Faraday Discussions

Cite this: *Faraday Discuss.*, 2014, **169**, 425

PAPER

Udock, the interactive docking entertainment system

Guillaume Levieux,*[a] Guillaume Tiger,[a] Stéphanie Mader,[a] Jean-François Zagury,[b] Stéphane Natkin[a] and Matthieu Montes*[b]

Received 18th December 2013, Accepted 31st January 2014
DOI: 10.1039/c3fd00147d

Protein–protein interactions play a crucial role in biological processes. Protein docking calculations' goal is to predict, given two proteins of known structures, the associate conformation of the corresponding complex. Here, we present a new interactive protein docking system, Udock, that makes use of users' cognitive capabilities added up. In Udock, the users tackle simplified representations of protein structures and explore protein–protein interfaces' conformational space using a gamified interactive docking system with on the fly scoring. We assumed that if given appropriate tools, a naïve user's cognitive capabilities could provide relevant data for (1) the prediction of correct interfaces in binary protein complexes and (2) the identification of the experimental partner in interaction among a set of decoys. To explore this approach experimentally, we conducted a preliminary two week long playtest where the registered users could perform a cross-docking on a dataset comprising 4 binary protein complexes. The users explored almost all the surface of the proteins that were available in the dataset but favored certain regions that seemed more attractive as potential docking spots. These favored regions were located inside or nearby the experimental binding interface for 5 out of the 8 proteins in the dataset. For most of them, the best scores were obtained with the experimental partner. The alpha version of Udock is freely accessible at http://udock.fr.

Introduction

Protein–protein interactions play a crucial role in biological processes. The prediction of the geometry of protein complexes is a difficult task that has been a goal of computational chemistry. Much effort has been invested in the last few decades to develop docking methods, their performance being assessed during the CAPRI experiment.[1] Most of the available protein docking methods explore the conformational space between the proteins to be docked; this exploration

[a]*Equipe Interactivité pour Lire et Jouer, Laboratoire CEDRIC, EA4626, Conservatoire National des Arts et Métiers, 292 Rue Saint Martin, 75003 Paris. E-mail: guillaume.levieux@cnam.fr*
[b]*Laboratoire Génomique Bioinformatique et Applications, EA4627, Conservatoire National des Arts et Métiers, 292 Rue Saint Martin, 75003 Paris. E-mail: matthieu.montes@cnam.fr*

being based either on fast Fourier transform correlations,[2–5] Monte Carlo sampling[6,7] or driven by biochemical or physical information.[8] Recent developments in haptic devices allowed the emergence of interactive molecular dynamics approaches[9,10] enabling systems' simulation while receiving real-time feedback.

To date, there is no protein docking method available that allows a quick and interactive handling of the proteins to be docked to perform human driven exploration of the protein–protein interfaces. To our knowledge, the closest attempt towards this goal was the prototype of DockingShop[11] that seems to be no longer in development. The computational chemistry research field might benefit from intuitive and interactive tools[10] that would lead to a rapid gain in general knowledge on the problem, or get new ideas by trial and error exploration.

Moreover, it might be useful to have even non-experts, so-called naïve users, use these kinds of tools. Indeed, the protein docking problem can be considered as a complex 3D shapes combination problem. Humans beings are intuitively good at shape recognition and abstraction,[12] and if given appropriate tools, even naïve users could intuitively propose appropriate solutions of complex problems[13] such as protein–protein interfaces by steered trial and error exploration.

Resolving protein–protein interaction challenges can foster non-expert users' motivation because it inherently provides what is needed to create a good video game: a goal-directed task that is, according to Atari's founder Bushnell's quote "both easy to learn and very hard to master".[14]

Here, we present the first version of an interactive docking system, Udock, that would allow a quick and easy-to-handle exploration of the possible conformations of a protein complex. First, we will describe protein animation and rendering with Udock, starting from a standard molecular description file until integration into a video game physics engine. We will also present our choices with regard to binding energy calculation. Then, we explain how we simplified the protein structure representation and the docking process task, so that we allow even naïve users to perform interactive docking. As a preliminary assessment of our approach, we present the results of a two week playtest: a user-based interactive cross-docking experiment on 8 proteins, with a limited number of users that have tried to reach the best binding score for each out of the 36 possible protein complexes.

Methods

Different steps are needed in Udock before allowing users to perform interactive molecular docking: preprocessing of the coordinate files, generation of the solvent excluded surfaces (SES), and smoothed coloration of the SES according to the atomic partial charges.

Preprocessing of the coordinate files

Udock uses protonated protein mol2 files with atomic partial charges computed using AMBER12 force-field.[15] To generate such files from protein PDB files, we use the dockprep procedure as implemented in Chimera[16] using default parameters.

Generation of the SES

Once mol2 files have been generated, they can be loaded by Udock. To generate a 3D mesh out of the protein mol2 file, we use a marching cubes algorithm as

described in ref. 17. For every atom in the mol2 file, we first generate the solvent accessible surface (SAS) using the sum of the atom radii and a 1.4 Å radius probe. Then, we roll the probe whose center is at the generated surface and remove all the cubes that the probe intersects, and thus obtain the solvent excluded surface (SES).

Coloration of the SES

The color of the SES surface mesh is used to represent the electrostatic potential at the surface. For every point of the surface, we calculate the mean of all the atomic partial charges within a 5 Å radius sphere. Each atom's partial charge is divided by the squared distance to a point located 1.4 Å above the surface point we calculate. We thus represent a smoothed approximation of the electrostatic potential of the protein. Fig. 1 shows the difference between the smoothed electrostatic potential displayed on the surface and the unsmoothed, basic output of atomic partial charges displayed on the surface. We use a pixel shader script to enhance the readability of the SES, mainly by using black contour lines to enhance the perception of the molecule's shape. Following the graphic chart provided by our graphical designer, neutral parts of the proteins are not white but rather a light blue color. Then, we slowly reach a strong blue or red color to indicate respectively positively and negatively charged areas.

