2

Advances in Human-Protein Interaction - *Interactive and Immersive Molecular Simulations*

Nicolas Férey, et al.*
CNRS - Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur - Université Paris XI Bâtiment 508, 512 et 502 bis, 91403 Orsay Cedex France

1. Introduction

Molecular simulations allow researchers to obtain complementary data with respect to experimental studies and to overcome some of their limitations. Current experimental techniques do not allow to observe the full dynamics of a protein at atomic detail. In return, experiments provide the structures, i.e. the spatial atomic positions, for numerous biomolecular systems, which are often used as starting point for simulation studies. In order to predict, to explain and to understand experimental results, researchers have developed a variety of biomolecular representations and algorithms. They allow to simulate the dynamic behavior of macromolecules at different scales, ranging from detailed models using quantum mechanics or classical molecular mechanics to more approximate representations. These simulations are often controlled *a priori* by complex and empirical settings. Most researchers visualise the result of their simulation once the computation is finished. Such post-simulation analysis often makes use of specific molecular user interfaces, by reading and visualising the molecular 3D configuration at each step of the simulation. This approach makes it difficult to interact with a simulation in progress. When a problem occurs, or when the researcher does not achieve to observe the predicted behavior, the simulation must be restarted with other settings or constraints. This can result in the waste of an important number of compute cycles, as some simulations last for a long time: several days to weeks may be required to reproduce a short timespan, a few nanoseconds, of molecular reality. Moreover, several biomolecular processes, like folding or large conformational changes of proteins, occur on even longer timescales that are inaccessible to current simulation techniques. It can thus be necessary to impose empirical constraints in order to accelerate a simulation and to reproduce

---
*Alex Tek, Benoist Laurent, Marc Piuzzi, Zhihan Lu and Marc Baaden (CNRS - Laboratoire de Biochimie Théorique, Institut de Biologie Physico-Chimique, 13, rue Pierre et Marie Curie, 75005 Paris, France)

Olivier Delalande (CNRS - Interactions Cellulaires et Moléculaires - Université de Rennes 1, Avenue du Professeur Léon Bernard, 35065 Rennes cedex, France)

Matthieu Chavent (Structural Bioinformatics and Computational Biochemistry Unit, Dept. of Biochemistry, University of Oxford, United Kingdom)

Christine Martin, Lorenzo Piccinali, Brian Katz and Patrick Bourdot (CNRS - Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur - Université Paris XI Bâtiment 508, 512 et 502 bis, 91403 Orsay Cedex, France)

Ludovic Autin(Molecular Graphics Laboratory Department of Molecular Biology, MB-5 - The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, CA 92037-1000, USA)

an experimental result in MD. These constraints have to be defined *a priori*, rendering it difficult to explore all possibilities in order to examine various biological hypotheses.

A new approach allowing to address these problems has emerged recently: Interactive Molecular Simulation (IMS). IMS consists in visualising and interacting with a simulation in progress, and provides the user with control over simulation settings in interactive time. With the recent advances in human computer interaction and the impressive increase of available computing power, the IMS approach allows a user to interact in 3D space in real time with a molecular simulation in progress. This approach provides quality control features by visualizing results of a simulation in progress and supplies interactive features, such as feeling forces involved in the simulation as well as triggering specific events by applying custom forces during the simulation in progress. These advances led to a new generation of scientific tools to better understand life science phenomena, which place the human expertise at the centre of the analysis process, complementarily to automatic computational methods.

The IMS approach emerged from the breakthrough initiated by the *Sculpt* precursor program proposed by Surles et al. (1994). Since then, the interactive molecular simulations field has been developing continuously. Initial interactive experiments using molecular mechanics techniques gave quickly rise to "guided" dynamics simulations [Wu & Wang (2002)] or *Steered Molecular Dynamics* (SMD) [Isralewitz et al. (2001)] [Leech et al. (1996)]. The interest for these methods increased with the enhancement of simulation accuracy and thanks to the exciting new possibilities for dynamic structural exploration of very large and complex biological systems. In the Interactive Molecular Dynamics (IMD) approach, steering forces are applied interactively with a chosen amplitude, direction and application point. This enables the user to explore the simulation system while receiving instant feedback information from real-time visualisation or haptic devices [Leech et al. (1997)]. Schulten's group has carried out several applications of IMS simulations to macromolecular structures [Grayson et al. (n.d.)] [Stone et al. (2001)]. This effort lead to the design of two efficient software tools facilitating the process of setting up an IMS : *NAMD* and *VMD* [Phillips et al. (2005)] [Nelson et al. (1995))]. The underlying exchange protocol is also supported by ProtoMol [Matthey et al. (2004)], LAMMPS [Plimpton (1995)], HOOMD-blue [Anderson et al. (2008)] and any software using the MDDriver library [Delalande et al. (2009)]. Similar projects proposing an interactive display for molecular simulations exist, such as the *Java3D* interface proposed in Knoll & Mirzaei (2003) and Vormoor (2001), or the *Protein Interactive Theater* [Prins et al. (1999)].

With fast generalization of new computer hardware devices and increasing accessibility to powerful computational infrastructures, IMS showes a fast and promising evolution, even for very large molecular systems (over 100.000 atoms). Such applications are now in the reach of state-of-art desktop computing. This evolution was possible given the strong increase in raw computing power leading to faster and bigger processing units (multi-processors, multi-core architectures). Currently ongoing technological developments such as GPU computing and the spread of parallelized entertainment devices (PS3, Cell) with specific graphic and processing capabilities open exciting new opportunities for interactive calculations. These approaches could provide even more processing power for highly parallelizable computational problems, for instance by differentiating the parallelisation of molecular calculations and graphical display functionalities. Given these developments, the range of accessible computational methods and representations is bound to grow. It may soon be possible to extend the IMS approach to *ab initio* or QM/MM calculations. Indeed, the precision achieved in the description of a system can be improved by switching to a more

accurate physical model and/or by improving the representation of the molecular context simulated. Thus, multi-scale simulations [Baaden & Lavery (2007)] would indeed benefit from an interactive approach leading to important advantages with respect to the study of complex biological systems. However, the raw increase in computer speed alone is not sufficient to grant a successful future evolution of the IMS approach. In addition, it is necessary to develop adapted software solutions, which are generally more efficient [Grayson et al. (n.d.)], as it is commonly admitted in the numeric simulation field. Finally, the most recent and famous work illustrating the revolution of this approach is the "Fold It" serious game, which allows a user to interactively propose a protein folding solution [Cooper et al. (2010)].

