBM Version)*, 96–97.

Namasivayam, Vigneshwaran, and Robert Günther. 2007. "Pso@autodock: A Fast Flexible Molecular Docking Program

Based on Swarm Intelligence." *Chemical Biology & Drug Design* 70 (6): 475–84.

Rakhmanov, E. A., E. B. Saff, and Y. M. Zhou. 1994. "Minimal Discrete Energy on the Sphere." *Mathematical Research Letters* 1 (6): 647–62.

Ritchie, David W., and Vishwesh Venkatraman. 2010. "Ultra-Fast FFT Protein Docking on Graphics Processors." *Bioinformatics* 26 (19): 2398–2405.

Ritchie, David W., Vishwesh Venkatraman, and Lazaros Mavridis. 2010. "Using Graphics Processors to Accelerate Protein Docking Calculations." *Studies in Health Technology and Informatics* 159: 146–55.

Schneidman-Duhovny, Dina, Andrea Rossi, Agustin Avila-Sakar, Seung Joong Kim, Javier Velázquez-Muriel, Pavel Strop, Hong Liang, et al. 2012. "A Method for Integrative Structure Determination of Protein-Protein Complexes." *Bioinformatics* 28 (24): 3282–89.

Schueler-Furman, Ora, Chu Wang, and David Baker. 2005. "Progress in Protein-Protein Docking: Atomic Resolution Predictions in the CAPRI Experiment Using RosettaDock with an Improved Treatment of Side-Chain Flexibility." *Proteins* 60 (2): 187–94.

Shoemake, Ken. 1985. "Animating Rotation with Quaternion Curves." *ACM SIGGRAPH Computer Graphics* 19 (3): 245–54.

Spilotros, Alessandro and Dmitri I. Svergun. 2014. "Advances in Small- and Wide-Angle X-Ray Scattering SAXS and WAXS of Proteins." In *Applications, Theory and Instrumentation*, 1–34.

Stumpf, Michael P. H., Thomas Thorne, Eric de Silva, Ronald Stewart, Hyeong Jun An, Michael Lappe, and Carsten Wiuf. 2008. "Estimating the Size of the Human Interactome."

Proceedings of the National Academy of Sciences of the United States of America 105 (19): 6959–64.

Theodoridis S. and Koutroumbas K. 1999. “Pattern Recognition” *Academic Press*, London England.

Tobi, Dror, and Ivet Bahar. 2006. “Optimal Design of Protein Docking Potentials: Efficiency and Limitations.” *Proteins* 62 (4): 970–81.

Venkatesan, Kavitha, Jean-François Rual, Alexei Vazquez, Ulrich Stelzl, Irma Lemmens, Tomoko Hirozane-Kishikawa, Tong Hao, et al. 2009. “An Empirical Framework for Binary Interactome Mapping.” *Nature Methods* 6 (1): 83–90.

Vidal, Miquel. 2015. “Anàlisi i optimització de l'aplicació bioinformàtica de docking: Lightdock” *UPC Commons* http://upcommons.upc.edu/handle/2099.1/24830

Viswanath, Shruthi, D. V. S. Ravikant, and Ron Elber. 2013. “Improving Ranking of Models for Protein Complexes with Side Chain Modeling and Atomic Potentials.” *Proteins* 81 (4): 592–606.

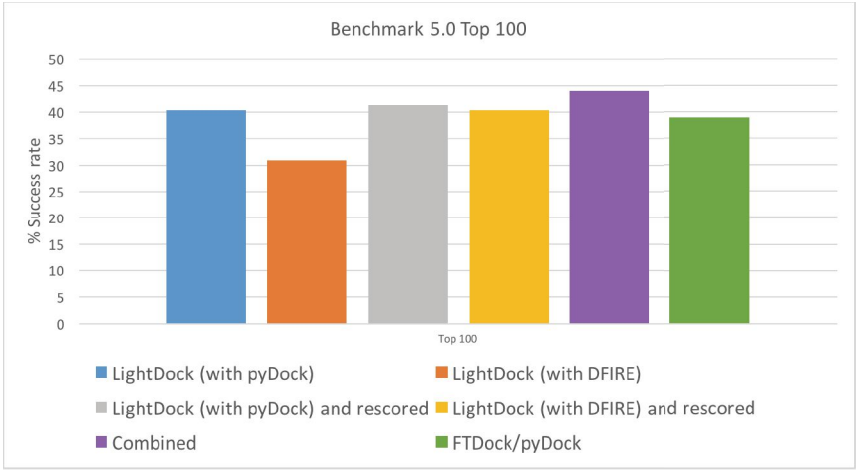
Vreven, Thom, Iain H. Moal, Anna Vangone, Brian G. Pierce, Panagiotis L. Kastritis, Mieczyslaw Torchala, Raphael Chaleil, et al. 2015. “Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2.” *Journal of Molecular Biology* 427 (19): 3031–41.

Ward Jr, JH. "Hierarchical grouping to optimize an objective function." *Journal of the American Statistical Association*. 58(301): pp. 236--44. 1963.

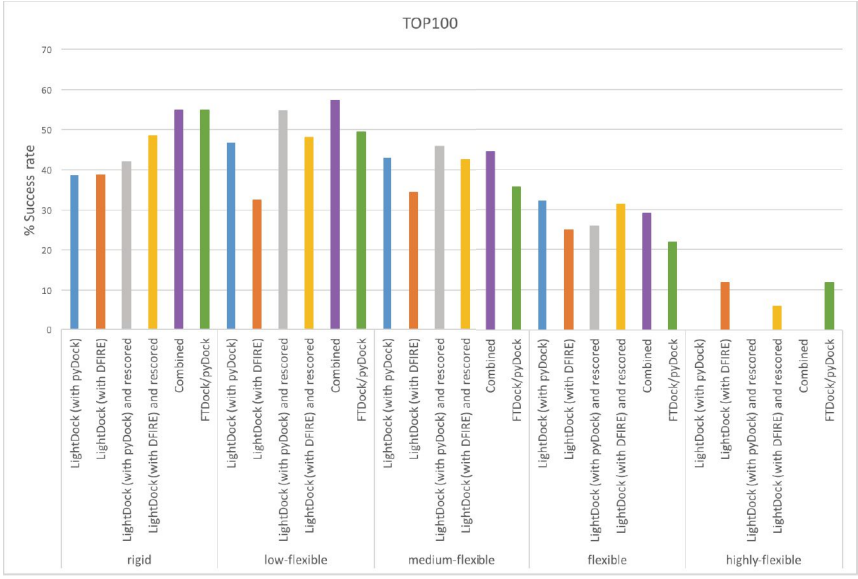
Zacharias, Martin. 2003. “Protein-Protein Docking with a Reduced Protein Model Accounting for Side-Chain Flexibility.” *Protein Science: A Publication of the Protein Society* 12 (6): 1271–82.

Zhengxin Huang, and Yongquan Zhou -. 2011. “Using Glowworm Swarm Optimization Algorithm for Clustering Analysis.” *Journal of Convergence Information Technology* 6 (2): 78–85.

Supplementary information



Suppl. Figure S1. Predictive success rates for LightDock on the Protein-Protein Docking Benchmark 5.0, n = 230. Success rates for the top 100 docking models are shown for: LightDock with pyDock scoring function (blue), LightDock with DFIRE scoring function (orange), LightDock-pyDock after rescoring by pyDock (grey), LightDock-DFIRE after rescoring by pyDock (yellow), final combination of LightDock-pyDock and LightDock-DFIRE after rescoring with pyDock (purple). For comparison, the performance of the standard FTDock/pyDock protein-protein docking protocol is shown (green).



Suppl. Figure S2. Predictive success rates for LightDock on Protein-Protein Docking Benchmark 5.0, n = 230, according to unbound-to-bound mobility. Top 100 success rates are shown for each LightDock strategy, in comparison to FTDock/pyDock standard protein-protein docking protocol.

4. Results summary

“Nobody expects the Spanish Inquisition!”

Monty Python

4.1 Optimization of complex modeling tools for HPC architectures and implementation in web applications

Web applications are especially useful for the biology community. First, it is the easiest way to encapsulate a workflow formed by different computational tools and to make it ready for non-expert users. The research group that make their computational tools available to the community their computational tools have a centralized way to track the changes on the software and a direct feedback from their users about the usefulness of their protocols. Second, it allows the opening of many protocols to the general public, without making distinction on the software or the resources required to use it, e.g. many potential users could not have access to HPC platforms to run a specific software. Finally, many protocols might be integrated in meta servers or databases which incorporate knowledge and capabilities of heterogeneous online tools.

Three different works are presented in this section. The first manuscript, describes a web server for protein-protein complex prediction using the pyDock (Cheng et al. 2007) protocol developed in our group. The second manuscript presents the CCharPPI web server, an online tool which helps characterizing protein-protein interfaces using up to 108 different energetic descriptors. These descriptors come from the public domain distributed software or have been be re-implemented in this web

application. This can be applied to characterize experimental complex structures, but can be also a valuable tool to score docking models. Finally, the last manuscript demonstrates how our protein-protein docking protocol, pyDock, can be extended using experimental SAXS data to better filter and classify predictions on basis to this extra information (Pons et al. 2010) on a public web server.

4.1.1 pyDockWEB: A Web Server for Rigid-Body Protein-Protein Docking Using Electrostatics and Desolvation Scoring

Despite its excellent predictive capabilities, the original pyDock protocol (Cheng et al., 2007) suffered from severe computing performance flaws, e.g. the protocol required single-CPU execution, and it was not available as a web server that could be used by external researchers. To overcome those problems, during this thesis the pyDock protocol was rewritten to avoid old library dependencies, the framework was parallelized, and a web application was developed.

The rewriting process made the source code of pyDock, written in Python, more robust to changes (more than 100 unit tests and regression tests were coded). After a few iterations of development, a version 3 of pyDock was internally released, only requiring Numpy (www.numpy.org) and Scipy (www.scipy.org) packages as external library dependencies. PyDock framework is

organized in different modules, for which the overall speedup was between 2 to 3 times faster in terms of computation time.

Parallelization process took two main aspects: sampling, by means of FTDock (Gabb et al., 1997), and scoring of the pool of generated poses (pyDockSER). Parallelization of the scoring step was almost trivial due to the embarrassingly parallel nature of the code. In the other hand, parallelizing FTDock protocol required a partial re-implementation using the distributed memory MPI library. Moreover, performance was dramatically increased thanks to the re-dimension of the grid to take advantage of better-performing convolution functions from the FFTW (www.fftw.org) library (see **Equation 1**).