Rendering and interaction

Once the colored SES is generated, its mesh is processed by an open source video game physics engine, Bullet,[18] to generate a collision mesh. We use the physics engine to handle the user interaction with the molecule: when the user clicks on the molecule, we apply 3D forces on the mesh based on the mouse input, and let the physic engine calculate the subsequent orientation of the molecule. To give the feeling of a molecule immersed in a solvent and facilitate interaction, we dampen angular speed and velocity so that if the user stops interacting with a protein, it takes exactly one second for it to stop moving. Moreover, the physics engine is also responsible for calculating and taking into account the collisions between the molecules. As a result, users do not have to take clashes into account when they try to dock a protein on the other one.

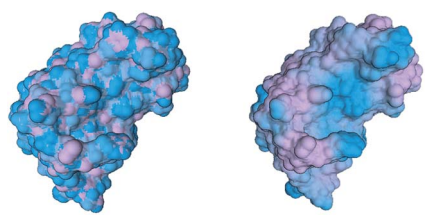

Fig. 1 Illustration of the smoothed approximation of the electrostatic potential on barnase (PDB id: 1brs, chain a). Left: atomic partial charges as computed by amber12 displayed on the solvent excluded surface. Right: our smoothed approximation of the electrostatic potential displayed on the SES.

A grapnel-based interface to perform user-steerable interactive docking

To make interactive docking a naïve-user-steerable task, we decided to use a grapnel-based representation. The users interactively select protein SES locations on which they will attach grapnels. Any number of grapnels can be attached between the SES of the proteins that will be docked. At any moment, the user can apply an attractive force on the grapnels to reduce their length gradually and put the proteins in contact. The sum of torque values applied by the grapnels on the proteins is monitored and adjusted on the fly in order to let the user orient the proteins as he wishes before collision occurs. Thanks to the physics engine, when proteins collide, no clashes between the atoms of the ligand and the atoms of the receptor are possible.

During the whole procedure, a force-field based intermolecular energy score is computed and displayed on the fly in the interface. The scoring function includes a soft van der Waals term for contacts and a Coulombic term for electrostatics. We used a distance dependent dielectric constant of $\varepsilon_0 = 20$ that resulted in balanced contributions of the different terms of the scoring function. The detailed form of the scoring function for the interaction energy of the atom pair i, j at distance r_{ij} is detailed below:

$$\text{Score} = -\left(-\frac{A_{ij}}{{r_{ij}}^6} + \frac{B}{{r_{ij}}^8} + f\frac{q_i q_j}{\varepsilon_0 r_{ij}}\right)$$

with q_i and q_j atomic partial charges of atoms i and j computed using AMBER12[15] as implemented in Chimera's dockprep.[16] A_{ij} and B_{ij} are, respectively, repulsive and attractive Lennard-Jones type parameters. f is a conversion factor for converting the electrostatics term to kcal mol^{-1}. We used $f = 332.0522$ according to the AMBER12 documentation.[15]

At any time, the user can launch a five second Monte Carlo rigid body optimization of the complex. The number of Monte Carlo steps that will be performed during the optimization procedure will depend on the number of atoms interacting in the system and the power of the CPU of the users' computer. For example, evaluation of the scoring function on a pair of atoms takes 19.5 ns on an Intel Core i7 3930 K (3.2G Hz), which corresponds to 300 Monte Carlo optimization steps of a contact between barnase and barstar within five seconds. During this Monte Carlo optimization procedure, the physics engine is switched off, allowing closer and more accurate contacts if favored by the scoring function. Indeed, the use of a soft repulsive term in the van der Waals part of the scoring function will allow (but still penalize) the existence of small clashes to simulate a pseudo-plasticity of the residues in the interface.

A detailed flowchart of Udock preprocessing and interactive docking is presented in Fig. 2.

Udock alpha version playtest

The duration of the online Udock alpha version test was set to two weeks during which the users could explore freely a test set of 4 binary complexes detailed below. The users were mostly computer science students and co-workers. Statistics on the users' age, play frequency and previous knowledge on structural biology were performed based on surveys upon registration.

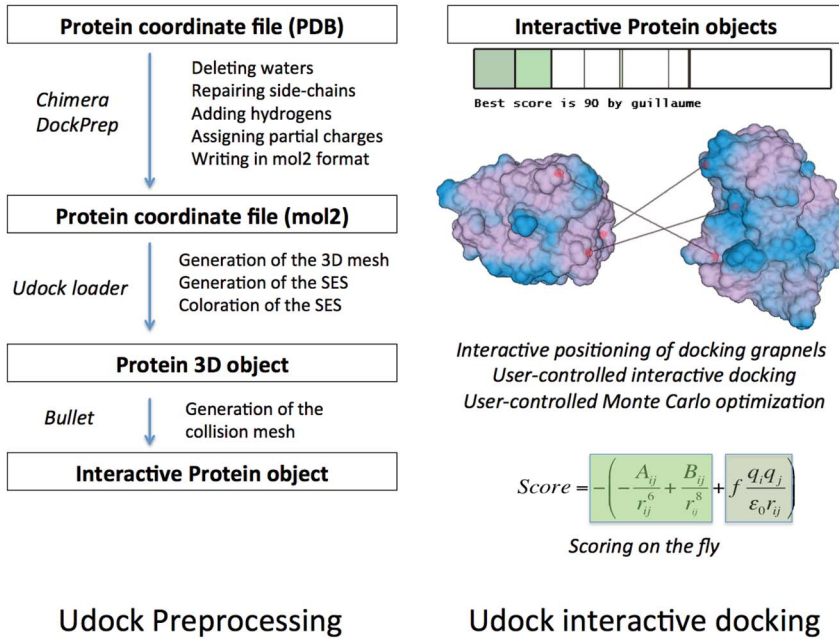

Fig. 2 Flowchart of Udock preprocessing and interactive docking.