We will describe in this chapter the recent advances relating to these IMS approaches previously described. As IMS implies to efficiently combine simulation and interaction features, we will explain how we designed specific simulation, visualisation, and interaction techniques to solve the real time constraint, to study complex biomolecular systems, and to address a larger simulation timescale. Then we will discuss software architectures to efficiently put the different building blocks together. Finally, we will explain how we apply IMS to different fields of research including various topics such as protein-protein docking in a virtual reality and multimodal context, an ion substitution study using an haptic device, and a study about the opening and closure of the Guanylate Kinase enzyme.

2. Multiscale and multiphysics protein simulation models

In structural biology, recent advances in experimental techniques allow us to solve larger and larger protein 3D structures. However, even if structure is known to be strongly linked to biological function, static states often lack in providing dynamical informations that are crucial for the understanding of the subtle mechanisms occuring at the molecular level. Thus, molecular simulations are nowadays used to complete experimental biostructural studies, especially to better understand the dynamic behaviour and the fundamental mechanisms involved in a protein complex. In spite of the increasing computational resources, classical simulation tools are not well adapted to quickly obtain insight into the global biomechanical properties, because of the limited timescale covered by all-atom or coarse-grained simulations. For these reasons, it is necessary to develop new modeling approaches at a larger scale, complementary to all-atom and coarse-grained models, especially designed to interactively study protein complex formation and biomechanical properties of large biomolecular structures. We present in this part unconventional approaches that could address these requirements. The first one, based on a rigid body model of a protein, was especially designed to study protein-protein interactions for an interactive rigid docking application. The second one, based on a spring netwok model, takes into account protein flexibility in order to study biomechanical behavior of large protein structures in interactive time.

2.1 A rigid body simulation model to interactively study protein-protein interactions

At a larger scale, it is sometimes not necessary to model and simulate the flexibility of a protein, but sufficient to consider the protein as a rigid body. Using a simple but accurate model at the macroscopic scale allows us to overcome the main constraint to provide an interactive time biophysical simulation as required for IMS: taking into account the user interaction during a simulation in progress. To present our rigid body simulation model dedicated to IMS, especially interactive rigid docking, we have to focus on the main phenomena that are involved in the protein interactions.

2.1.1 Geometry and surface

Proteins can be viewed as both the building blocks and the workforce of cells. They are synthesized based on portions of DNA (Deoxyribonucleic Acid) called coding sequences or genes. Genes are transcribed in the form of mRNA (Messenger RiboNucleic Acid), which is then translated by ribosomes in the form of a protein, following a specific coding scheme (figure 1A). Each triplet of mRNA bases corresponds to one AA (Amino Acid) or residue, of which there are twenty basic types. The various physicochemical properties of AAs give rise to interactions at the atomic level, inducing protein folding which contributes in turn to protein stability (figure 1B). These properties also play a crucial part in protein-protein interactions.
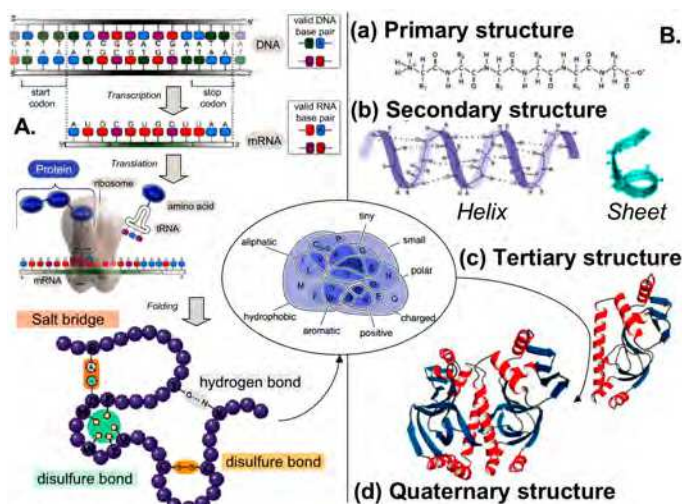

Fig. 1. (**A**) - Overall view of protein synthesis: transcription of DNA to messenger RNA (mRNA) and translation of mRNA to amino acid sequences chosen from 20 possible varieties, here shown according to their physicochemical properties (using a Venn diagram). (**B**) - Based on the chemical nature of component amino acids, resulting interactions cause the protein to fold to a favorable arrangement in space. This 3D shape can be described according to four levels: the (a) primary, (b) secondary, (c) tertiary and (d) quaternary structure.

Proteins, therefore, can be seen as long chains composed of successive amino acids folded in space, which are the product of the expression of an organism's genetic makeup. But in order to execute their functions within cells, proteins must undergo folding and take a specific 3D form. This form may be characterized following four levels of structure (see figure 1B). The order in which residues are linearly arranged, *i.e.* their sequence, constitutes the protein's primary structure. (see figure 1B-a). Some of the structure's segments organize themselves into sequences of specific substructures called secondary structures (see figure 1B-b). These structures, stabilized by hydrogen bonds, can be divided into two groups: regular secondary structures, called alpha helices and beta sheets, which are linked together by irregular structures called loops. The arrangement of these secondary structures thus

constitutes the 3D, or tertiary structure of the protein (see figure 1B-c), which determines protein function within the cell.