Thanks to the technical improvements in the pyDock framework, a web application exposing many features and modules of the protocol was developed. The web server had an excellent reception by the community, with more than 4,600 jobs served at the time of writing this thesis, and with many regular users that produce actual scientific research and cited the article 28 times since its publication in 2013.

4.1.2 CCharPPI Web Server: Computational Characterization of Protein-Protein Interactions from Structure

A wide variety of biophysical functions and energy potentials have been developed by the community in an effort to characterize protein–protein interactions at atomic level, which is key to understand their role in biological systems. These tools are scattered through many publications and web servers, and are as diverse as the number of research groups that have developed them. Moreover, many of these predictive methods are only available at the literature and no specific implementation was reported. Within this context, we developed an online tool called CCharPPI for collecting or re-implementing many of these methods in an easy-to-use web tool. CCharPPI is able to calculate up to 108 different energetic descriptors for a given protein-protein complex structural model, and has demonstrated to be a useful tool for developing new methods based on machine-learning techniques (Moal et. al. "*Web-search based integration of biophysical models for protein assembly selection*", in preparation).

CCharPPI is a popular service with more than 1,950 served jobs at the time of the writing of this thesis. These jobs can represent a single protein-protein complex prediction or a batch job of up to 100 protein-protein complexes.

4.1.3 pyDockSAXS: Protein-Protein Complex Structure by SAXS and Computational Docking

Many high-resolution experimental techniques have strong technical limitations for determining transmembrane protein complexes or describing the dynamic nature of protein-protein interactions. On the contrary, SAXS and other low-resolution techniques can be applied in a high-throughput context, but with resolution limitations. Previous work in the group showed that the combination of SAXS data and the protein docking protocol pyDock doubled the predictive success rates of docking. In this thesis, pyDockSAXS has been made it available to the experimental community as a web server.

The web tool accepts receptor and ligand PDB structures, and CRYSOL SAXS data as input. For advanced, there exists the option to use previous docking pyDockWEB results as a starting point. At the time of publication of the article, there was only another web application with similar capabilities. Despite its recent publication, pyDockSAXS tool has served more than 280 jobs to external users.

4.2 Validation and current challenges in protein-protein docking methods

The growing interest in protein-protein interactions and the technical advances in the computational field have fostered the number of *in silico* tools developed in the past years. With the aim of modeling protein complexes starting from the isolated component structures, testing and comparing these computational methodologies have become fundamental in order to assess their performance, identify their limitations, and encourage new developments in the field. In this context, community-wide experiments such as CAPRI provide a common ground for testing the predictive capability of currently available docking methods.

First, the performance of our pyDock protocol (Cheng et al., 2007) on the last CAPRI round (Lensink and Wodak, 2013) will be evaluated and discussed. Second, a suitable set of protein-RNA complex structures has been compiled in order to establish a common framework for the evaluation of different protein-RNA interaction predicting methods. The last manuscript describes an update of the protein-protein benchmark, which integrates the affinity and protein-protein benchmarks and where our group has participated in the evaluation of the new complexes included and the success rate of our protocol has been updated.

4.2.1 Expanding the Frontiers of Protein-Protein Modeling: From Docking and Scoring to Binding Affinity Predictions and Other Challenges

The fifth CAPRI edition (2010-2012) (Lensink and Wodak, 2013) was formed by a total of 15 targets, including special targets consisting in the prediction of binding affinity values and free energy changes upon mutation (Moretti et al., 2013), as well as in the prediction of sugar binding and interface water molecules (Lensink et al., 2014). Our group participated in all the proposed targets with high success: our predictions were globally placed among the top 5 ranked groups out of more than 60 participants (Lensink and Wodak, 2013) (**Fig 1** from **section 3.1.2** and **Table S11A** from Lensink and Wodak, 2013).

Compared to previous participations, our increase in performance was due to the generation of docking poses with FTDock (Gabb et al., 1997) at a grid resolution of 0.7 Å (instead of 1.2 Å as in the past), as well as with SwarmDock program (Moal and Bates, 2010) for part of the targets. In selected targets (T47, T48 and T58), external biological information was applied in form of distance restraints, although this hardly made any difference. In target T58, SAXS data were used for complementary scoring by applying our pyDockSAXS protocol (Pons et al., 2010). The predictive performance did not improve at all due to the globular shape of the complex, a difficult situation for the SAXS technique. In general, our pyDock protocol submitted consistently good models for all non-difficult cases.

These represented realistic conditions in which subunits were in the unbound conformation or needed to be modeled based on homologous templates. In all cases but one, pyDock successful models were ranked within the first five submitted solutions, being ranked as first in two out of six successful cases.

In the non-standard targets, T47, T55, T56 and T57, new *ad hoc* protocols had to be developed. T47 was an easy target regarding protein-protein complex prediction, but the real challenge was the prediction of the water molecules within the complex interface. We based our predictions in the DOWSER *ab initio* optimization procedure (Hermans et al. 1996), with reasonable results. However, the top performing methods were based in deriving initial water positions from interfaces of similar complexes followed by an energetic minimization (Lensink et al., 2014). In T55 and T56 targets, the goal was to predict the binding affinity changes upon mutations on two designed influenza hemagglutinin protein binders. Our approach was based on a machine learning protocol using 85 different energy descriptors. Our protocol was one of the most successful ones, placed within the top 3 out of 22 groups (Moretti et al., 2013). Target T57 involved the prediction of a protein-sugar interaction. In this case, we developed a new protocol based on a combination of different scoring functions, but no correct models were submitted.

In general, for the standard protein-protein docking cases, the different docking methods showed robustness in the delivered results, but there were two especially difficult targets: T46 and

T51. In the case of T46, the main difficulty came from the need of modelling the partners from remote homologues, which constitutes one of the main challenges in protein-protein docking as pointed in **section 1.5.1**. Moreover, in target T51 the main difficulty was the big RMSD difference between the bound and unbound conformations of the interacting partners, which shows the importance of addressing conformational flexibility in future docking developments. This is actually related to one of the main objectives of this thesis, which focus on the development of a new docking framework to permit the inclusion of flexibility in the docking model.

4.2.2 A Protein-RNA Docking Benchmark (II): Extended Set from Experimental and Homology Modeling Data

The first protein-RNA docking benchmark was compiled, comprised a total of 106 protein-RNA complexes. In 71 of them, the unbound coordinates were available for at least one of the molecules, and in the remaining 35 cases, at least one of the molecules needed to be built by homology modelling. Due to the scarcity of available structures in the PDB at the time of the compilation of the benchmark, the benchmark was extended by increasing the criterion of non-redundancy up to 70% of sequence identity, and by including homology models as well as pseudo-unbound and bound structures in the case of the RNA molecules. This dataset of protein-RNA complexes can be used

to test new and current protein-RNA docking methods, and is expected to foster methodology development in the field of protein-RNA structural modeling. The benchmark was updated to its version 1.1 in September 2015 to fix some errors in renumbering, and it has been used by many groups for testing their new methods.

4.2.3 Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2

An updated version of the protein-protein docking benchmark (Hwang et al. 2010) is reported in this work. This work includes also an updated version of the binding affinity benchmark (Moretti et al., 2013), which has been updated too. The benchmarks consist of non-redundant, high-quality structures of protein–protein complexes along with the unbound structures of their components. This version 5.0 of the docking benchmark contains fifty-five new complexes, 35 of which have experimentally measured binding affinities. These updated docking and affinity benchmarks now contain 230 and 179 entries, respectively. The antibody–antigen complexes is the category which has increased more in number, with an increment of 67% and 74% in the docking and affinity benchmarks, respectively. This probably reflects the growing interest in antibody-based therapeutics. This update also includes complexes with multiple binding modes,

which have been split into multiple cases. The composition of this benchmark still favors the number of easy-cases (rigid-body) as compared to the difficult ones, but the difference is more balanced than in previous versions of the benchmark.

Several previously developed docking and affinity prediction algorithms have been tested on the new cases. Regarding the docking predictions, and considering only the top 10 models, an overall prediction success rate of 38% is achieved on the new 55 cases. If only rigid-body cases are considered (32), the top 10 success rate increases up to 50%. Regarding the affinity prediction algorithms, predicted scores show significant correlation with the experimental values (r=0.52). If only rigid-body cases are considered, the correlation coefficient largely increases (r=0.72).

The docking success rates and the affinity predictive results were lower as compared to previous versions of the benchmarks. These new challenging cases included in these updated versions of the benchmarks show that new developments are needed in the structural and energetic modeling of protein-protein interactions.

4.3 New methods for structural protein-protein complex prediction

The aim of protein-protein docking methods is to predict the complex structure starting from the structure of the unbound partners. The nature of this problem is very complex and intractable by more physically accurate methods such as molecular dynamics. From the late 70s of the past century, several docking methods have been proposed, with very promising results. But community-wide experiments such as the CAPRI international contest have demonstrated the limitations of the current methods. In order to overcome many of the limitations of the current protein-protein docking methods, a new method called LightDock is proposed in this thesis. LightDock has been developed with the purpose in mind of being an experimenting platform where current and future developments could easily be prototyped. LightDock is a scoring function agnostic framework, in terms that the users can incorporate their own one to the framework, written in Python with capabilities of incorporating normal mode analysis, precomputed ensembles and local non-gradient minimization, in order to model the protein flexibility.

4.3.1 LightDock: a framework for multi-scoring function flexible protein-protein docking

The LightDock method is tested in the last version of the Protein-Protein Docking Benchmark (Vreven et al. 2015). The

configuration of the framework used is i) number of non-trivial normal modes for receptor and ligand equal to 10, ii) number of clusters for each complex in the benchmark is of 400, iii) number of glowworm agents per cluster is 300 and iv) the scoring functions tested are a more coarse-grained version of pyDock (Romero-Durana et al. in preparation), but still more physically accurate and not based in propensities as the second one used, DFIRE (Liu et al. 2004).

The use of the two different scoring functions demonstrates that the framework is capable of successfully minimizing in different force-fields thus the success rate will depend on the goodness of the chosen scoring function. Moreover, the results show how combining the ranking from the use of the two different scoring functions prior to the rescoring of them using pyDock (Cheng et al. 2007) energy function improves notably compared to the default pyDock protocol.

The analysis performed shows that the use of normal modes in the rigid-body cases of the Protein-protein benchmark deteriorates the results compared to protein-protein docking rigid-body methods as pyDock, but helps in the medium and high-flexible categories.