Construction of the test set

The users explored four binary enzyme–inhibitor complexes used in the pioneer cross-docking experiment of Sacquin-Mora *et al.*,[19] namely barnase/barstar (PDB ID: 1BRS), acetylcholinesterase/fasciculin II (PDB ID: 1FSS), thermitase/eglin c (PDB ID: 2TEC) and CDC42 GTPase/CDC42 GAP (PDB ID: 1GRN) which led to 36 possible complexes to be explored by the users. In order to prevent the users from using external information about the proteins or their geometry, we anonymized the proteins in the dataset by giving them random names as detailed in Table 1. The proteins of the dataset vary in size and complexity, as can be seen in Fig. 3.

Table 1 Summary of the protein complexes investigated in the study. #residues is the number of residues of the corresponding protein. Throughout this work, we refer to the proteins by their index, given in the last column

PDB ID_Chain	Protein	Random name	#residues	Index
1BRS_A	Barnase	Dwaylith	110	**1**
1BRS_D	Barstar	Cilan	89	**2**
1FSS_A	Acetylcholinesterase	Eralg	535	**3**
1FSS_B	Fasciculin II	Taurith	61	**4**
1GRN_A	CDC42 GTPase	Bisil	200	**5**
1GRN_B	CDC42 GAP	Prok	199	**6**
2TEC_E	Thermitase	Etinna	279	**7**
2TEC_I	Eglin c	Bloc	63	**8**

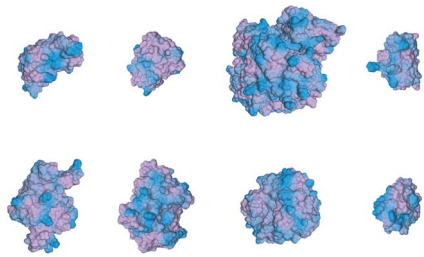

Fig. 3 Proteins of the dataset, as rendered by the Udock engine. From top left to bottom right: barnase (**1**), barstar (**2**), acetylcholinesterase (**3**), fasciculin II (**4**), CDC42 GTPase (**5**), CDC42 GAP (**6**), thermitase (**7**), eglin c (**8**).

Udock alpha version playtest users' statistics

42 users registered to Udock and played for a total of 25 h. 27 out of the 42 registered users played at least 5 min and among them only 12 played at least 30 min. The cumulated time spent by the users exploring the geometry of the 36 different complexes varied from 10 to 87 min (see Table 2). The users were on

Table 2 Cumulated time (in minutes) spent by the users on the exploration of the geometry of the 36 different possible complexes in the cross-docking dataset. A gradient of color has been applied from the less explored complexes (yellow) to the most explored complexes (dark green)

Index	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	*47*	*66*	*28*	*31*	*24*	*14*	*27*	*17*
2	*66*	*13*	*24*	*37*	*40*	*12*	*36*	*16*
3	*28*	*24*	*12*	*21*	*10*	*54*	*31*	*14*
4	*31*	*37*	*21*	*25*	*33*	*27*	*46*	*37*
5	*24*	*40*	*10*	*33*	*27*	*43*	*26*	*29*
6	*14*	*12*	*54*	*27*	*43*	*19*	*31*	*16*
7	*27*	*36*	*31*	*46*	*26*	*31*	*40*	*87*
8	*17*	*16*	*14*	*37*	*29*	*16*	*87*	*25*

average 31 years old ($\sigma = 6.7$). 26 out of the 42 registered users were frequent players. Most of the users (37 out of 42) were naïve in structural biology.

Determination of the solvent accessible surface (SAS)

To determine the SAS value of each atom, we used the marching cubes algorithm as described in ref 17 using a 1.4 Å radius probe added to each atom radius and 0.4 Å wide cubes. For each atom, we recorded the number of polygons generated as an approximation of the SAS value.

Determination of the interface atoms

All the atoms of a given protein within 4 Å of any atom of the interacting protein were considered as interface atoms.

Generation of the exploration maps

To describe the users' exploration of each possible complex of the dataset, we generated exploration maps for each of the 8 proteins of the cross-docking dataset. We logged all user-explored interface atoms every time a user called for the Monte Carlo optimization process at a specific position. Exploration maps were generated using the user-explored interface atoms' polar coordinates, θ and ϕ as follows:

$$x = \frac{\theta w}{\pi} \text{and } y = \frac{\phi h}{2\pi} \text{ with } w \text{ and } h \text{ being the width and height of the image space.}$$

Definition of the experimental interface covering ratio

Experimental interface covering ratio (CR) was logged every time a user called for the Monte Carlo optimization procedure and was defined as follows:

$$\mathrm{CR} = \frac{|\mathrm{cia} \cap \mathrm{cei}|}{|\mathrm{cei}|}$$

where cia is the set of atoms in the current interface and cei, the set of atoms in the experimental interface.

Results

Udock alpha version playtest

To establish the proof of concept of Udock, we proceeded to a two week duration alpha playtest consisting of a small cross-docking experiment on 4 binary complexes, which led to 36 possible complexes to be explored by the users.

Users exploration of the dataset

After analyzing the data provided by the users during the Udock alpha playtest, we generated exploration maps for each protein of the dataset presented in Fig. 4. According to the exploration maps, we observe that the users did explore, at least one time, each atom of the experimental interface. Very few atoms of the experimental interfaces of proteins **2**, **3** and **5** have never been explored by the users during the playtest (displayed in blue in the exploration maps).