Once folded, proteins carry out various functions within the cell, such as transporting molecules to and from various components of the organism (*e.g.* hemoglobin, chaperone proteins), inter- and intracellular signaling and communications (*e.g.* hormones, neurotransmitters, ions), immune defense functions (immunoglobulins, adhesion molecules) or cellular metabolism (chlorophyll, apoptosis proteins, transcription factors, ATP synthesis). These cellular functions are closely linked to the protein's tertiary structure, but also to its interactions with other proteins.

In short, better understanding of protein-protein interactions is a major stake for biomedical research. Indeed, designing new drugs increasingly involves targeting specific protein-protein interactions [Villoutreix et al. (2008)], or alternatively, involves synthesizing recombinant proteins meant to emulate interactions with the original native protein [Pipe (2008)]. It becomes more and more necessary, therefore, to identify the 3D structure of protein complexes. Two main experimental methods are currently used to determine the 3D structure of a protein complex. These are X-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy. All known publicly available protein structures are currently housed on the website of the Protein Data Bank (PDB) [Berman et al. (2000)]. This database contains now several hundred thousand protein structures for many organisms. However, this number remains small in comparison to estimates of the number of existing proteins in the natural world. This is because experimental determination of protein structure is often difficult, and in some cases impossible. Indeed, solving a problem of this kind involves mass production and purification of the protein, and in the case of crystallography, production of diffractive crystals. In determining the structure of a protein complex, difficulties in production and purification are all the more critical, because partner proteins must be produced at the same time for complexes to form. Furthermore, the time necessary for crystallization may be incompatible with the lifespan of some complexes. For all these reasons, many scientists have attempted to predict the structure of such complexes using computational tools through methods and algorithms for molecular docking.

Current techniques for the experimental study of the 3D structure of protein complexes (crystallography, NMR, electron cryomicroscopy, SAXS, etc.) have several limitations (in terms of size and type of proteins) and are costly in terms of time and money. For that reason, computer-based (*in silico*) docking methods have been developed in the past, to deduce the functional 3D structure of a complex based on single molecules, which turns out to be considerably easier and cheaper than experimental *in vitro* methods. Current approaches are strictly computational and results are evaluated using visualization tools. These approaches can be divided into 4-5 successive stages (figure 2): (1) choice of the representation mode for proteins (atomic view, pseudo-atoms, grid, etc.); (2) conformational exploration (taking into account position, orientation, and shape of the ligand); (3) minimization of the function used to evaluate binding energy (i.e. *score*) for conformations derived from the exploration; (4) grouping by similarity and classification through evaluation or fine-tuning of the *scores*, augmented with a manual stage of visualization when the score alone doesn't allow native conformations (i.e. the ones present in nature) to be discriminated from other generated conformations; (5) an optional stage for fine-tuning selected complexes, through energy minimization or molecular dynamics.
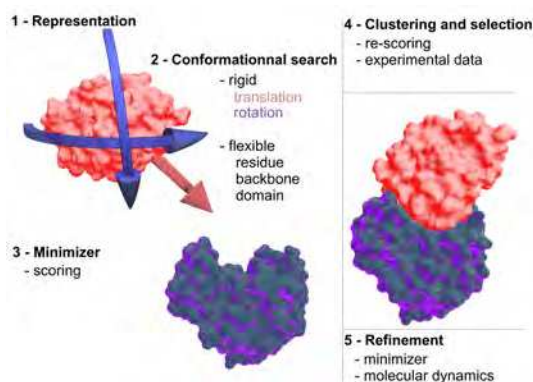
Fig. 2. The 5 stages of the docking task

A large number of fully automatic computational docking algorithms depend on an comprehensive approach of conformational exploration, the main problem being combinatorial explosion of the number of possible solutions. These approaches can be sorted into three categories: those based on systematic sampling, on molecular dynamics techniques and on classification interaction modes between proteins. An ideal function would yield, for a given mode of interaction, the binding energy of two proteins involved in a complex (see section 2.1.3). Such functions aim to reproduce experimental values of free binding energy, and through minimization, to reach the overall minimum energy in the set of all possible protein-protein complexes.

Consequently, in real life cases, automatic docking algorithms, such as *ClusPro* [Comeau et al. (2004)] or *Hex* [Ritchie (2003)], must manage two difficulties in order to reach a relevant result. The first is to process a space of potential solutions which increases in size along with the number of degrees of freedom in describing protein position and conformation, thus running the risk of not beeing processed in an acceptable amount of time. The second problem is that search algorithms produce local minima, and cannot easily find the global minimum that is associated to the native form of the complex [Wang et al. (2003)].

To finalize a docking simulation, experts rely upon a manual stage of visualization to analyse the generated complexes. This task consists in a detailed analysis of residues and atoms involved in the interface each complex, through the identification of hydrogen bonds, salt bridges, and especially the presence of hotspots, *i.e.* amino acids at the interface, known from experimental studies to be an essential part of this interface. However, it can be difficult to manipulate two 3D structures at the same time to observe the interface with traditional interaction tools, since one protein usually hides the other. Therefore docking assisted by user interaction is a useful alternative to improve the work of experts in this field. Such techniques might allow a more intuitive interaction with 3D protein structures.

Finally, two approaches are used to "thin the herd" of selected complexes. One consists in minimizing the rigid bodies and lateral chains of amino acids present at the interface. This approach is implemented in several applications such as *ICM-DISCO* [Fernandez-Recio et al. (2003)], *MMTK* [Hinsen (2000)], *FireDock* [Andrusier et al. (2007)], *PELE* [Borrelli et al. (2005)], *ATTRACT* [Zacharias (2005)], etc. The other approach involves studying the dynamic behavior of the selected complex. The software program *Gromacs* [Hess et al. (2008)], for

example, allows evaluation of atomic positions over time based on their physicochemical properties. This approach allows first to evaluate the complex stability, as well as possible conformational changes induced by the interaction, *e.g.* loop deformation. We should add, however, that this approach remains very costly in terms of processing time, compared to minimizers which allow users to process a given configuration very quickly.