In summary, the combination of pyDock and DFIRE scoring functions in the LightDock framework outperforms the success rate of the pyDock rigid-body protocol: 19.11% versus 16.45% in the top 10 and 44% vs 38.96% in the top 100 respectively.

5. Discussion

And following our will and wind
we may just go where no one's been.
We'll ride the spiral to the end
and may just go where no one's been.
Spiral out. Keep going...

Lateralus, Tool

In these days, in which sequencing the entire genome of many organisms is a relatively cheap and easy task, the so-called post-genomics era (Kenyon et al. 2002), the next big challenge in life sciences is the unraveling of the complex protein-protein interaction networks, with the ultimate goal of understanding life processes at molecular level. High-throughput experimental techniques combined with computational methods have contributed to partially describe the interactomes of several organisms, but a complete understanding of the mechanisms underlying these protein-protein interactions requires a complete vision at atomic detail of all protein complexes. Despite the success of some experimental techniques like X-ray crystallography, NMR, SAXS or the more promising cryo-EM, there is still a huge gap between the number of estimated protein-protein interactions and the actual number of structures solved and available. Computational methods such as protein-protein docking could shed some light on reducing that gap, complementing experimental techniques by providing structural and energetic large-scale modeling of protein interactions. Nevertheless, computational models are limited too in many aspects. In the protein-protein docking problem, conformational flexibility remains one of the main challenges due to its inherent and computational complexity. As a consequence, an important concern is the general poor success rate of current docking methods and their high computational cost, which limits their possible application to large-scale projects.

The work in this thesis has produced new developments and optimizations of our docking protocol pyDock, with emphasis in its computational performance, to facilitate its application to large-scale systems and its implantation in web-based tools. Moreover, a novel strategy for sampling and new methods for including conformational flexibility have been developed. During all the methodology development in this thesis, we always had in mind a series of proposed good practices in developing scientific software, which are important for the reproducibility of the research and the usefulness to the community.

PyDock optimizations for HPC architectures

As pointed in **section 4.1**, our pyDock protocol had several flaws in terms of computational performance. After the rewriting of the scoring function module and the parallelization of the sampling with FTDock, pyDock was able to perform protein docking predictions in terms of minutes, and not of hours or even days as before, without losing any predictive accuracy.

One of the main advantages of the docking sampling in discrete steps is its inherent embarrassingly parallel nature. New poses in FFT-based methods as FTDock are generated by a simple angle increment in the FFT space. This increment is independent from previous calculations so it makes the protocol a perfect candidate for parallelization. However, after the parallelization of the sampling process, the computation time was not dependent of the size of the protein partners. After some research, it was clear

that the problem was due to the use of different functions internally in the FFTW library that depended on specific values of the total number of grid cells used (Jiménez-García et al. 2013, Equation 1). The solution came when the number of grid cells was increased to match Equation 1, reducing the computation time and, finally, making it dependent only on the size of the protein complex.

Further optimizations could be made in the sampling process, e.g. using GPU-based architectures. Other FFT-based protocols such as Hex (Ritchie and Venkatraman 2010) or ClusPro (Landaverde and Herbordt 2014) have explored this possibility with notable speedups. In our case, this possibility was discarded due to the limited access to GPU-capable hardware at the time these optimizations were developed. It could be a good exercise in the future to rewrite the FTDock application to use GPU accelerators.

In terms of overall speedup, the new and optimized version of the pyDock protocol outperformed previous iterations. In the version used in the pyDockWEB server (Jiménez-García et al. 2013), the use of 14 CPU cores made the total computation time two orders of magnitude faster (Figure S.1. Jiménez-García et al. 2013). As a consequence, the protocol was fast enough to finish in a few hours, and thus, to be implemented in a web server to make it available to the scientific community. In addition to the development of the web server, the increase in performance allowed its application to large-scale docking experiments. With

previous versions of the slower protocol, the limitation in computational resources was an important drawback and only partial interactome experiments were able to be performed (Mosca et al., 2009). On the contrary, the fastest version of pyDock has been used in larger experiments for hot-spot predictions in diseasomes such as HIGM5, LHON, CRC, MCI, HIV-1 or CMH (Barradas and Fernández-Recio, in preparation).

Development of web tools for the scientific community

Web server and web-based tools are powerful resources for the scientific community. They allow different researchers to use published protocols and workflows without caring about computational resources or maintaining or installing complex pieces of software. These advantages have contributed without a doubt to the popularization of these services in the past years.

Although there is no specific study about the real contribution of web-based tools to published research, two main aspects point to the real utility of these tools: 1) special issues in important journals are published every year only comprising web servers and online tools, and 2) if we trust the number of citations as a metric of utility, many of the most cited articles in the life sciences domain are describing web tools (mainly within the genomics scope).

During this thesis, five web-based tools were developed, and one more is still under development: pyDockWEB (Jiménez-García et

al. 2013), CCharPPI (Moal et al. 2015), pyDockSAXS server (Jiménez-García et al. 2015), OPRA server (http://life.bsc.es/pid/opra, manuscript in preparation), PyDockRescoring server (http://life.bsc.es/pid/pydockrescoring, manuscript in preparation) and TailBuilder (http://life.bsc.es/pid/tailbuilder, under development). All of these tools make intensive use of HPC resources, an aspect that makes specially complicated their development and maintenance. Despite the lack of specific frameworks to develop scientific online tools, the development of the first tool, the pyDockWEB server, helped us to fully understand the requirements and problems concerning the development of a web server. The other tools have a similar architecture based on a web2py (http://www.web2py.com/) front-end, a Python server side in charge of scheduling jobs in our private cluster and a relational database which stores all the information and glues the other two layers. This structure is sufficiently flexible to be adapted for the development of new tools and it can be easily deployed.

As pointed in **section 4**, these web tools have successfully computed and served many jobs since their availability. In addition to their popularity, two other important aspects can be remarked. First, pyDockWEB server scored in the second position out of many other participants of the server category in the last CAPRI meeting (6^{th} edition). Second, CCharPPI server will be incorporated to the CAPRI-EBI analysis pipeline to calculate energetic and structural features on participant's

decoys in the near future. These two points are especially relevant in order to encourage future developments and updates on the published web tools, which development abandonment is an important issue in many published web servers.

New methods of sampling and energy optimization

Protein-protein docking energetic landscape is described by a high number of dimensions and degrees of freedom, so energetic optimization on such a huge search space is still a hard computational problem.

Karl Pearson introduced in 1905 the problem of the random walk. In his own words "the most probable place to find a drunken man who is at all capable of keeping on his feet is somewhere near his starting point!" (Pearson 1905). A random walk is a mathematical formalization of a path that consists of a succession of random steps. Typically, an ensemble of these *walkers* are used, starting from random initial points in the search dominium, to integrate or *to sample* the value of the objective function in multi-dimensional spaces. The next step of the walker or *sampler* will look for a reasonably high contribution towards the total integral. This technique is known as random walk Monte Carlo (RWMC) and molecular energy optimization has been typically described using this method. The strategy for refusing some of the proposed random walks can vary, but important methods are Gibbs sampling, multiple-try Metropolis and slice

sampling among others. While RWMC methods are very useful for describing the protein-protein docking energetic landscape, they suffer from important drawbacks. First, it seems not so obvious how many random walks are required for substantially describing a given landscape. Second, further strategies are required if the information gathered from previous random walks is going to be reused for subsequent random walks, e.g. direction tensors with an associated probability. Finally, more strategies are required to differentiate exploration from exploitation phases in order to not get trapped in local minima or maxima. In summary, all of these limitations justify more developments in methods for sampling and energetic optimization.

Swarm intelligence (SI) is a family of artificial intelligence algorithms inspired by emergent systems in nature. Basically, these algorithms make use of agents which follow very simple rules to interact locally with other agents in a decentralized way. In nature, these interactions lead to complex emergent patterns or systems, e.g. fish schooling and bird flocking, ants forage for food, wasp nest building or termite mounds. SI algorithms have many interesting properties: heuristics are generally simpler because there is no need of central control, they are inspired by nature metaphors, which makes their parameters easy to choose and to understand by humans and, finally, they tend to be easily scalable as more agents can be added at any time. SI algorithms have been applied to many different problems, from numerical optimization to multi-robot coordination and navigation. In the life sciences context, SI algorithms have been applied to some

specific problems as data classification for disease diagnosis (Assarzadeh and Naghsh-Nilchi 2015; Zyout et al. 2015), protein-ligand docking (Chen et al. 2007; Namasivayam and Günther 2007; Ng et al. 2015; Liu et al. 2013) or protein-protein docking (Moal and Bates 2010).

In this thesis, I have developed LightDock, a new method for the sampling and energetic description of the protein-protein docking landscape. LightDock makes use of the GSO from the SI family of algorithms for predicting complexes given the initial unbound structures of the partners. GSO implementation in LightDock has very interesting properties, such as multiple capturing different local and global energetic minima in the same run, and it has mechanisms to avoid the leap-frog problem in comparison to traditional RWMC approaches. Moreover, all the parameters from the GSO are nicely described by the bio-inspired metaphor that represents the glowworms attraction mechanism, which makes the algorithm easily understandable for humans and simplifies the process of choosing the parameter values.

The evaluation of LightDock on the Protein-Protein Benchmark 5.0 showed promising results. In addition, LightDock has novel capabilities compared to other protein-protein docking methods, such as the possibility of being extended with other scoring functions. However, there is room for improvement. In this context, several approaches can be explored in the future. First, the use of conformational ensembles could improve the overall success rate in low and medium flexible categories (Pallara et al.

submitted). The support for conformational ensembles has been included in the LightDock framework, but we need to explore more exhaustively different strategies of how the agents in the algorithm exchange conformer information. Second, local energetic minimization could help the algorithm to converge to the most favorable energetic values when the near-native energy well has been correctly identified. This could also improve the ratio true/false positives, since better energy is expected from true positive predictions. While we have included in the framework support for optimizing the best glowworm in terms of energy at each step, this strategy needs to be more exhaustively explored. Finally, while some certain degree of flexibility in the backbone is included thanks to the use of the ANM, a more efficient side-chain conformation prediction could dramatically improve the overall performance. One possible strategy could be to estimate the value of backbone deformation on the unbound proteins by using a fixed number of non-trivial normal modes, and then try to discretize the orientation of the side-chains using a rotamer library. Once all of the side-chains are discretized, they could be included in the optimization vector of the GSO algorithm, so that the probability of moving from one orientation to another would be given by the value in the rotamer library for that side-chain. This strategy presents a combinatorial problem due to the huge number of dimensions to be optimized, although performance could improve if the optimization on the side-chains were only computed at the protein-protein interface. This strategy opens all sorts of interesting questions from both technical and theoretical points of view. In addition, the use of ANM opens a

new problem of how estimating the degree of flexibility of a complex to make use or not of ANM to avoid sampling noise. All of these considerations have not been exhaustively studied in this thesis due to time limitations, but are important points for future developments.