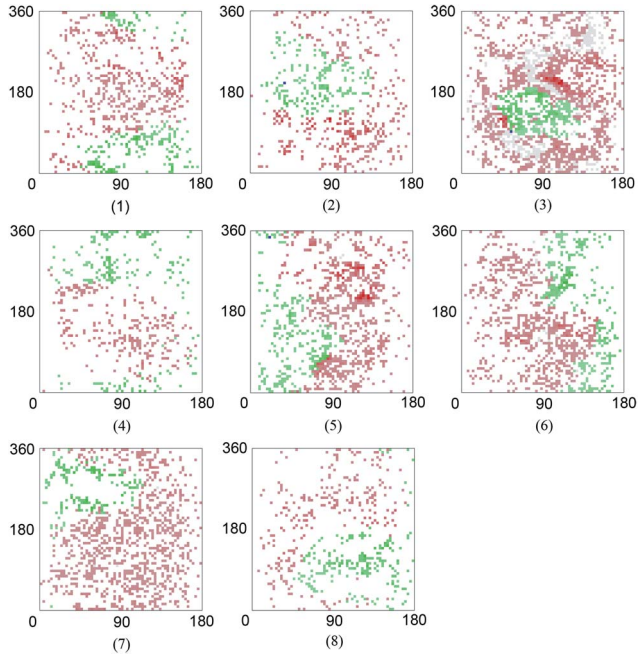

Fig. 4 Exploration maps generated for the 8 proteins of the dataset (index **1** to **8**) with the polar coordinates θ and ϕ along the horizontal and vertical axis, respectively. User-explored interface atoms within the experimental interface are displayed in green. User-explored interface atoms not within the experimental interface are displayed in red. Atoms within the experimental interface that have not been explored by the users are displayed in blue. Atoms not within the experimental interface that have not been explored by the users are displayed in grey. A gradient of grey (from light grey to dark grey) has been applied to these atoms depending on their corresponding normalized atomic SAS value. A gradient of darkness was applied to the user-explored interface atom corresponding color depending on the frequency of their exploration (lightest color: least explored interface atom; darkest color: most explored interface atom).

To enrich the information given by the exploration maps, we detailed, for each protein of the dataset, the frequency of the amount of exploration of a given atom (see Fig. 5). We wanted to highlight whether the users explored more intensely specific parts of the surface as dark red-colored surface areas of proteins **3** or **5** tend to show in their exploration maps. As expected, the frequency of the explored atoms all along the surface is not uniform, since some regions were more intensively explored. The difference of exploration in the surface points along proteins **3**, **6** and **7** was particularly striking, with a very small number of highly explored atoms and a very high number of not-highly explored atoms. For the other proteins in the dataset, the exploration of the surface atoms was more uniform.

We decided to plot the frequency of the experimental interface covering ratio (CR) to quantify how much the users explored within the experimental interfaces of each protein during Udock alpha playtests. The frequency of CR for each protein towards the entire dataset is presented in Fig. 6. The frequency of CR for each protein towards its corresponding experimental partner in the dataset is

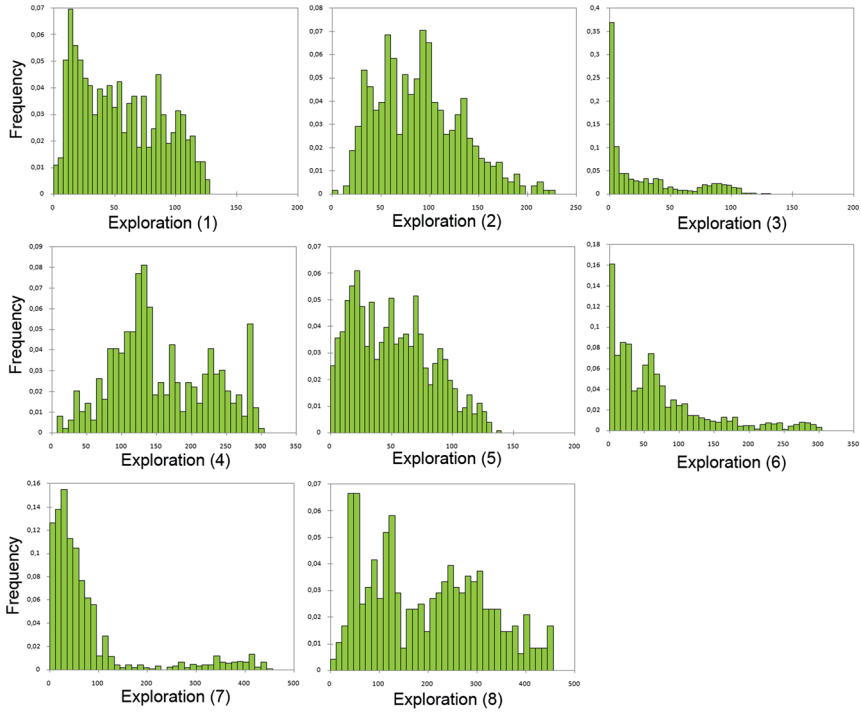

Fig. 5 Frequency of the amount of exploration for a given atom for each protein of the dataset (index **1** to **8**). An interface atom is considered as explored every time the user calls the Monte Carlo optimization procedure. The atoms with normalized SAS = 0 were not included.

presented in Fig. 7. Proteins **4** and **8** were particularly explored in the experimental binding interface whatever the protein that was involved in the complex (either the experimental partner or a decoy). The users could successively identify the experimental binding interface for the complexes acetylcholinesterase/fasciculin II (**3**,**4**) and thermitase/eglin c (**7**,**8**). They could identify the experimental interface for CDC42 GAP (**6**) but not for CDC42 GTPase (**5**).