As the automatic docking software programs previously presented did not respond to the interactive time constraint, we developed a new simulation tool dedicated to interactive protein-protein rigid docking. Our protein docking method is essentially based on two sets of criteria: geometric/topological criteria, and biophysical criteria.

2.1.2 Interactive time evaluation of geometry and surface complementarity

One of the earliest criteria identified in protein-protein interaction is surface topology of the proteins involved. In most known structures of 3D complexes, partners exhibit good surface complementarity. Studies have also shown that the surface of the protein-protein interface generally covers between 1000 and 2500 square Angstroms. This criteria allowed the development of first-generation docking software, based solely on shape recognition [Connolly (1983)] (*i.e.* complementarity of molecular surfaces). This approach is well adapted to "hard" rigid protein docking. We used these geometric/topological criteria in our multimodal immersive environment in two ways:

Surface collision. For each protein, a surface mesh is computed using the *MSMS* software before interactive docking occurs [Sanner et al. (1996)]. The resolution of this mesh can be adjusted using parameters. Collision detection during interaction then uses the *RAPID* library [Gottschalk et al. (1996)], which allows real-time computation of a list of colliding triangles among the two protein surface meshes during docking. This set of triangles can be used to generate feedback based on triangle normals and on the intersection volume of the two protein surfaces.

Atomic surface complementarity. Atomic surface complementarity is estimated essentially as a calculation of the variance of the inter-atomic distances on the two protein surfaces. We use this overall atomic surface complementarity score in audio or visual feedback.

2.1.3 Interactive time computation of physicochemical properties and energies

However, geometric criteria turned out to be insufficient to predict the structure of a complex. Thus, we had to rely on methods including energy criteria. Protein-protein complexes seem to follow the rule of thumb that the active configuration is the one whose level of free energy is lowest [Wang et al. (2003)]. In order to evaluate free energies between two proteins, we rely on molecular mechanics methods. For this purpose, atoms are viewed as spheres, and interactions between atoms can be computed using the van der Waals and electrostatic potentials. The free energy for protein-protein interaction can then be approximated by the sum of these potentials, which is known as the *score*. In the context of real-time immersive docking, the choice of equations and methods to evaluate the energy of a complex and hence its score is a crucial issue [Wang et al. (2003)].

Van der Waals interactions. Van der Waals interactions are an empirical approximation of atomic interactions. The van der Waals force, obtained by constructing a gradient of the potential field, is defined by the Lennard-Jones potential equation (equation 1). In this

equation, r_{ij} is the distance between two atoms i and j, σ the interatomic distance for which the potential becomes zero, and ϵ the depth of the potential well. ϵ and σ are determined empirically and depend on what pair of atoms is considered. The van der Waals potential includes an attractive component when atoms are bound, and a repulsive component when atoms are too close to each other. It prevents two proteins from penetrating into each other during interactive docking, through calculation of interatomic forces at the protein-protein interface.

$$U_{vw}(r) = 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}] \tag{1}$$

These forces apply only to very short distances and mostly concern surface atoms. As computing distances between all pairs of atoms has a quadratic complexity, we apply specific filtering rules to keep only surface atoms and opposite atoms from each protein (see figure 3). The resultant translational and rotational components of van der Waals's forces on each atom are calculated and applied to the barycenters of the proteins.
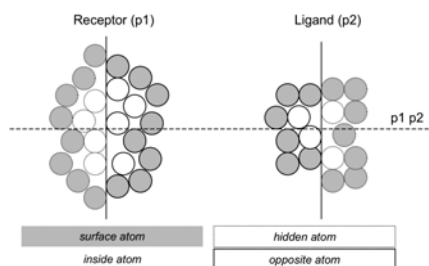

Fig. 3. Dynamic and static atom filtering for optimized computing of van der Waals interactions

Electrostatic interactions Unlike van der Waals interactions, electrostatic interactions even operate when "long" distances (about 10 Angstrom) separate groups of electrically charged atoms. Indeed some amino acids or atoms may present a positive or negative electric charge, which gives rise to electrostatic phenomena allowing formation of a protein-protein complex. Two approaches have been implemented to compute electrostatic phenomena.

We consider the interaction between two point charges in vacuum, and we use Coulomb's law (equation 2) with r_{ij} being the distance between the barycenters of charges q_i and q_j of the atoms considered, and ϵ_0 is the constant of the permittivity of vacuum. This potential can be translated to a force (F_{el}) usable for haptic interaction for example. This first approach involves calculating the forces to apply to each electrically charged particle considering only pairs of charged particles. This computation has quadratic complexity, because all distances between atoms must be computed. But it remains relevant in the case of medium-sized proteins, since the number of charged particles in a protein is limited in several models.

$$U_{el}(r_{ij}) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \tag{2}$$

In the second approach (see figure 4, designed for a more efficient optimised calculation, the overall field of the electrostatic potential of the target protein (receptor) is computed beforehand using the *APBS* software [Baker et al. (2001)]. It allows to generate a 3D electrostatic potential grid, which can be used as a 3D texture. The gradient of the electrostatic

potential allows computation of force field vectors for each point of the grid. Atoms from the ligand protein are then "immersed" in this 3D force field surrounding the receptor. This method allows us to compute electrostatic forces for each atom in linear time, depending only on the number of charged atoms in the ligand. In both cases, we are able to obtain overall electrostatic energy and electrostatic forces on each atom.
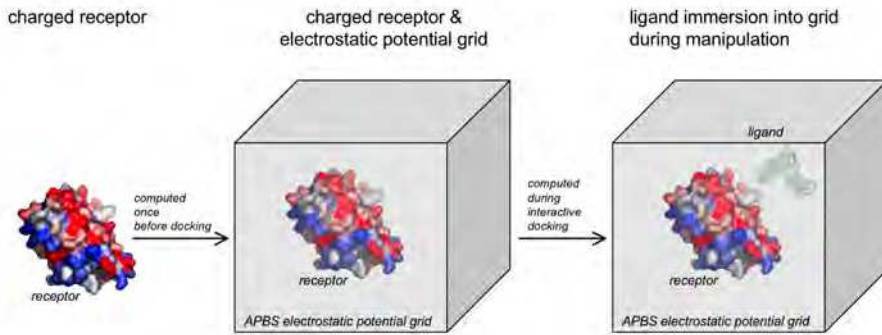

Fig. 4. Ligand immersion in the electrostatic potential grid of the receptor

2.1.4 Other criteria

In order to reach a finer description of protein-protein interactions, other criteria, based on energy, can be taken into account. To geometric/topological and biophysical criteria, one can add other criteria of utmost importance to protein-protein interactions, such as hydrogen bonds or the hydrophobic effects.