On the performance of the first joint CASP-CAPRI experiment

The first joint CASP-CAPRI experiment consisted in the structural modeling of homo- and hetero-dimers, and homo-tetramers, with 25 targets from the CASP11 2014 round. PyDock submitted at least one acceptable model in 11 out of 12 easy homo-dimer targets, either as predictors or as scorers. In addition, it successfully identified two out of the six difficult homo-dimer targets as scorers, and one out of the two hetero-complex targets. On the other hand, pyDock did not submit any successful model for any of the five tetramer targets. The main difficulties of these targets were the inaccuracy of the homology-built subunit models (and the cumulative error for each considered subunit), and the smaller pair-wise interfaces. Compared to other docking methods, pyDock predictions were placed within the top 10 ranked groups out of a total of 25 predictor groups, and within the top 5 ranked groups out of a total of 12 scorers participating in Round 30 (**Table 4** from Lensink et al. 2016).

The overall performance of the different methods was encouraging. The results showed that the prediction of homo-

dimer assemblies by a combination of homology modeling and docking can be successful for targets with subunits featuring large enough interfaces to represent stable associations. On the contrary, the inaccurate estimation of the oligomeric state added a confounding factor for the predictions. On tetrameric targets, the performance was really disappointing for all the methods, with the same problems that pyDock suffered. Interestingly, the analysis shows that in the structural prediction of homo-oligomers, docking procedures tend to perform better than standard homology modeling techniques, and that highly accurate models of the protein components are not always required to identify their association modes with acceptable accuracy (Lensink et al. 2016).

Building quality into scientific software

Software quality is a problem which has been historically only addressed in the industry. From the monetary point of view, it is clear that a buggy software would affect economic balances at the end of the year if the software is an essential piece of the business, although a buggy software could have dramatic consequences in other areas, e.g. the software which controls braking assistance on a car or the software for early detection of attacking missiles. May the software that research scientists develop not be as critical as the previous examples, but it can lead to wrong results and conclusions if its quality is not carefully considered.

There is still room for raising awareness on the problem within the scientific community, but reasonable efforts have been made in that direction. Important journals in different fields are becoming aware of the problem, especially concerning reproducibility and repeatability issues, asking for the source code of the software used for research and to analyze produced data. But this is still not sufficient as the peer review process rarely includes a quality assessment of the software nor other techniques as testing are not usually taken into account.

Here are detailed some good practices that could help tackling the problems of reproducibility and repeatability in scientific software development. They are organized in hardware abstraction, operating system and software management, software development, and research data management sections.

Hardware abstraction

There is no such thing as a standard hardware platform. This is a consequence of the expansion and popularity of the open PC platform at the eighties, so the actual situation is that there are many available options, from technological architecture to manufacturer quality, when we consider physical devices, such as the CPU or the RAM memory. Although at a given moment there are always some architectures that can be considered more popular than the others, there is no general framework of reference regarding the reproducibility of the output from computer programs, given that the instruction set varies: a small difference in the order of execution of one instruction could affect

the value of a floating point operation, and this difference would be propagated to the following operations leading to a problem of numerical instability.

Hardware is neither free of design nor of execution bugs. There are many famous examples in computing history, such as the Pentium FDIV bug reported in 1994. There was an error in the Intel P5 floating point unit due to missing entries in the lookup table used by the floating-point division circuitry (Price 1995). This error lead to inconsistencies in some floating point operations such as the decimal division of 4195835.0 by 3145727.0 which was calculated in the flawed Pentium as 1.333739 when the correct value is 1.333820 (Cipra 1995). In a long-term molecular dynamics simulation (Karplus and McCammon 2002), this difference in the fourth significant figure could imply a complete disaster in the prediction.

In the past decade, hardware virtualization has experienced an explosion in development and usage, and nowadays it could be considered a mature technology, widely used in cloud and grid facilities. Hardware virtualization hides the physical attributes of computing platforms and offers the users a new abstraction layer that represents a different hardware platform. If the new abstraction layer can be packed, exported and executed over the same hardware virtualization software (virtual machine monitor) in a sufficient amount of different hardware platforms, then we have a good opportunity to avoid hardware peculiarities and to assure reproducibility of our software independently from the

actual hardware where it is executed. There are several virtualization software solutions available, but probably the most popular projects out now are VirtualBox (www.virtualbox.org), VMware (www.vmware.com) and QEMU (www.qemu.org). VirtualBox, now formerly Oracle VM VirtualBox, was initially released as free and open source software in January 2007 with GNU GPL2 license. It is a mature project with support for Windows, OS X, Linux and Solaris operating systems among others and now is part of the Oracle company. VMware desktop software is another mature option, VMware was one of the first companies to successfully virtualize the x86 platform, and runs in Linux, Windows and OS X platforms. Their basic solution is free, but the most advanced features require a license. QEMU is a virtualization tool only available for Linux machines, but it does not require administrator privileges to run, since it is built on the top of the Kernel-based Virtual Machine system available in the Linux kernel. This property makes it a perfect candidate for small virtual machines that will be run on Linux hosts. Packaging and developing the research activity in a virtual machine has many advantages: stable hardware, frozen versions of operating system and library dependencies, capability of taking snapshots of different stages of a given virtual machine, and a way of easily deploying clone copies of the virtual machine. But there are a few drawbacks of using this technology too: the loss of performance (an extra layer of abstraction between hardware and software is being used), or the size of the virtual machine at the time of distributing it. In any case, we strongly believe that the

advantages of having a controlled and distributable environment clearly compensates the drawbacks of size and performance.

Operating system abstraction and software management

Operating systems are living entities: they are periodically updated in order to fix security or software bugs, or to improve the performance of existing libraries, for example. Thus, at the time of reproducing a given research, it is not sufficient to only mention the version of the operating system, because we would be losing information on the libraries and software versions. In GNU/Linux distributions, software management is usually performed using package tools like apt-get in Debian and Ubuntu flavours, rpm in Red Hat flavours, etc. These tools always store the actual version of all the different software installed and the sources where the software has been downloaded, so it is possible to reproduce the actual version of the operating system and its software if those lists are shared. But in other operating systems such as OS X or Windows, this problem could not be solved in an easy way. Another important issue is that the software we would like to execute could have some library dependencies which are in conflict with the ones that are actually installed in our operating system, or maybe there is no available information on the version of the library at all. In the previous section, we recommended to use a virtual machine to freeze the actual version of the software and the operating system, but if it

is not possible to use virtualization tools, there are alternative solutions as follows.

Docker tool (www.docker.com) allows the creation of software containers in Linux in a more lightweight way that using virtual machines. Docker makes use of the characteristics of the Linux kernel of separating execution namespaces and encapsulating resources via cgroups to pack any given software with its dependencies in a distributable container. Theoretically, It is enough to have Docker installed to use any third-party container. Moreover, in OS X operating system, software can be installed or removed in a Linux-way using homebrew (brew.sh) or macports (www.macports.org) for example. In other cases, there are some packages that encapsulate the most commonly used libraries. That is the case of scientific libraries in Python, where the cross-platform Anaconda project (www.continuum.io) provides compilations of libraries with a specific version that makes distributing Python scripts an easy task: sharing the Python source code and the version of the Anaconda package used should be enough to reproduce the software execution conditions.

Software development

Developing software is a complex process which involves tasks as writing, testing, debugging and maintaining the source code. Interestingly, 38% of scientists spend at least one fifth of their

time developing software, but only 47% of them have a good understanding of software testing, and only the 34% of them consider that formal training in developing software is important (Merali 2010). Scientific and research software can range from small command-line scripts to huge pieces of software that can be considered as final products. The requirements for every piece of software will clearly vary, and not a single development methodology will fit all the different needs, but there are some common guidelines to be followed in order to improve the quality of the software.

In terms of reproducibility and repeatability, we are interested in having a reference framework based on known results. That reference, the so-called golden data, will help us to identify bugs or possible issues at any time we are introducing changes in our code. The process of comparing known results, golden data, with the actual output of our code is called testing. Without testing, we cannot guarantee that our code is doing what it is claiming to. There are different ways of testing and it will depend on the amount and complexity of the code that we need to test: from unit tests of small pieces of code such as functions, to regression or point-to-point tests where the code is tested against other pieces of software it has to interact with. There are many testing frameworks that can help in this task, and choosing one or another will depend on the technology, availability, programming language, etc., but there is a testing methodology that can help in the process of designing new software: TDD (Test-Driven Development) (Janzen and Saiedian 2005). In TDD,

the philosophy of writing software differs from the typical point of view of writing code and then writing some testing code that will test the original code. In TDD, the test is written before writing the code that will fulfill the requirements. This methodology can be a good option for scientific software (Mugridge 2003) because we can build and write our software starting from known examples we would like to reproduce, and then it can be expanded to deal with new features or cases. Previous research in this field demonstrates that TDD helps improving the overall software quality (George et al. 2004).

Dealing with simulations, especially if they start from some random initial conditions, can be a hard problem in terms of reproducibility. There are two approaches to test the performance of our code in that situation: i) guarantee that the initial conditions are always the same, or ii) check at the end of the process for the correctness of the results. This second approach can be a hard problem depending on the complexity of the interpretation of the results, and usually some human action will be required. However, the option of guaranteeing the initial conditions can be easily performed if we make use of simulation seeds with the random number generators of the code. Storing these seeds and using them in unit or regression tests can help to automatize the whole reproduction process.

Source code is dynamic in its nature and it mutates depending on the changes in the requirements as new features are included or problems are fixed. Keeping the changes made to the source

code is a major concern in software development, and tools such as version control systems are accepted as a standard in industry. With open source tools such as subversion (https://subversion.apache.org), git (https://git-scm.com/) or mercurial (https://www.mercurial-scm.org/), and online spaces to share and managing code as GitHub (github.com) or Atlassian Bitbucket (bitbucket.org), the changes are tracked and it is easy to recover old versions, sharing the code with others and fostering collaboration between developers. These tools support not only source code, but other types of documents that require version control.

Another important point we cannot forget is documenting the code. Researchers tend to produce prototypes that are poorly documented and difficult to interpret, which makes the process of reproducing the results in the future a hard task. Inline documentation has to be considered as an essential part of the source code if we want our code to be understandable and maintainable and readable for our users.