High-scores obtained by the users during the playtest

During the Udock alpha playtest, we recorded all the best scores obtained by the users for every possible complex of the cross-docking dataset. The mean of the 3 best high scores obtained by the users for each protein with every other protein of the dataset is presented in Fig. 8. The score resulting from the rescoring with the Udock engine of the experimental geometry observed in the original PDB is also provided as an indication of a high score that could have been attained by the users in these particular cases. Except for proteins **7** and **8**, the best scores obtained by the users were far from the score they could have attained if they could reproduce the exact geometry observed in the PDB file. They found the highest score for the experimental partner for half of the proteins in the dataset, namely acetylcholinesterase/fascivulin II (**3**,**4**) and thermitase/eglin c (**7**,**8**). For barnase/barstar (**1**,**2**) and CDC42 GTPase/CDC42 GAP (**5**,**6**) the score obtained

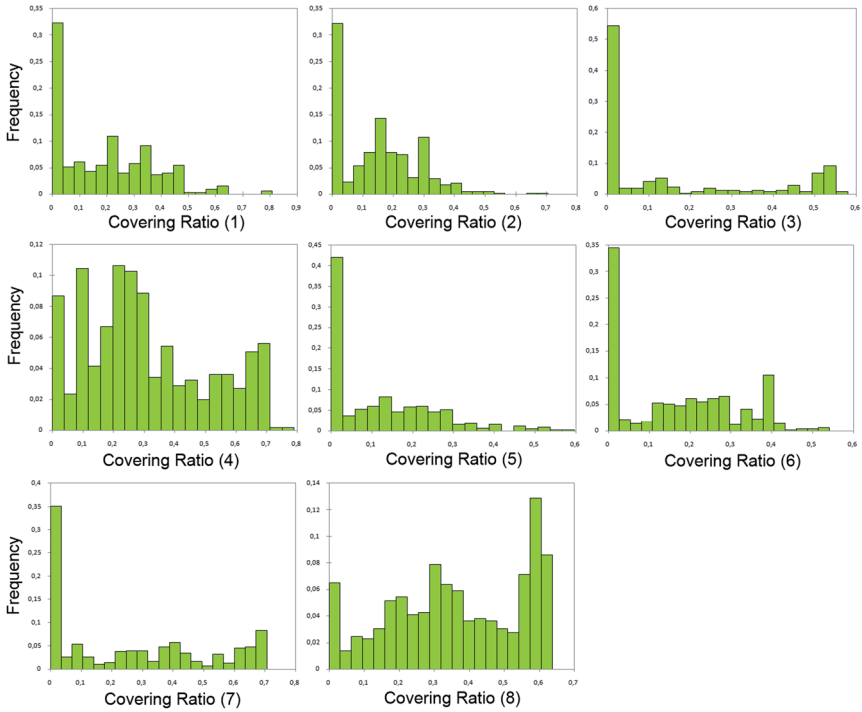

Fig. 6 Frequency of experimental interface covering ratio for the 8 proteins of the dataset (index **1** to **8**) towards the entire dataset.

with the experimental partner didn't stand out compared to the score obtained with the decoys. These observations seem consistent with the previous results, as the covering ratios show that users hardly found the correct interface for proteins **1**, **2** and **5**, leading to average docking scores for couples (**1**,**2**) and (**5**,**6**).

Discussion

Users' exploration of the dataset

Different reasons could explain the differences in the surface of the protein explored by the users. Intuitively, if proteins are explored equally, larger proteins should display more atoms that are less explored relative to smaller proteins. This is clearly the case with the 4 largest proteins (**3**, **5**, **6** and **7**), which display the highest number of less explored atoms, as illustrated in Fig. 5. It should be noted that we randomized the list of the proteins presented to the users at the start of the 2 weeks-playtest and not at each login. This randomized list exhibiting the largest proteins in the middle (**1**, **2**, **4**, **5**, **3**, **6**, **7**, **8**) could have impacted the choice of the complexes to explore selected by the users. An additional reason for this variability in the protein surface exploration is that the users could have identified at the surface of a given protein a very attractive potential docking spot that they tended to use more than other points of the surface. For all proteins, the users seemed to identify attractive spots that could be mostly in the experimental interface (proteins **1**, **4**, **6**, **7** and **8**) or out of the experimental interface (proteins **2**

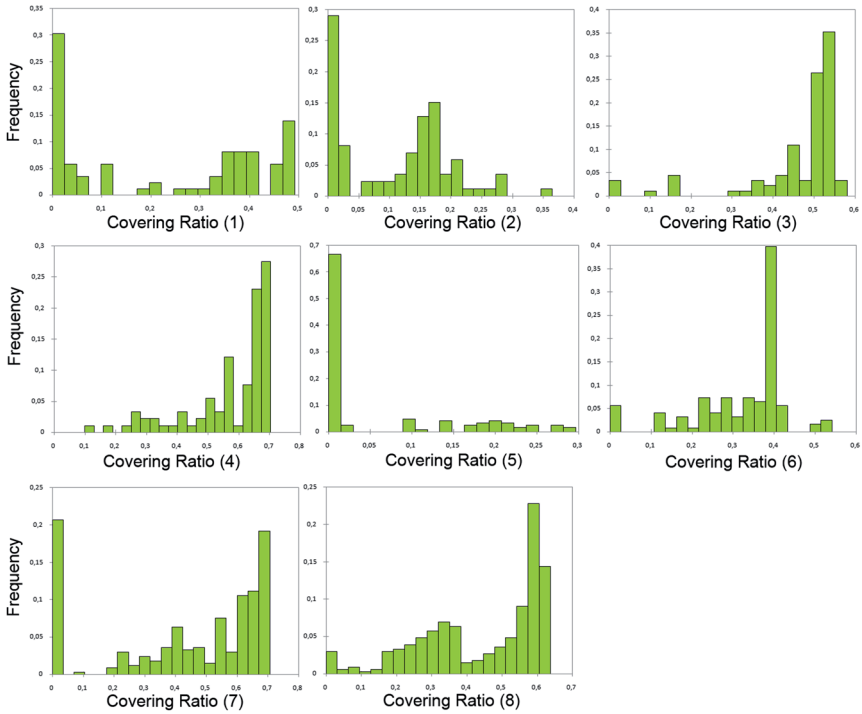

Fig. 7 Frequency of experimental interface covering ratio for the 8 proteins of the dataset (index **1** to **8**) towards their corresponding experimental partner in the dataset.

and **5**). For protein **3**, the users explored mostly in the experimental interface and its surroundings (see Fig. 4). This information from the exploration maps is confirmed by the analysis of the experimental interface covering ratios (Fig. 6 and 7), notably for proteins **4** and **8** that were particularly explored in the experimental binding interface. This points out that, even with decoy partners (Fig. 6), the experimental interface seemed particularly obvious for these proteins to the users of the alpha playtest. It is interesting to note that these proteins were the smallest of the dataset, which could result in a region of the surface that stands out in terms of geometry or charge, particularly because we used bound experimental structures to constitute the dataset. Then, when small proteins displayed particularities on the surface, as in proteins **4** and **8**, the users tended to dock them in a similar manner to the decoys or to the experimental partner since it seemed to be intuitively a good docking spot for the users. Interesting extreme profiles were also obtained for proteins **3** and **7** that were explored highly in and out of the interface. In both proteins, there is a particularly large cavity on the surface (the experimental binding site) that intuitively seemed mandatory to explore for the users, particularly for protein **3** (acetylcholinesterase).