Hydrogen bonds. Hydrogen bonds (*e.g.* figure 1 in the bottom left corner) may strongly contribute to the favorable interactions of the complex binding energy. On average, there are 5-6 hydrogen bonds per protein-protein interface. In our application, when several atoms (nitrogen and oxygen) on the surface of each protein are close enough, closer than a distance of 3 Angstroms, and when their chemical environment is adequate, hydrogen bonds are created between these atoms. We use the same methods as described above for van der Waals interactions to filter surface atoms in order to decrease the complexity of calculating distances between atoms.

Hotspots at the interface. The number of "hotspots" at the complex interface refers to the list of amino acids present within the current interface region and previously identified using experimental methods as being important actors to stabilize the protein-protein complex.

2.1.5 Conclusion

This simulation model, based on rigid body docking, including optimisation to efficiently compute geometrical as well as biophysical properties, allows us to present these properties in real time during the interactive building of a protein complex performed by the user. Designing real time simulations is a first step of the IMS approach, providing a user with interactive control on the simulated object in real time.

2.2 A multiphysics and multiscale approach based on elastic networks

Simulation models dedicated to IMS must also deal with the intrinsic flexibility of proteins, and especially take into account local moves as well as large conformational changes. The representation used for our interactive simulation approach is quite simple, yet innovative, and has proven efficient on large biomolecular structures. Our method is based on a spring network simulation, inspired by the success of the Normal Mode Analysis method (NMA), known to accurately reproduce the elastic behavior [Cui & Bahar (2006)]. Moreover NMA is not sensitive to the scale of representation used for modeling. Hence this method can be applied to all-atom, coarsed-grain and residue/CA representations. We augmented this spring network model with non-bonded interactions and we propose to surround the charged spring network by an electrostatic field, allowing us to study conformational changes guided by electrostatic constraints. We called our implementation of this method *BioSpring* and describe it in this section. *BioSpring* allows us to simulate large structures in real time, fulfilling the most important constraint to provide IMS features.

2.2.1 BioSpring : an enhanced interactive spring network model

The first step of our approach is to build the spring network according to the 3D structure of the biomolecular system [Berman et al. (2000)]. At this stage, the user needs to choose a scale, targeting for example an all atom (AA), a coarse-grained (CG), or an alpha-carbon representation (CA). In this context, individual atoms can be considered as separate particles (AA model), or can be grouped into a single pseudo-particle, according to rules defined by the user (e.g. at the CG or CA level). In this way, we can adapt our approach to most commonly-used modeling approaches in theorical biochemistry.

The second step, is to connect the particles by springs obtained in the previous step. For this purpose, we define a distance cut-off, and we add a spring between two particles if the distance between them is less than the cut-off distance. This cut-off will depend on the scale and the representation mode (all-atom, coarse-grained, alpha-carbon, ...). For example, a cut-off between 7 and 15 angstrom is classically used for the CA representation [Cui & Bahar (2006)]. This process can be computationnally very time-consuming, especially on large structures. For each particle, we need to test if any other particles are closer than the cut-off distance. This approach has a quadratic complexity according to the number of particles. In order to deal with large structures and to decrease the complexity of the previous approach, we use a classical technique based on a regular 3D grid to partition a three dimensional space into cubes, also named "voxels". The grid covers the entire space occupied by the particles. The size of each voxel is the cut-off distance. According to its coordinates in space, each particle is projected into its voxel on the grid. In order to determine for a given particle p, all particles near p within the cut-off distance, we have to test the particles in the same and in the direct neighbouring voxels. This method has linear complexity according to the number of particles, allowing us to address very large structures. We will see in the next part of this paper how we can use the same grid to efficiently compute non-bonded interactions between particles.

After building the initial spring network, interactive manipulation of this molecular structure is provided using a classical newtonian particle-based simulation, taking into account spring forces (see equation 3) between particles and external forces $\vec{F}_{control}(p)$ (see 5 equation) provided by the user on a particles p through a specific graphical user interface. In the

equation 3, $k_{stiffness}$ is a global stiffness for all the springs and the force between two particles p and p' linked by a spring depends on the distance between these particles. If this distance between p and p' is equal to the equilibrium spring length $e_{pp'}$, which is the distance between these particles in the initial structure used to compute the network, this force is null by definition. In the other cases, when the distance $d_{pp'}$ between p and p' changes because of external forces, the generated spring forces tend to bring the structure back to its equilibrium conformation at $e_{pp'}$. Damping forces are used to stabilize the system (see equation 4). This is necessary because the user injects energy into the system by adding external arbitrary forces. It should be noted that some experimental and theorical studies provide estimates for $k_{stiffness}$, which allows us to work with magnitudes of forces in the simulation that are relevant from a biophysical point of view.

$$\vec{F}_{spring}(p) = \sum_{p' \in Springs(p)} k_{stiffness}\vec{u}_{pp'}(d_{pp'} - e_{pp'}) \tag{3}$$

$$\vec{F}_{damping}(p) = -k_{damping}\vec{V}(p) \tag{4}$$

$$\vec{F}(p) = \vec{F}_{spring}(p) + \vec{F}_{damping}(p) + \vec{F}_{control}(p) \tag{5}$$