Finally, there is an important issue concerning code writing by scientists: they often do not consider their code sufficiently good for being published. We agree with previous authors (Barnes 2010) in that if the code is good enough to do the job for which it was written, it is sufficiently good to be released as open source. Openness can help to improve both the code used by the scientists in many projects and the ability of the public to engage

with it (Barnes 2010). This opinion is also supported by many other authors (Ince et al. 2012).

Research data management

Managing research data is another important and very challenging topic. There are several issues to consider. One of them is to keep the data alive. That means that when using a specific data format for the output of our research, we should assure that there is going to be a piece of software capable of reading and managing our type of data in the present. That not only applies to the format of the data, but the physical support where data is stored. Several problems should be considered here: physical devices could become obsolete in a few years with no hardware available to read them, the support could be corrupted in time, lifespan of CD support can be shorter than expected (as compared with other technologies), etc. Nowadays, scientists usually publish their research data on the Internet, usually as a support data of a published article, or as a piece of code in popular sites such as github.com or pastebin.com. In order to follow the scientific research method, it is important to have available all the relevant information and data, and in the correct order. Plain-text format such as TSV (tab-separated values) could help to improve readability and can be easily imported by many applications and online services. In addition to output data, the slides, posters and all the different material generated during the research process could be susceptible of

publication and reference, but at the moment there is no consensus or a standard way to do it.

Final remarks

In this thesis we worked on the identification of the different problems regarding reproducibility and repeatability in scientific software development in order to tackle the problem. Finally, we gave a set of good practices to follow based on successful histories of software development. This set of good practices is a good starting point, but more effort is required. Scientific community is still not concerned about the dimension of the problem and the issue is likely to get worse as many researchers consider research software a distraction. This discussion will remain opened for many time in the near future, but here are some points that could mitigate the problem: i) more training at the academic and research levels on software development, ii) mixed research groups where software engineering knowledge is present, iii) journals making research and analysis software mandatory, iv) congresses on scientific and research software quality and v) development of testing and publication tools.

6. Conclusions

“Bob: I don't want to leave.
Charlotte: So don't. Stay here with me. We'll start a jazz band.”

Lost in Translation

1. Technical improvements in the pyDock computational performance have facilitated the development of the pyDockWEB protein-protein docking web server. This web server has shown an excellent predictive performance by ranking 2nd out of 14 automatic web servers in the CAPRI 6th evaluation meeting.

2. The CCharPPI web server has been built to bring together many different descriptors for characterizing protein-protein interfaces, which can be applied to fast prototyping new predictive models.

3. A new pyDockSAXS web server has been built to efficiently combine experimental data from small scattering X-ray (SAXS) and protein-protein docking predictions.

4. pyDock showed excellent performance in the last two editions of the blind community CAPRI experiment, by ranking within the top 5 predictors out of more than 60 participants.

5. We have designed and compiled the Protein-Protein (version 5.0) and Protein-RNA (version 1.0) docking benchmarks, which are important resources for the community to test and to develop new methods against a reference set of curated cases.

6. A new protein-protein docking framework, LightDock, has been developed, aiming to be a versatile tool for the use of different scoring functions within a flexible-backbone model. Moreover, the framework can be easily extended to other interesting docking problems such as protein-DNA, protein-RNA and protein-peptide.

7. A set of good practices to try to tackle common problems regarding the development of scientific software have been proposed.

7. Bibliography

"How DARE you and the rest of your
barbarians set fire to my library?
Play conqueror all you want, Mighty Caesar!
Rape, murder, pillage thousands, even millions of human beings!
But neither you nor any other barbarian has the right to destroy
one human thought!"

William Shakespeare, *Julius Caesar*

Aad G, Abajyan T, Abbott B, Abdallah J, Abdel Khalek S, Abdelalim AA, et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. Phys Lett B. 2012 Sep;716(1):1–29.

Acuner Ozbabacan SE, Engin HB, Gursoy A, Keskin O. Transient protein-protein interactions. Protein Eng Des Sel. 2011 Sep;24(9):635–48.

Aebersold R, Hood LE, Watts JD. Equipping scientists for the new biology. Nat Biotechnol. 2000 Apr;18(4):359.

Alberts B. The cell as a collection of protein machines: preparing the next generation of molecular biologists. Cell. 1998 Feb 6;92(3):291–4.

Aloy P, Russell RB. The third dimension for protein interactions and complexes. Trends Biochem Sci. 2002 Dec;27(12):633–8.

Amir N, Cohen D, Wolfson HJ. DockStar: a novel ILP-based integrative method for structural modeling of multimolecular protein complexes. Bioinformatics. 2015 Sep 1;31(17):2801–7.

Anderson DP, Jeff C, Eric K, Matt L, Dan W. SETI@home: an experiment in public-resource computing. Commun ACM. 2002;45(11):56–61.

Andreani J, Faure G, Guerois R. InterEvScore: a novel coarse-grained interface scoring function using a multi-body statistical potential coupled to evolution. Bioinformatics. 2013 Jul 15;29(14):1742–9.

Andrusier N, Nussinov R, Wolfson HJ. FireDock: fast interaction refinement in molecular docking. Proteins. 2007 Oct 1;69(1):139–59.

Anishchanka I, Ivan A, Kundrotas PJ, Tuzikov AV, Vakser IA. Docking Benchmark Set of Protein Models. Biophys J. 2014;106(2):656a.

Assarzadeh Z, Naghsh-Nilchi AR. Chaotic particle swarm optimization with mutation for classification. J Med Signals Sens. 2015 Jan;5(1):12–20.

Azé J, Jérôme A, Thomas B, Sylvie H, Anne P, Ritchie DW. Using Kendall-τ Meta-Bagging to Improve Protein-Protein Docking Predictions. Lecture Notes in Computer Science. 2011. p. 284–95.

Bahadur RP, Chakrabarti P, Rodier F, Janin J. A dissection of specific and non-specific protein-protein interfaces. J Mol Biol. 2004 Feb 27;336(4):943–55.

Bahadur RP, Zacharias M. The interface of protein-protein complexes: analysis of contacts and prediction of interactions. Cell Mol Life Sci. 2008 Apr;65(7-8):1059–72.

Bai X-C, McMullan G, Scheres SHW. How cryo-EM is revolutionizing structural biology. Trends Biochem Sci. 2015 Jan;40(1):49–57.

Barnes N. Publish your computer code: it is good enough. Nature. 2010 Oct 14;467(7317):753.

Baxter SM, Day SW, Fetrow JS, Reisinger SJ. Scientific Software Development Is Not an Oxymoron. PLoS Comput Biol. 2006;2(9):e87.

Berendsen HJC, van der Spoel D, van Drunen R. GROMACS: A message-passing parallel molecular dynamics implementation. Comput Phys Commun. 1995;91(1-3):43–56.

Bernadó P. Low-resolution structural approaches to study biomolecular assemblies. Wiley Interdiscip Rev Comput Mol Sci. 2011;1(2):283–97.

Bernstein FC, Koetzle TF, Williams GJB, Meyer EF, Brice MD, Rodgers JR, et al. The protein data bank: A computer-based archival file for macromolecular structures. Arch Biochem Biophys. 1978;185(2):584–91.

Brenke R, Hall DR, Chuang G-Y, Comeau SR, Bohnuud T, Beglov D, et al. Application of asymmetric statistical potentials to antibody-protein docking. Bioinformatics. 2012 Oct 15;28(20):2608–14.

Castellani F, van Rossum B, Diehl A, Schubert M, Rehbein K, Oschkinat H. Structure of a protein determined by solid-state magic-angle-spinning NMR spectroscopy. Nature. 2002 Nov 7;420(6911):98–102.

Chaudhury S, Sircar A, Sivasubramanian A, Berrondo M, Gray JJ. Incorporating biochemical information and backbone flexibility in RosettaDock for CAPRI rounds 6-12. Proteins. 2007 Dec 1;69(4):793–800.

Chelliah V, Blundell TL, Fernández-Recio J. Efficient restraints for protein-protein docking by comparison of observed amino acid substitution patterns with those predicted from local environment. J Mol Biol. 2006 Apr 14;357(5):1669–82.

Cheng TM-K, Blundell TL, Fernandez-Recio J. pyDock: electrostatics and desolvation for effective scoring of rigid-body protein-protein docking. Proteins. 2007 Aug 1;68(2):503–15.

Chen H-M, Liu B-F, Huang H-L, Hwang S-F, Ho S-Y. SODOCK: swarm optimization for highly flexible protein-ligand docking. J Comput Chem. 2007 Jan 30;28(2):612–23.

Chen R, Mintseris J, Janin J, Weng Z. A protein-protein docking benchmark. Proteins. 2003 Jul 1;52(1):88–91.

Chen R, Rong C, Zhiping W. Docking unbound proteins using shape complementarity, desolvation, and electrostatics.

Proteins: Structure, Function, and Genetics. 2002;47(3):281–94.

Choura M, Rebaï A. Topological features of cancer proteins in the human NR-RTK interaction network. J Recept Signal Transduct Res. 2012 Oct;32(5):257–62.

Cipra B. How number theory got the best of the pentium chip. Science. 1995 Jan 13;267(5195):175.

Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009 Jun 1;25(11):1422–3.

Comeau SR, Gatchell DW, Vajda S, Camacho CJ. ClusPro: an automated docking and discrimination method for the prediction of protein complexes. Bioinformatics. 2004 Jan 1;20(1):45–50.

Csermely P, Palotai R, Nussinov R. Induced fit, conformational selection and independent dynamic segments: an extended view of binding events. Trends Biochem Sci. 2010 Oct;35(10):539–46.

Doerr A, Allison D. Structural biology: Cryo-EM strikes gold. Nat Methods. 2015;12(2):102–3.

Dominguez C, Cyril D, Rolf B, Bonvin AM. HADDOCK: A Protein−Protein Docking Approach Based on Biochemical or

Biophysical Information. J Am Chem Soc. 2003;125(7):1731–7.

Douguet D, Chen H-C, Tovchigrechko A, Vakser IA. DOCKGROUND resource for studying protein-protein interfaces. Bioinformatics. 2006 Nov 1;22(21):2612–8.

Duhovny D, Ruth N, Wolfson HJ. Efficient Unbound Docking of Rigid Molecules. Lecture Notes in Computer Science. 2002. p. 185–200.