The users successfully identified the experimental binding site for proteins **3**, **4**, **6**, **7** and **8**. For proteins **1**, **2** and **5**, they explored a lot out of the experimental interface. This could be due to less available striking particularities in terms of charge or shape that could be identified by the users as favored docking spots at the surface of these proteins.

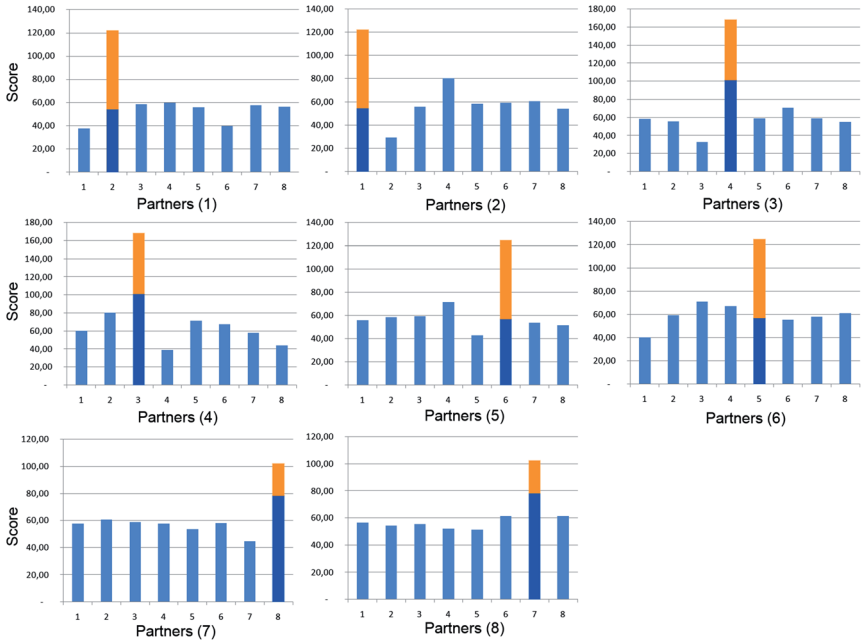

Fig. 8 Mean of the 3 best users' high scores for each protein against the whole dataset. The experimental partner is in dark blue, while the orange bar corresponds to the rescoring of the experimental geometry observed in the original PDB.

For acetylcholinesterase, fasciculin II, thermitase and eglin c (proteins **3**, **4**, **7** and **8**), the highest scores were obtained with the experimental partner. In these cases, the users seemed to be able to identify the right partner among the decoys. These complexes were the ones where the users got the closest scores to the score that could be obtained using the experimental geometry of the complex. In the other complexes, the users couldn't get higher scores with the experimental partner compared to the decoys. These results were in good correlation with the ability of the users to successfully identify the experimental interface.

The challenge of representation

Contrarily to classical protein docking approaches intended to be used by scientists who are ultimately experts in protein docking, Udock is also intended for naïve users that have not necessarily been sensitized to structural biology and protein energetics. Therefore, one of the challenges of our work was to tackle protein docking critical features in an accessible manner in order to also be performed by naïve users.

Making the representation of complex protein structures accessible to naïve users was the first challenge to overcome. Proteins are composed of several thousand atoms that render an explicit all-atom representation very confusing for non-experts. Since the problem in protein docking is to optimize the geometry of the complex by optimizing the binding energy between the two interacting partners, we chose to focus the representation of the system in Udock on these critical features, namely the protein's shape and electrostatics. Thus, we did not chose

classical displays used by structural biologists to represent protein structures like wireframe, van der Waals volumes or balls-and-sticks, but focused on a global representation of the shape by displaying the solvent excluded surface (SES) of the protein. When using a van der Waals surface representation, the users' general view of the shape can be perturbed by the numerous invaginations occurring on the protein surfaces. SES carried the advantage of hiding these non-critical details about the protein shape.

We chose to use a standard 1.4 Å probe size for the SES generation, but our approach allowed different sizes of probes to be used. Indeed, we only used the generated surface for display and early collision detection. Ultimately, protein docking was performed using an all-atom rigid-body Monte Carlo optimization procedure that does take the SES into account. In the next version of Udock, we plan to use a different mesh for early collision detection and visualization, so that we would be using much more simplified representations, while still allowing proteins to be docked together. For example, the SES of barnase is displayed in Fig. 9 with 3 different probe sizes for the SES generation (1.4 Å, 2.4 Å, 3.4 Å), leading to a less and less detailed shape.

To color the SES, we decided not to use the classical CPK coloration guide[20] but to represent a simplified smoothened electrostatic potential derived from the atomic partial charges as computed by AMBER12.[15] We first represented the atomic partial charge at the surface, which resulted in precise information that was much too detailed and complex to be used during the docking process by a naïve user. When using our smoothing algorithm, we found it much easier to understand the global electrostatic configuration of the protein using larger stains of colour. It resulted in less precise but more concise information to drive the docking process, as the user tries to match globally positive areas to globally negative areas, and use the local automated optimization procedure to do the fine tuning. Moreover, this representation of the electrostatics on the surface can be helpful to the user to remember the global shape of the protein. Large and coloured stains can be used as landmarks by the user: for instance on Fig. 1, barnase can be described as featuring a positive concave region surrounded by two positive salient shapes.