At each time step of the simulation, to compute new positions and velocities according to the spring and external forces applied on the particles, we use a velocity verlet integrator described in equation 6.

$$\vec{P}(t + \Delta t) = \vec{P}(t) + \vec{P}(t)\Delta t + \tfrac{1}{2}\vec{A}(t)\Delta t^2$$

$$\vec{V}(t + \Delta t) = \vec{V}(t) + \frac{\vec{A}(t) + \vec{A}(t + \Delta t)}{2}\Delta t \tag{6}$$

Finally, the graphical user interface must provide interactive simulation features, allowing a user to visualise the spring network simulation in progress and to interactively apply external forces $\vec{F}_{control}(p)$ on particles. Combining these interactive simulation features and our interactive spring network simulation approach, a user can manipulate some parts of a large biomolecular system, and interactively observe the effects of this manipulation highlighting biomechanical properties such as rigid vs. flexible areas or allosteric effects.

However, even if the spring network model embeds an approximation of bonded and non-bonded interactions at the local scale, this model is not able to deal with long range and steric interactions during a simulation, because the spring 'particles' (atom, coarse grain, or residue-level) are considered as points. For example, domain interpenetration is allowed in the default spring network model. This is not a problem when the goal is to highlight local flexibility or rigid areas, but it is a critical issue for our objective of interactive modeling of large biomolecular systems. Similarly, it is also necessary to take into account electrostatic interactions during interactive modeling. For these reasons, in addition to spring forces, we introduce classical non-bonded forces to take into account steric and electrostatic interactions between particles in our model.

In order to meet the specific needs of the user, BioSpring provides a variety of terms to represent steric interactions. The most simple term is the linear steric model (see equation 7) which can be used to avoid atom or pseudo-atom collisions and take into account the 3D shape of the biomolecular model. This avoids domain interpenetration during the interactive manipulation, without taking into account complex realistic steric energy considerations and

in particular attractive terms. More classical models such as Lennard-Jones (see equation 8) are also avalaible, in order to take into account both attractive and repulsive interactions between atoms or pseudo-atoms, and to compute a more relevant steric energy during an interactive simulation. Atom radius r, epsilon ϵ and sigma σ parameters can be set up using configuration files, allowing us to use many of the currently available forcefields. However, this method has a complexity in $O(n^2)$ which is quadratic according to the number n of particles, because for each particle we need to compute the distance with respect to all other particles in the simulation. To address larger biomolecular systems, we necessarily have to decrease the complexity. We can remark that beyond a certain distance, several pairwise interactions become null or negligible. This is especially the case for linear or Lennard-Jones steric interactions. In this case, according to this distance cutoff, we can use the same optimisation techniques as in section 2.2.1, projecting particles into a 3D grid to accelerate the distance computation between particles at each time step of the simulation, by reducing quadratic complexity to linear complexity according to the number of particles in the simulation. The complexity is in $D.O(n)$ which is linear according to the number n of particles and the mean number D of particles in a voxel, which can be considered as a constant because it is related to the mean density of particles at a molecular scale.

We also use another way to optimize our simulation method as in many cases some part of the biomolecular complex can be considered as a rigid component. In this case, we can consider that these particles are static, *i.e.* have a constant position in space, because they belong to a rigid component. Hence it is useless to compute their interactions and to apply a positional integration on these static particles. We only have to take into account interactions between the dynamic particles belonging to the flexible part, and the interactions originating from the static particles and acting on the dynamic ones. This is a simple way to decrease complexity.

The following equations 7 to 9 describe the last two optimisations. *Dynamic* is the dynamic particle set, which contains all the particles belonging to the flexible part. The complexity is in $D(|Dynamic|)$ which is linear according to the number $|Dynamic|$ of particles in the flexible part.

$$s_{pp'} = (r_p + r_{p'}) - d_{pp'}$$

$$\vec{F}_{linearsteric}(p \in Dynamic) = \begin{cases} \vec{0} & \text{if} \quad s_{pp'} \leq 0 \qquad (7) \\ \\ \sum_{p' \in Neighbors(p)} -k_{steric}\vec{u}_{pp'}s_{pp'} & \text{else} \end{cases}$$

$$\vec{F}_{lennardjonessteric}(p \in Dynamic) = \sum_{p' \in Neighbors(P)} \vec{u}_{pp'}4\epsilon_{pp'}\left[\left(\frac{\sigma_{pp'}}{9d_{pp'}}\right)^9 + \left(\frac{\sigma_{pp'}}{7d_{pp'}}\right)^7\right] \quad (8)$$

$$\vec{F}_{coulomb}(p \in Dynamic) = \sum_{p' \in Neighbors(p)} -\vec{u}_{pp'}\frac{q_pq_{p'}}{4\pi\epsilon_0 d_{pp'}^2} \quad (9)$$

We can highlight another important fact: the previous approach is well-adapted to efficiently compute steric interactions by defining a distance cut-off. For long range interactions such as electrostatic ones, we must be extremely careful with this cut-off. It is preferable to avoid the use of cut-offs to stay biophysically relevant, but in this case, we fall down to quadratic complexity. We thus propose an efficient alternative to take into account long range

electrostatic interactions, considering that some parts of a biomolecular complex are rigid. A charge distribution can be translated into an electrostatic potential map, using the APBS tools [Baker et al. (2001)] for example. Charged particles belonging to the rigid components of our complex can be considered as a charge distribution, and are used as an input for APBS. The results of APBS can be interpreted as a 3D grid, and each voxel $V_{i,j,k}$ of this grid contains an electrostatic potential $E_{i-1,j,k}$. For a dynamic particle belonging to $V_{i,j,k}$, using this potential, we can compute an electrostatic force $\vec{F}_{map}$ using the charge of the particle V_p, by spatial derivation of this electrostatic potential (see equation 10). The potential forces $\vec{F}_{map}$ act on the flexible part and originate from the electrostatic potential map. They are defined by computing of the electrostatic potential gradient using the finite central difference method. In equation 10, we consider particle p belonging to the voxel $V_{i,j,k}$ of the electrostatic potential grid, and $E_{i,j,k}$ the value of the potential in this voxel. We define the gradient as the mean of the difference between the $E_{i,j,k}$ potential and the potentials of the six adjacent voxels, two for each axis. This method of computing the gradient reduces the bias related to the discretization of the grid. As regular grids are usually provided by tools such as APBS, Δ_x, Δ_y and Δ_z is the size of the voxel.