Esquivel-Rodríguez J, Yang YD, Kihara D. Multi-LZerD: multiple protein docking for asymmetric complexes. Proteins. 2012 Jul;80(7):1818–33.

Fernández-Recio J, Totrov M, Abagyan R. ICM-DISCO docking by global energy optimization with fully flexible side-chains. Proteins. 2003 Jul 1;52(1):113–7.

Fields S, Song O. A novel genetic system to detect protein-protein interactions. Nature. 1989 Jul 20;340(6230):245–6.

Fischer D, Lin SL, Wolfson HL, Nussinov R. A geometry-based suite of molecular docking processes. J Mol Biol. 1995 Apr 28;248(2):459–77.

Fomel S, Sergey F, Claerbout JF. Guest Editors' Introduction: Reproducible Research. Comput Sci Eng. 2009;11(1):5–7.

Fraser HB. Evolutionary Rate in the Protein Interaction Network. Science. 2002;296(5568):750–2.

Freire J, Bonnet P, Shasha D. Computational reproducibility: state-of-the-art, challenges, and database research opportunities. Proceedings of the 2012 international conference on Management of Data - SIGMOD '12. New York, New York, USA: ACM Press; 2012. p. 593.

Gabb HA, Jackson RM, Sternberg MJ. Modelling protein docking using shape complementarity, electrostatics and biochemical information. J Mol Biol. 1997 Sep 12;272(1):106–20.

Gao Y, Douguet D, Tovchigrechko A, Vakser IA. DOCKGROUND system of databases for protein recognition studies: unbound structures for docking. Proteins. 2007 Dec 1;69(4):845–51.

Garzon JI, Lopéz-Blanco JR, Pons C, Kovacs J, Abagyan R, Fernandez-Recio J, et al. FRODOCK: a new approach for fast rotational protein-protein docking. Bioinformatics. 2009 Oct 1;25(19):2544–51.

George B, Boby G, Laurie W. A structured experiment of test-driven development. Information and Software Technology. 2004;46(5):337–42.

Gilbert W, Walter G. Towards a paradigm shift in biology. Nature. 1991;349(6305):99–99.

Goff SA, Ricke D, Lan T-H, Presting G, Wang R, Dunn M, et al. A draft sequence of the rice genome (Oryza sativa L. ssp. japonica). Science. 2002 Apr 5;296(5565):92–100.

González-Barahona JM, Gregorio R. On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. Empirical Software Engineering. 2011;17(1-2):75–89.

Hannaway O, Owen H, Steven S, Simon S. Leviathan and the Air-Pump: Hobbes, Boyle and the Experimental Life. Technol Cult. 1988;29(2):291.

Hermans J, Zhang L, Van Deusen C, Xia X. Hydrophilicity of cavities in proteins. Acta Crystallogr A. 1996;52(a1):C86–C86.

Hinsen K, Konrad H. A data and code model for reproducible research and executable papers. Procedia Comput Sci. 2011;4:579–88.

Hwang H, Howook H, Brian P, Julian M, Joël J, Zhiping W. Protein-protein docking benchmark version 3.0. Proteins: Struct Funct Bioinf. 2008;73(3):705–9.

Hwang H, Howook H, Thom V, Joël J, Zhiping W. Protein-protein docking benchmark version 4.0. Proteins: Struct Funct Bioinf. 2010;78(15):3111–4.

Inbar Y, Benyamini H, Nussinov R, Wolfson HJ. Combinatorial docking approach for structure prediction of large proteins

and multi-molecular assemblies. Phys Biol. 2005 Nov;2(4):S156–65.

Ince DC, Leslie H, John G-C. The case for open computer programs. Nature. 2012;482(7386):485–8.

Janin J, Henrick K, Moult J, Eyck LT, Sternberg MJE, Vajda S, et al. CAPRI: a Critical Assessment of PRedicted Interactions. Proteins. 2003 Jul 1;52(1):2–9.

Janin J. Welcome to CAPRI: A Critical Assessment of PRedicted Interactions. Proteins: Structure, Function, and Genetics. 2002;47(3):257–257.

Janzen D, Saiedian H. Test-driven development concepts, taxonomy, and future direction. Computer . 2005;38(9):43–50.

Jeong H, Mason SP, Barabási AL, Oltvai ZN. Lethality and centrality in protein networks. Nature. 2001 May 3;411(6833):41–2.

Jiménez-García B, Pons C, Fernández-Recio J. pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring. Bioinformatics. 2013 Jul 1;29(13):1698–9.

Joachimiak LA, Tanja K, Stoddard BL, David B. Computational Design of a New Hydrogen Bond Network and at Least a 300-fold Specificity Switch at a Protein−Protein Interface. J Mol Biol. 2006;361(1):195–208.

Jonsson PF, Bates PA. Global topological features of cancer proteins in the human interactome. Bioinformatics. 2006 Sep 15;22(18):2291–7.

Kanelis V, Voula K, Forman-Kay JD, Kay LE. Multidimensional NMR Methods for Protein Structure Determination. IUBMB Life. 2001;52(6):291–302.

Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. Nat Struct Biol. 2002 Sep;9(9):646–52.

Kasap S, Server K, Khaled B. Parallel Processor Design and Implementation for Molecular Dynamics Simulations on a FPGA-Based Supercomputer. Journal of Computers [Internet]. 2012;7(6). Available from: http://dx.doi.org/10.4304/jcp.7.6.1312-1328

Kastritis PL, Bonvin AMJJ. Are scoring functions in protein-protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. J Proteome Res. 2010 May 7;9(5):2216–25.

Kastritis PL, Moal IH, Hwang H, Weng Z, Bates PA, Bonvin AMJJ, et al. A structure-based benchmark for protein-protein binding affinity. Protein Sci. 2011 Mar;20(3):482–91.

Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA. Molecular surface recognition: determination of geometric fit between proteins and their ligands by

correlation techniques. Proc Natl Acad Sci U S A. 1992 Mar 15;89(6):2195–9.

Kauppinen T, Espindola GM de. Linked Open Science-Communicating, Sharing and Evaluating Data, Methods and Results for Executable Papers. Procedia Comput Sci. 2011;4:726–31.

Kenyon GL, DeMarini DM, Fuchs E, Galas DJ, Kirsch JF, Leyh TS, et al. Defining the mandate of proteomics in the post-genomics era: workshop report. Mol Cell Proteomics. 2002 Oct;1(10):763–80.

Khuri S, Wuchty S. Essentiality and centrality in protein interaction networks revisited. BMC Bioinformatics. 2015 Apr 1;16:109.

Kirys T, Ruvinsky AM, Singla D, Tuzikov AV, Kundrotas PJ, Vakser IA. Simulated unbound structures for benchmarking of protein docking in the DOCKGROUND resource. BMC Bioinformatics. 2015 Jul 31;16:243.

Kozakov D, Brenke R, Comeau SR, Vajda S. PIPER: an FFT-based protein docking program with pairwise potentials. Proteins. 2006 Nov 1;65(2):392–406.

Kühlbrandt W. Cryo-EM enters a new era. Elife. 2014 Aug 13;3:e03678.

Kundrotas PJ, Ivan A, Tuzikov AV, Vakser IA. Docking Benchmark Set of Protein Models. Biophys J. 2011;100(3):320a.

Kundrotas PJ, Zhu Z, Janin J, Vakser IA. Templates are available to model nearly all complexes of structurally characterized proteins. Proc Natl Acad Sci U S A. 2012 Jun 12;109(24):9438–41.

Landaverde R, Herbordt MC. GPU Optimizations for a Production Molecular Docking Code. IEEE Conf High Perform Extreme Comput [Internet]. 2014 Sep;2014. Available from: http://dx.doi.org/10.1109/HPEC.2014.7040981

Lander ES, Langridge R, Saccocio DM. Computing in molecular biology: mapping and interpreting biological information. Computer . 1991;24(11):6–13.

Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature. 2001 Feb 15;409(6822):860–921.

Lattman EE. Protein structure prediction: A special issue. Proteins: Structure, Function, and Genetics. 1995;23(3):i – i.

Launay G, Simonson T. A large decoy set of protein-protein complexes produced by flexible docking. J Comput Chem. 2011 Jan 15;32(1):106–20.

Lensink MF, Méndez R, Wodak SJ. Docking and scoring protein complexes: CAPRI 3rd Edition. Proteins. 2007 Dec 1;69(4):704–18.

Lensink MF, Moal IH, Bates PA, Kastritis PL, Melquiond ASJ, Karaca E, et al. Blind prediction of interfacial water positions in CAPRI. Proteins. 2014 Apr;82(4):620–32.

Lensink MF, Velankar S, Kryshtafovych A, Huang S-Y, Schneidman-Duhovny D, Sali A, et al. Prediction of homo- and hetero-protein complexes by protein docking and template-based modeling: a CASP-CAPRI experiment. Proteins [Internet]. 2016 Apr 28; Available from: http://dx.doi.org/10.1002/prot.25007

Lensink MF, Wodak SJ. Docking and scoring protein interactions: CAPRI 2009. Proteins. 2010 Nov 15;78(15):3073–84.

Lensink MF, Wodak SJ. Docking, scoring, and affinity prediction in CAPRI. Proteins. 2013 Dec;81(12):2082–95.

Lensink MF, Wodak SJ. Score_set: a CAPRI benchmark for scoring protein complexes. Proteins. 2014 Nov;82(11):3163–9.

Levinthal C, Wodak SJ, Kahn P, Dadivanian AK. Hemoglobin interaction in sickle cell fibers. I: Theoretical approaches to the molecular contacts. Proc Natl Acad Sci U S A. 1975 Apr;72(4):1330–4.

Liao M, Cao E, Julius D, Cheng Y. Structure of the TRPV1 ion channel determined by electron cryo-microscopy. Nature. 2013 Dec 5;504(7478):107–12.

Li S-H, Li X-J. Huntingtin-protein interactions and the pathogenesis of Huntington's disease. Trends Genet. 2004 Mar;20(3):146–54.

Liu S, Gao Y, Vakser IA. DOCKGROUND protein-protein docking decoy set. Bioinformatics. 2008 Nov 15;24(22):2634–5.

Liu S, Vakser IA. DECK: Distance and environment-dependent, coarse-grained, knowledge-based potentials for protein-protein docking. BMC Bioinformatics. 2011 Jul 11;12:280.

Liu S, Zhang C, Zhou H, Zhou Y. A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. Proteins. 2004 Jul 1;56(1):93–101.