We chose to generate SES surface and electrostatic potentials on the fly, every time a user loads a pair of proteins. This choice allowed us to have software that only relied on mol2 molecular description files, which are relatively small and commonly used in computational chemistry. This choice carries two advantages. First, it becomes very easy for anyone to modify the models loaded in Udock as we used a standard mol2 format. Second, these files can be easily transferred *via* the internet from our servers allowing updates of the datasets explored by the users

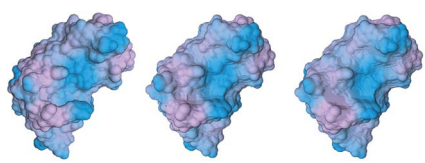

Fig. 9 Solvent excluded surface of barnase generated using probes of different size (from left to right: 1.4 Å, 2.4 Å, 3.4 Å).

without requiring large amounts of data to be downloaded every time. Yet, this choice comes with its drawbacks. Surface generation requires computational resources, and even if we optimized this step, the users reported waiting too long when loading a pair of molecules. We thus plan to optimize further our surface generation algorithms and to cache the generated models when possible to reduce the waiting time between docking runs.

The challenge of scoring on the fly during interactive docking

One of the objectives with the development of Udock was to perform interactive docking and scoring on the fly. The scoring is indeed a very difficult task to address during interactive protein docking on the fly since it takes a lot of computational resources that will also be needed by the physics and rendering engine to maintain a good fluidity in the animation of the objects and their interactions.

For instance, the calculation of the interaction score for a barnase/barstar pose takes around 50 ms, depending on the number of atoms in contact. To maximize the resource that will be used by the physics and rendering engine, we compute the interaction score on the fly every 250 ms to maintain a correct frame rate. Still, when running on computers with very limited resources, animation and manipulation of the proteins can become less fluid as the calculation of the interaction score takes too much time.

To limit the impact of this problem, we plan in the next version to let the user deactivate this feature as long as he is manipulating the molecules. Then, we would only compute the interaction score when the molecules become still, letting the user always manipulate them in a fluid fashion while still being informed on the quality of a specific pose.

Gamification of the protein docking challenge

To foster the user's motivation, we made a first step towards gamifying the protein docking process. The docking process is indeed a very interesting task to gamify: it can be viewed as a very simple pattern-matching task and as one of the most complex tasks to perform since it is not even mastered by the experts (CAPRI[1] is still considered as a very challenging experiment). The challenge was thus to make this task accessible to naïve users as a pattern-matching toy, while slowly guiding users into the realms of protein docking.

The simplification of rendering and manipulation we discussed in the previous section can be seen as the first step of the gamification process. We needed the users to be able to interact with the system very quickly, so that they could start learning by practice as soon as possible. But even if we simplified the docking process, naïve users still needed to be guided at the beginning and since we could not rely on classical documentation that would be too complex for naïve users, we created a basic tutorial. Even then, some users hardly understood some of the basic principles of the protein-docking task. For example, we did not anticipate that the representation of the charges on the surface would be counter-intuitive for naïve users since they intuitively wanted to match similar colors.

We also decided to provide users with a Monte Carlo rigid body optimization procedure in order to gamify what we felt to be the right part of the protein docking process. Indeed, we felt that users were inherently good at understanding

the global shape of proteins and could be very efficient at identifying promising binding sites. Once the potential binding site was found, we felt that a local automatic optimization procedure would be much more efficient than the user at finding the precise geometry that would optimize the interaction score. One further step could be to design a gamified task that could replace or assist the optimization performed by the Monte Carlo procedure. This would be an entirely new activity for the user that would require new tools and visualization techniques to maintain Udock as accessible and motivating to naïve users. Yet, we feel that Udock is at a sweet spot between human and machines, using in an optimal manner the power of the brain and the power of the computer. As a consequence, perturbing this balance could be both destructive to the user's motivation and to the quality of the collected data.

To foster the motivation of the user, we needed to provide clear goals. Hopefully, the interaction score could be used as a game score, giving the user a clear goal of beating his own score. Also, since the users' behavior is logged on a web server, we could compute a global ranking among the users, and thus create a competitive element. When a user starts to dock two proteins together, the best interaction scores obtained by the other users with this particular complex are displayed on the interface. This provides users with a clear set of goals: beat his own high score and every high score displayed in the score bar.

We also tried to add feedback to inform the user about the quality of his performance. Feedback is fundamental to gamification as it guides the user and helps sustain his motivation.[21] Every time a user beats his own high score, we immediately inform and reward him with graphical and sound feedback. If the user beats one of the other users' score, he is also informed by the corresponding feedback, and given the name of the beaten user.

Finally, we created a compelling and immersive atmosphere by developing a dedicated soundtrack and using advice on color schemes from a graphical designer.

Still, the gamification process in Udock is far from being complete. In this first version, we could identify two major issues in the gameplay. First, even if we tried to make the docking process accessible to naïve users, the tutorial seemed clearly too short for the users to fully understand the challenge of protein docking. In the next version, we will need to make it much richer. Second, users that understood the docking process did not play for a long time because the game did not foster long term motivation. Beating scores can be seen as fun, but it's clearly not enough since we need to keep the users learning. We will need to provide tools that will assist them to perform even better docking, and provide them with new opportunities to learn and try different docking strategies. For instance, we could provide different displays of the proteins (not only the shape), give more information about the individual proteins to dock (protein sequence, for example) and thus give them the opportunity to perform protein docking, not only with regard to the shape and electrostatics, but by using other information.

Conclusion

In summary, we developed an interactive docking system, Udock, that allows a quick and easy-to-handle human driven exploration of protein–protein interfaces. We simplified the representation of protein structures and gamified the protein

docking task to make it accessible to even naïve users. To validate our approach, we designed an open alpha cross-docking playtest during two weeks on 4 experimentally resolved protein complexes, leading to 36 possible complexes to explore.