$$\vec{F}_{map}(p \in V_{i,j,k}) \simeq \begin{bmatrix} \frac{E_{i+1,j,k} - E_{i-1,j,k}}{2\Delta_x} \\ \frac{E_{i,j+1,k} - E_{i,j-1,k}}{2\Delta_y} \\ \frac{E_{i,j,k+1} - E_{i,j,k-1}}{2\Delta_z} \end{bmatrix} \tag{10}$$

To summarize, this last optimization technique is particularly well-adapted to study the behaviour of flexible biomolecules interacting with a large rigid biomolecular complex. Flexible parts are immersed into a grid and guided by a potential field induced by the rigid component, computed before the simulation. We have combined Eulerian (particle-based) and Lagrangian (grid-based) representations for molecular simulations, inspired by Joe Stam's works on Computational Fluid Dynamics [Stam (1999)]. This approach is also called semi-lagrangian or semi-eulerian method.

During the simulation, forces are computed and applied on the dynamic particle set ($P_d ynamic$). We explicitly consider potential, van der Waals, Coulomb and external forces. Finally, these new forces are summed with an external force $\vec{F}_{control}(p)$ provided by the user through the graphical interface during the simulation.

$$\vec{F}(p) = \vec{F}_{spring}(p) + \vec{F}_{damping}(p) + \vec{F}_{map}(p) + \vec{F}_{steric}(p) + \vec{F}_{coulomb}(p) + \vec{F}_{control}(p) \tag{11}$$

2.2.2 Conclusion

BioSpring allows a user to quickly study the biomechanical properties, by interactively highlighting rigidity, flexibility, and allosteric effects, in order to provide new hypotheses about a biomolecular system. Moreover, our approach is also designed to help the user in the complex task of modeling large biomolecular complexes before using more classical (and more time-consuming) simulation tools.

3. Multimodal interaction models

In order to interact with biomolecular complexes during a particle-based interactive simulation such as *Gromacs* [Hess et al. (2008)], *NAMD* [Phillips et al. (2005)] or *BioSpring* (see section 2.2.1) in progress, it is common to use a mouse for adding force constraints on

particles, providing two degrees of freedom (2DoF), e.g. the x- and y-axes, for the interaction. Using a 3DoF device such as a 3D mouse or a 3D haptic device is even better adapted to this task, in particular for selecting and moving particles in 3D space. Such a device with three instead of two degrees of freedom is more intuitive and efficient for interacting with a complex three-dimensional object, especially when stereoscopic features are used to improve the spatial perception. Furthermore, the immediate force feedback using a haptic device when a particle is actually picked significantly improves the user experience and greatly helps to immerse the user in the molecular scene. If visual feedback is essential especially during the selection and picking task of a particle, the user often asks for additional explanations before getting started. With force feedback, this barrier is lifted, as the interactive simulation becomes more intuitive and is comparable to intuitive dextrous manipulations such as those carried out in daily life. Hardware requirements are modest. In our experience, this approach is viable using a small and affordable haptic device, providing 3D positions and handling 3D directional force feedback. Such an entry-level solution designed for a desktop use is targeted at a large user community and is very easy to set up.

3.1 Pick and pull particle interaction models

The haptic device is used in order to control the direction of the forces applied to selected particles and to adjust the amplitude of these forces. This interaction method contains two stages. The first stage comprises the selection of a single probe particle or a set of particles that we will name $P_{selection}$, using a 3D tool attached to a haptic device and its buttons. In a second stage, the model described by equation 12 is used in order to compute the forces $F_{control}$ applied to the selected particles and sent to the BioSpring simulation as control force (see section 2.2.1). $F_{control}$ is proportional to the distance between the geometric centre of the particle set and the tracker position $P(tool)$. For computing force feedback, the main idea of this approach is to link the selected atoms and the 3D haptic tool with a spring. Instead of providing direct haptic rendering of forces computed in the simulation, the force feedback $F_{feedback}$ only depends on the spring length according to equation 13, which in turn is influenced by the way the simulation reacts to the applied force.

$$\vec{F}_{control}(p \in Selection) = -k_{control}[\vec{P}(tool) - \frac{1}{|Selection|} \sum_{p' \in Selection} \vec{P}(p')] \qquad (12)$$

$$\vec{F}_{feedback}(tool) = -k_{feedback}[\vec{P}(tool) - \frac{1}{|Selection|} \sum_{p' \in Selection} \vec{P}(p')] \qquad (13)$$

The resulting forces are rendered by haptic feedback if a haptic device is used, and by visual feedback such as the blue arrows shown in the Molecular User Interface (MUI), top left part of Figure 5. These forces are simultaneously sent to the interactive simulation. It will take these forces applied by the user on the selected atoms set into account as a control force as described in section 2.2.1.