Liu Y, Zhao L, Li W, Zhao D, Song M, Yang Y. FIPSDock: a new molecular docking technique driven by fully informed swarm optimization algorithm. J Comput Chem. 2013 Jan 5;34(1):67–75.

Li X, Moal IH, Bates PA. Detection and refinement of encounter complexes for protein-protein docking: taking account of macromolecular crowding. Proteins. 2010 Nov 15;78(15):3189–96.

Markram H, Henry M. The Human Brain Project. Sci Am. 2012;306(6):50–5.

Mashiach E, Nussinov R, Wolfson HJ. FiberDock: Flexible induced-fit backbone refinement in molecular docking. Proteins. 2010 May 1;78(6):1503–19.

May A, Zacharias M. Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein-protein docking. Proteins. 2008 Feb 15;70(3):794–809.

McCool M, Reinders J, Robison A. Structured Parallel Programming: Patterns for Efficient Computation. Elsevier; 2012.

Méndez R, Leplae R, De Maria L, Wodak SJ. Assessment of blind predictions of protein-protein interactions: current status of docking methods. Proteins. 2003 Jul 1;52(1):51–67.

Méndez R, Leplae R, Lensink MF, Wodak SJ. Assessment of CAPRI predictions in rounds 3-5 shows progress in docking procedures. Proteins. 2005 Aug 1;60(2):150–69.

Merali Z. Computational science: ...Error. Nature. 2010 Oct 14;467(7317):775–7.

Mintseris J, Julian M, Kevin W, Brian P, Robert A, Rong C, et al. Protein-protein docking benchmark 2.0: An update. Proteins: Struct Funct Bioinf. 2005;60(2):214–6.

Misa T, Thomas M. Understanding “How Computing Has Changed the World.” IEEE Ann Hist Comput. 2007;29(99):x3.

Moal IH, Bates PA. SwarmDock and the use of normal modes in protein-protein docking. Int J Mol Sci. 2010 Sep 28;11(10):3623–48.

Moal IH, Fernández-Recio J. SKEMPI: a Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models. Bioinformatics. 2012 Oct 15;28(20):2600–7.

Moal IH, Moretti R, Baker D, Fernández-Recio J. Scoring functions for protein-protein interactions. Curr Opin Struct Biol. 2013a Dec;23(6):862–7.

Moal IH, Torchala M, Bates PA, Fernández-Recio J. The scoring of poses in protein-protein docking: current capabilities and future directions. BMC Bioinformatics. 2013b Oct 1;14:286.

Moretti R, Fleishman SJ, Agius R, Torchala M, Bates PA, Kastritis PL, et al. Community-wide evaluation of methods for predicting the effect of mutations on protein-protein interactions. Proteins. 2013 Nov;81(11):1980–7.

Morin A, Urban J, Adams PD, Foster I, Sali A, Baker D, et al. Research priorities. Shining light into black boxes. Science. 2012 Apr 13;336(6078):159–60.

Mosca R, Céol A, Aloy P. Interactome3D: adding structural details to protein networks. Nat Methods. 2013 Jan;10(1):47–53.

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;5(8):e1000490.

Mugridge R. Test driven development and the scientific method. Proceedings of the Agile Development Conference, 2003 ADC 2003 [Internet]. 2003. Available from: http://dx.doi.org/10.1109/adc.2003.1231452

Namasivayam V, Günther R. pso@autodock: a fast flexible molecular docking program based on Swarm intelligence. Chem Biol Drug Des. 2007 Dec;70(6):475–84.

Narumi T, Susukita R, Koishi T, Yasuoka K, Furusawa H, Kawai A, et al. 1.34 Tflops Molecular Dynamics Simulation for NaCl with a Special-Purpose Computer: MDM. ACM/IEEE SC 2000 Conference (SC'00) [Internet]. 2000. Available from: http://dx.doi.org/10.1109/sc.2000.10016

Ng MCK, Fong S, Siu SWI. PSOVina: The hybrid particle swarm optimization algorithm for protein-ligand docking. J Bioinform Comput Biol. 2015 Jun;13(3):1541007.

Nooren IM and Thornton JM. NEW EMBO MEMBER'S REVIEW: Diversity of protein-protein interactions. EMBO J. 2003;22(14):3486–92.

Ofran Y, Rost B. Analysing six types of protein-protein interfaces. J Mol Biol. 2003 Jan 10;325(2):377–87.

Oti M. Predicting disease genes using protein-protein interactions. J Med Genet. 2006;43(8):691–8.

Pallara C, Jiménez-García B, Pérez-Cano L, Romero-Durana M, Solernou A, Grosdidier S, et al. Expanding the frontiers of protein-protein modeling: from docking and scoring to binding affinity predictions and other challenges. Proteins. 2013 Dec;81(12):2192–200.

Pearson K. The Problem of the Random Walk. Nature. 1905;72(1865):294–294.

Perkins JR, Diboun I, Dessailly BH, Lees JG, Orengo C. Transient protein-protein interactions: structural, functional, and network properties. Structure. 2010 Oct 13;18(10):1233–43.

Petoukhov MV, Svergun DI. Global rigid body modeling of macromolecular complexes against small-angle scattering data. Biophys J. 2005 Aug;89(2):1237–50.

Pierce B, Tong W, Weng Z. M-ZDOCK: a grid-based approach for Cn symmetric multimer docking. Bioinformatics. 2005 Apr 15;21(8):1472–8.

Pierce B, Weng Z. ZRANK: reranking protein docking predictions with an optimized energy function. Proteins. 2007 Jun 1;67(4):1078–86.

Pincus MR, Zimmerman SS, Scheraga HA. Prediction of three-dimensional structures of enzyme-substrate and enzyme-inhibitor complexes of lysozyme. Proc Natl Acad Sci U S A. 1976 Dec;73(12):4261–5.

Pons C, Carles P, Marco D, Svergun DI, Modesto O, Pau B, et al. Structural Characterization of Protein–Protein Complexes by Integrating Computational Docking with Small-angle Scattering Data. J Mol Biol. 2010;403(2):217–30.

Pons C, Jiménez-González D, González-Álvarez C, Servat H, Cabrera-Benítez D, Aguilar X, et al. Cell-Dock: high-performance protein-protein docking. Bioinformatics. 2012 Sep 15;28(18):2394–6.

Pons C, Talavera D, de la Cruz X, 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 Feb 28;51(2):370–7.

Popov P, Ritchie DW, Grudinin S. DockTrina: docking triangular protein trimers. Proteins. 2014 Jan;82(1):34–44.

Price D. Pentium FDIV flaw-lessons learned. IEEE Micro. 1995;15(2):86–8.

Puig O, Oscar P, Friederike C, Guillaume R, Berthold R, Emmanuelle B, et al. The Tandem Affinity Purification (TAP) Method: A General Procedure of Protein Complex Purification. Methods. 2001;24(3):218–29.

Rajovic N, Nikola R, Carpenter PM, Isaac G, Nikola P, Alex R, et al. Supercomputing with commodity CPUs. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis on - SC '13 [Internet]. 2013. Available from: http://dx.doi.org/10.1145/2503210.2503281

Ravikumar KM, Huang W, Yang S. Coarse-grained simulations of protein-protein association: an energy landscape perspective. Biophys J. 2012 Aug 22;103(4):837–45.

Rigaut G, Shevchenko A, Rutz B, Wilm M, Mann M, Séraphin B. A generic protein purification method for protein complex characterization and proteome exploration. Nat Biotechnol. 1999 Oct;17(10):1030–2.

Ritchie DW. Recent progress and future directions in protein-protein docking. Curr Protein Pept Sci. 2008 Feb;9(1):1–15.

Ritchie DW, Kemp GJ. Protein docking using spherical polar Fourier correlations. Proteins. 2000 May 1;39(2):178–94.

Ritchie DW, Venkatraman V. Ultra-fast FFT protein docking on graphics processors. Bioinformatics. 2010 Oct 1;26(19):2398–405.

Rolland T, Taşan M, Charloteaux B, Pevzner SJ, Zhong Q, Sahni N, et al. A proteome-scale map of the human interactome network. Cell. 2014 Nov 20;159(5):1212–26.

Rual J-F, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, Li N, et al. Towards a proteome-scale map of the human protein-protein interaction network. Nature. 2005 Oct 20;437(7062):1173–8.

Sandak B, Nussinov R, Wolfson HJ. A method for biomolecular structural recognition and docking allowing conformational flexibility. J Comput Biol. 1998 Winter;5(4):631–54.

Schneidman-Duhovny D, Hammel M, Tainer JA, Sali A. Accurate SAXS profile computation and its assessment by contrast variation experiments. Biophys J. 2013 Aug 20;105(4):962–74.

Schneidman-Duhovny D, Inbar Y, Nussinov R, Wolfson HJ. PatchDock and SymmDock: servers for rigid and symmetric docking. Nucleic Acids Res. 2005 Jul 1;33(Web Server issue):W363–7.

Schneidman-Duhovny D, Kim SJ, Sali A. Integrative structural modeling with small angle X-ray scattering profiles. BMC Struct Biol. 2012a Jul 16;12:17.

Schneidman-Duhovny D, Rossi A, Avila-Sakar A, Kim SJ, Velázquez-Muriel J, Strop P, et al. A method for integrative

structure determination of protein-protein complexes. Bioinformatics. 2012b Dec 15;28(24):3282–9.

Schueler-Furman O, Wang C, Baker D. Progress in protein-protein docking: atomic resolution predictions in the CAPRI experiment using RosettaDock with an improved treatment of side-chain flexibility. Proteins. 2005 Aug 1;60(2):187–94.

Shaw DE. Anton: A Specialized Machine for Millisecond-Scale Molecular Dynamics Simulations of Proteins. 2009 19th IEEE Symposium on Computer Arithmetic [Internet]. 2009. Available from: http://dx.doi.org/10.1109/arith.2009.33

Shaw DE. Anton. Proceedings of the 22nd international symposium on High-performance parallel and distributed computing - HPDC '13 [Internet]. 2013. Available from: http://dx.doi.org/10.1145/2493123.2465528

Sibille N, Bernadó P. Structural characterization of intrinsically disordered proteins by the combined use of NMR and SAXS. Biochem Soc Trans. 2012 Oct;40(5):955–62.

Stelzl U, Worm U, Lalowski M, Haenig C, Brembeck FH, Goehler H, et al. A human protein-protein interaction network: a resource for annotating the proteome. Cell. 2005 Sep 23;122(6):957–68.