Despite the small amount of time allowed for the Udock open alpha playtest and the relatively small number of active users (12 that played at least 30 min), different observations could be derived. The users explored almost all of the surfaces of the proteins that were available in the dataset but favored certain regions that seemed more attractive as potential docking spots. These favored regions were inside or close to the experimental binding interface and, for 5 out of the 8 proteins, the most explored regions covered the majority of the binding interface. For half of the proteins in the dataset (acetylcholinesterase, fasciculin II, thermitase and eglin c), the highest scores were obtained with the experimental partner.

This work could give preliminary insight on (1) the power of crowd sourcing on challenging tasks, *i.e* protein–protein docking; (2) protein–protein interfaces and interactions, as the users could identify experimental interfaces and sometimes the partners in interaction, and (3) a better way to craft games for science.

Acknowledgements

The authors are grateful to Dr Patrick Fuchs, Dr Anne Lopes, Clément Pillias and Hélène Manche for fruitful discussions. We also would like to thank all Udock alpha-testers for their time and investment.

References

1 J. Janin, K. Henrick, J. Moult, L. T. Eyck, M. J. Sternberg, S. Vajda, I. Vakser and S. J. Wodak, Critical Assessment of, P. I., CAPRI: a Critical Assessment of PRedicted Interactions, *Proteins: Struct., Funct., Genet.*, 2003, **52**, 2–9.

2 D. W. Ritchie and G. J. Kemp, Protein docking using spherical polar Fourier correlations, *Proteins: Struct., Funct., Genet.*, 2000, **39**, 178–94.

3 R. Chen, L. Li and Z. Weng, ZDOCK: an initial-stage protein-docking algorithm, *Proteins: Struct., Funct., Genet.*, 2003, **52**, 80–7.

4 D. Kozakov, R. Brenke, S. R. Comeau and S. Vajda, PIPER: an FFT-based protein docking program with pairwise potentials, *Proteins: Struct., Funct., Bioinf.*, 2006, **65**, 392–406.

5 I. A. Vakser, Low-resolution docking: prediction of complexes for underdetermined structures, *Biopolymers*, 1996, **39**, 455–64.

6 J. J. Gray, S. E. Moughon, T. Kortemme, O. Schueler-Furman, K. M. Misura, A. V. Morozov and D. Baker, Protein-protein docking predictions for the CAPRI experiment, *Proteins: Struct., Funct., Genet.*, 2003, **52**, 118–22.

7 R. Abagyan, M. Totrov and D. Kusnetsov, ICM - a new method for protein modelling and design. Applications to docking and structure prediction from the distorted native conformation, *J. Comput. Chem.*, 1994, **15**, 488–506.

8 C. Dominguez, R. Boelens and A. M. Bonvin, HADDOCK: a protein-protein docking approach based on biochemical or biophysical information, *J. Am. Chem. Soc.*, 2003, **125**, 1731–7.

9 A. M. Wollacott and K. M. Merz, Jr, Haptic applications for molecular structure manipulation, *J. Mol. Graphics Modell.*, 2007, **25**, 801–5.
10 O. Delalande, N. Ferey, G. Grasseau and M. Baaden, Complex molecular assemblies at hand *via* interactive simulations, *J. Comput. Chem.*, 2009, **30**, 2375–87.
11 T. C. Lu, J. Ding and S. N. Crivelli, DockingShop, a Tool for interactive protein docking, *IEEE Visualization 2005*, 2005.
12 A. Kawrykow, G. Roumanis, A. Kam, D. Kwak, C. Leung, C. Wu, E. Zarour, p. Phylo, L. Sarmenta, M. Blanchette and J. Waldispuhl, Phylo: a citizen science approach for improving multiple sequence alignment, *PLoS One*, 2012, **7**, e31362.
13 S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic and F. Players, Predicting protein structures with a multiplayer online game, *Nature*, 2010, **466**, 756–60.
14 C. Crawford, *The Art of Computer Game Design*, 1982, ISSN: B0052QA5WU.
15 D. A. Case; T. A. Darden; T. E. Cheatham; C. L. Simmerling; J. Wang; R. E. Duke; R. Luo; R. C. Walker; W. Zhang; K. M. Merz; B. Roberts; S. Hayik; A. Roitberg; G. Seabra; J. Swails; A. W. Goetz; I. Kolossvai; K. F. Wong; F. Paesani; J. Vanicek; R. M. Wolf; J. Liu; X. Wu; S. R. Brozell; T. Steinbrecher; H. Gohlke; Q. Cai; X. Ye; J. Wang; M.-J. Hsieh; G. Cui; D. R. Roe; D. H. Mathews; M. G. Seetin; R. Salomon-Ferrer; C. Sagui; V. Babin; T. Luchko; S. Gusarov; A. Kovalenko; P. A. Kollman, *AMBER 12*. University of California, San Francisco, 2012.
16 E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng and T. E. Ferrin, UCSF Chimera–a visualization system for exploratory research and analysis, *J. Comput. Chem.*, 2004, **25**, 1605–12.
17 W. E. Lorensen and H. E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *ACM SIGGRAPH Comput. Graphics*, 1987, **21**, 163–169.
18 Bullet physics library, real-time physics simulation. http://bulletphysics.org.
19 S. Sacquin-Mora, A. Carbone and R. Lavery, Identification of protein interaction partners and protein-protein interaction sites, *J. Mol. Biol.*, 2008, **382**, 1276–89.
20 R. Corey and L. Pauling, Molecular Models of Amino Acids, Peptides, and Proteins, *Rev. Sci. Instrum.*, 1953, **24**, 621–627.
21 K. Salen; E. Zimmerman, *Rules of Play, Game Design Fundamentals*, MIT Press 2003, ISBN-13: 978–0262240451.