Stodden V. The Scientific Method in Practice: Reproducibility in the Computational Sciences. SSRN Electronic Journal

[Internet]. Available from: http://dx.doi.org/10.2139/ssrn.1550193

Stumpf MPH, Thorne T, de Silva E, Stewart R, An HJ, Lappe M, et al. Estimating the size of the human interactome. Proc Natl Acad Sci U S A. 2008 May 13;105(19):6959–64.

Sukhwani B, Bharat S, Herbordt MC. GPU acceleration of a production molecular docking code. Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units - GPGPU-2 [Internet]. 2009. Available from: http://dx.doi.org/10.1145/1513895.1513898

Sun J, Jingchun S, Peilin J, Zhongming Z. Global Network Features of Cancer Genes in the Human Interactome. 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing [Internet]. 2009. Available from: http://dx.doi.org/10.1109/ijcbs.2009.62

Sun J, Zhao Z. A comparative study of cancer proteins in the human protein-protein interaction network. BMC Genomics. 2010 Dec 1;11 Suppl 3:S5.

Szilagyi A, Zhang Y. Template-based structure modeling of protein-protein interactions. Curr Opin Struct Biol. 2014 Feb;24:10–23.

Taiji M, Narumi T, Ohno Y, Konagaya A. MDGRAPE-3: A petaflops special-purpose computer system for molecular

dynamics simulations. Advances in Parallel Computing. 2004. p. 669–76.

The Janus Collaboration, Collaboration TJ. Janus2. Proceedings of the Future HPC Systems on the Challenges of Power-Constrained Performance - FutureHPC '12 [Internet]. 2012. Available from: http://dx.doi.org/10.1145/2322156.2322158

Tobi D. Designing coarse grained-and atom based-potentials for protein-protein docking. BMC Struct Biol. 2010 Nov 15;10:40.

Vakser IA. Evaluation of GRAMM low-resolution docking methodology on the hemagglutinin-antibody complex. Proteins. 1997;Suppl 1:226–30.

Vakser IA. Low-resolution structural modeling of protein interactome. Curr Opin Struct Biol. 2013 Apr;23(2):198–205.

Varma BSC, Sharat Chandra Varma B, Kolin P, Balakrishnan M. Accelerating 3D-FFT Using Hard Embedded Blocks in FPGAs. 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems [Internet]. 2013. Available from: http://dx.doi.org/10.1109/vlsid.2013.169

Varma BSC, Sharat Chandra Varma B, Kolin P, Balakrishnan M. FPGA-Based Acceleration of Protein Docking. Springer Series in Advanced Microelectronics. 2016. p. 39–54.

Venkatesan K, Rual J-F, Vazquez A, Stelzl U, Lemmens I, Hirozane-Kishikawa T, et al. An empirical framework for

binary interactome mapping. Nat Methods. 2009 Jan;6(1):83–90.

Vreven T, Thom V, Moal IH, Anna V, Pierce BG, Kastritis PL, et al. Updates to the Integrated Protein–Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. J Mol Biol. 2015;427(19):3031–41.

de Vries SJ, Chauvot de Beauchêne I, Schindler CEM, Zacharias M. Cryo-EM Data Are Superior to Contact and Interface Information in Integrative Modeling. Biophys J. 2016 Feb 23;110(4):785–97.

Wooley JC. CYBERINFRASTRUCTURE FOR THE BIOLOGICAL SCIENCES (CIBIO). Grid Computing in Life Sciences [Internet]. 2006. Available from: http://dx.doi.org/10.1142/9789812772503_0002

Wooley JC, Lin HS, editors. Catalyzing Inquiry at the Interface of Computing and Biology. Washington (DC): National Academies Press (US); 2010.

Xia B, Mamonov A, Leysen S, Allen KN, Strelkov SV, Paschalidis IC, et al. Accounting for observed small angle X-ray scattering profile in the protein-protein docking server ClusPro. J Comput Chem. 2015 Jul 30;36(20):1568–72.

Zacharias M. Protein-protein docking with a reduced protein model accounting for side-chain flexibility. Protein Sci. 2003 Jun;12(6):1271–82.

Zhang Y. Progress and challenges in protein structure prediction. Curr Opin Struct Biol. 2008 Jun;18(3):342–8.

Zyout I, Czajkowska J, Grzegorzek M. Multi-scale textural feature extraction and particle swarm optimization based model selection for false positive reduction in mammography. Comput Med Imaging Graph. 2015 Dec;46 Pt 2:95–107.

List of publications and thesis advisor report

The PhD thesis of Brian Jiménez García has produced nine scientific articles, six of them as first author. Five of these articles have been already published in international peer-reviewed journals with impact factor between 2.627 and 9.112 (as indexed in ISI). One more article is soon to be submitted. Here is a list of all the articles to which Brian Jiménez García has contributed during the PhD thesis. Only the articles in bold are part of the thesis. Authors marked with # contributed equally to the work.

A protein-RNA docking benchmark (II): Extended set from experimental and homology modeling data
By: Perez-Cano, Laura; Jimenez-Garcia, Brian; Fernandez-Recio, Juan
PROTEINS-STRUCTURE FUNCTION AND BIOINFORMATICS
Volume: 80 Issue: 7 Pages: 1872-1882 Published: JUL 2012.
(Impact factor: 2.627; 13 citations)

Integration of protein-protein docking tools for multi-scale approach to complex structural prediction
By: Jimenez-Garcia, Brian; Fernandez-Recio, Juan
Conference: 22nd IUBMB Congress/37th FEBS Congress
Location: Seville, SPAIN Date: SEP 04-09, 2012
Sponsor(s): IUBMB; FEBS

FEBS JOURNAL Volume: 279 Special Issue: SI Supplement: 1 Pages: 532-532 Published: SEP 2012 (Impact factor: 4.001; 0 citations)

pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring

By: Jimenez-Garcia, Brian; Pons, Carles; Fernandez-Recio, Juan

BIOINFORMATICS Volume: 29 Issue: 13 Pages: 1698-1699 Published: JUL 1 2013 (Impact factor: 4.981; 34 citations)

Expanding the frontiers of protein-protein modeling: From docking and scoring to binding affinity predictions and other challenges

By: Pallara, Chiara #; Jimenez-Garcia, Brian #; Perez-Cano, Laura; et al.

PROTEINS-STRUCTURE FUNCTION AND BIOINFORMATICS Volume: 81 Issue: 12 Special Issue: SI Pages: 2192-2200 Published: DEC 2013 (Impact factor: 2.627; 9 citations)

Blind prediction of interfacial water positions in CAPRI

By: Lensink, Marc F.; Moal, Iain H.; Bates, Paul A.; et al.

PROTEINS-STRUCTURE FUNCTION AND BIOINFORMATICS Volume: 82 Issue: 4 Pages: 620-632 Published: APR 2014 (Impact factor: 2.627; 15 citations)

CCharPPI web server: computational characterization of protein-protein interactions from structure

By: Moal, Iain H. #; Jimenez-Garcia, Brian #; Fernandez-Recio, Juan

BIOINFORMATICS Volume: 31 Issue: 1 Pages: 123-125 Published: JAN 1 2015 (Impact factor: 4.981; 3 citations)

pyDockSAXS: protein-protein complex structure by SAXS and computational docking

By: Jimenez-Garcia, Brian; Pons, Carles; Svergun, Dmitri I.; et al.

NUCLEIC ACIDS RESEARCH Volume: 43 Issue: W1 Pages: W356-W361 Published: JUL 1 2015 (Impact factor: 9.112; 7 citations)

Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2

By: Vreven, Thom; Moal, Iain H.; Vangone, Anna; et al.

JOURNAL OF MOLECULAR BIOLOGY Volume: 427 Issue: 19 Pages: 3031-3041 Published: SEP 25 2015 (Impact factor: 4.333; 4 citations)

Community-wide evaluation of methods for predicting the effect of mutations on protein-protein interactions

By: Moretti, Rocco; Fleishman, Sarel J.; Agius, Rudi; et al.
PROTEINS-STRUCTURE FUNCTION AND BIOINFORMATICS
Volume: 81 Issue: 11 Pages: 1980-1987 Published: NOV 2013 (Impact factor: 2.627; 46 citations)

Prediction of homo- and hetero- protein complexes by protein docking and template-based modeling: a CASP-CAPRI experiment
By: Mark Lensink et al. (102 authors in total including Jimenez-Garcia, Brian)
PROTEINS-STRUCTURE FUNCTION AND BIOINFORMATICS
IN PRESS Published: APR 2016 (Impact factor: 2.627; 0 citations)

Congress contributions

Posters

2012

1. Jiménez-García B, Pons C, Fernández-Recio J. (2012) **pyDockWEB: a new web-server for energy-based protein-protein docking**. XII Congress SBE. Barcelona (Spain).
2. Jiménez-García B, Fernández-Recio J. (2012) **Integration of protein-protein docking tools for multi-scale approach to complex structural prediction.** IUBMB-FEBS. Seville (Spain).

2013

3. Jiménez-García B, Pallara C, Triki D, Fernández-Recio J. (2013) **PyDock version 3: improvements for high-performance docking and general applicability for non-peptidic molecules**. CAPRI 5th. Utrecht (Netherlands).

2015

4. Jiménez-García B, Fernández-Recio J. (2015) **LightDock: a novel protein-protein docking**

framework for the new challenges in the interactomics era. SEBBM2015. Valencia (Spain).

2016

5. Jiménez-García B, Fernández-Recio J. (2016) **High-performance computational tools for the characterization of protein-protein interactions.** BIFI2016. Zaragoza (Spain).
6. Jiménez-García B, Roel J, Fernández-Recio J. (2016) **LightDock: a novel protein-protein docking framework for the new challenges in the interactomics era.** 6th CAPRI evaluation meeting. Tel-Aviv (Israel).

Oral communications

2013

1. Jiménez-García B. (2013) **Computational approaches to protein-protein docking**. National Institute of Biomedical Innovation. Osaka (Japan).

2014

2. Jiménez-García B. (2014) **Swarm intelligence**. Jornada d'Investigadors Predoctorals Interdisciplinària (JIPI). Barcelona (Spain).
3. Jiménez-García B. (2014) **Development and optimization of high-performance computational tools for protein-protein docking**. BSC Days. Barcelona (Spain).

2015

4. Jiménez-García B. (2015) **Development and optimization of high-performance computational tools for protein-protein docking**. Life Sciences Seminars (BSC). Barcelona (Spain).
5. Jiménez-García B. (2015) **LightDock: a novel protein-protein docking framework for the new challenges in the interactomics era**. III Bioinformatics and Computational Biology Symposium (BIB). Barcelona (Spain).