We emphasize that the haptic loop computation frequency must be between at least 300 to 1000 Hz in order to provide a haptic rendering of good quality. A strong point of the approach described above is that a low physical simulation framerate does not cause instabilities and does not affect the quality of the haptic feedback. With this decoupled spring model, force feedback can be computed at a very high frequency required by the haptic device.
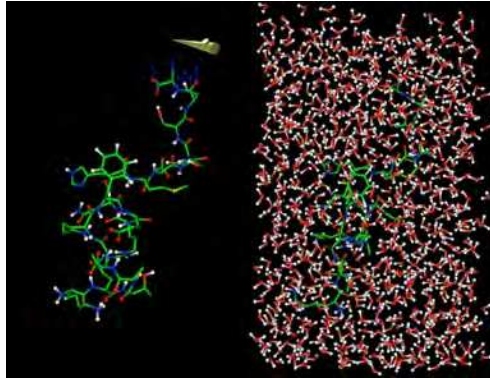
Fig. 5. Dynamic haptic control of a simple polypeptide with (right) or without (left) solvent - "ball and stick" representation

3.2 Interaction models for manipulating proteins as rigid body

The interaction method is more complicated when we want to provide controls and feedback during a rigid body based simulation, comparing to pick and pull a particle set as described in the last section. In order to manipulate both individual proteins and attempt to interactively study interactions between two proteins, the user may rely on various devices and interaction paradigms. A first paradigm associates the position and orientation of the protein with a 6DoF (6 degrees of freedom, 3 for translation, 3 for rotation) devices, such as a 3D mouse or a haptic device. Commonly used in the Virtual Reality domain, haptic devices are specifically used for manipulation and assembly tasks. Collision feedback rendered by 6 Degrees of Freedom (6DoF) haptic devices helps users to assemble 3D objects (see figure 6).
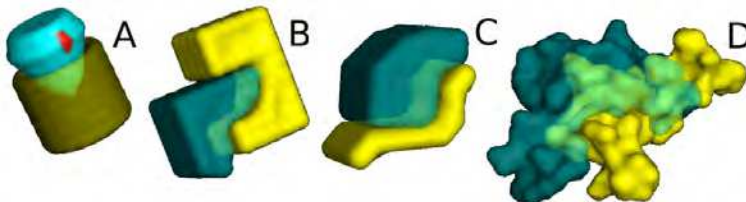

Fig. 6. Different kinds of assembly

3.2.1 Related works dealing with device workspace limitations

All devices have a limited workspace, a limited precision, and limited rotational movements. In order to overcome these limitations, a basic manipulation control is the clutching/unclutching interaction technique, which is however time consuming and does not allow a user to focus on his task. When the user is physically stopped in his movement, by reaching either a boundary of the device or an uncomfortable wrist position, he can press the clutching button to find a better position without moving the virtual object. When releasing, the object is re-attached with the same position and orientation. In this technique, the position and orientation of the 3D virtual object is an isomophic mapping of the position and orientation of the device.

Other solutions avoiding clutching/declutching, such as the *Bubble* technique, [Dominjon et al. (2005)] propose, to perceive (via haptic and visual feedback) the hardware limitations of the device, and provide a rate control based on an isomorphic mapping when the device is far from its workspace boundaries, and on a non-isomorphic mapping near the boundaries, also proposed LaViola & Katzourin (2007).

3.2.2 Related works that deal with high precision assembly

Morover protein-protein docking tasks require high precision. Haptic-guidance based approaches are often used in order to help the user reach a precise and predefined assembly goal. However, there are a few haptic interaction techniques designed to facilitate microassembly tasks for which haptic guidance is unsuitable, such as protein docking. The objective is to find an optimal but precise 3D configuration by interactive exploration.

3.2.3 A haptic interaction paradigm for rigid body based biophysical simulation

We propose an innovative technique to both overcome the physical limitations of the device and to reach the high accuracy required by micromanipulation tasks without a predefined goal, as it is the case during interactive docking simulations. This approach is based on a non-isomorphic mapping around a neutral referential retrieved by an elastic haptic feedback, in addition to external haptic feedback computed by the biophysical rigid body simulation.

In contrast to the method based on the haptic workspace boundaries, our approach is based on a *neutral referential*. Our solution is an implementation of a rate control technique with a 6DoF feedback device, based on a neutral referential, inspired by Bourdot & Touraine (2002).

Our contribution is to use the elastic force feedback to help the user return to the position/orientation of the neutral referential. First, we define a neutral referential with an origin corresponding to the most convenient position/orientation for the user holding the device. For each movement, we calculate a feedback force and torque to bring the user back to this neutral orientation/position (see figure 7 A). In order to compensate the inherent imprecisions of the device, we define a "dead zone" near the neutral referential, in which no movement occurs. The device is then physically restrained inside a comfortable workspace, while the virtual objects have an infinite motion space.

The rate control is based on the difference between the position/orientation of neutral referential initially chosen by the user and the position/orientation of the device during manipulation. The interpolation of movements is obtained by a downscale factor for translation and by a quaternion interpolation for rotation. The level of interpolation varies according to the distance of the two objects to be assembled. Concerning translational motion, the interpolation is done by rescaling the distance vector representing the position of the device from the origin of the neutral referential. In the following equation, $\vec{S}$ is the interpolated translation, $\vec{p}$ the current device's vector position and i the scaling factor. Concerning rotation, at any time of the manipulation, the rotational motion of the device controls the angular velocity of the object. The orientations of the device q_d and the object q_o are represented by quaternions. The rotational motion q_s of the object is then given by the multiplication of the two quaternions (see figure 7 B). The *SLERP* interpolation [Shoemake (1985)] is traditionally used to calculate intermediate frames between two quaternions (start and end orientations) in order to produce smooth rotation. Here, we use the *SLERP* interpolation to calculate a range of

quaternion orientations, between the current object's orientation and the one it would adopt after the motion. Then, an orientation at the time t can be picked according to the desired level of attenuation. In the following equation, q_s is the quaternion representing the rotational motion applied to the object without interpolation, q_d is the quaternion of the device and q_o the quaternion of the object. q_a is the softened speed using the *SLERP* function applied between q_o and q_s at the time t of the interpolation. Here, t and i are functions of the distance between the manipulated object and the area of assembly. The attenuation increases when this distance decreases.

$$\vec{S} = \vec{p} \cdot i \tag{14}$$

$$q_s = q_d \cdot q_o \tag{15}$$
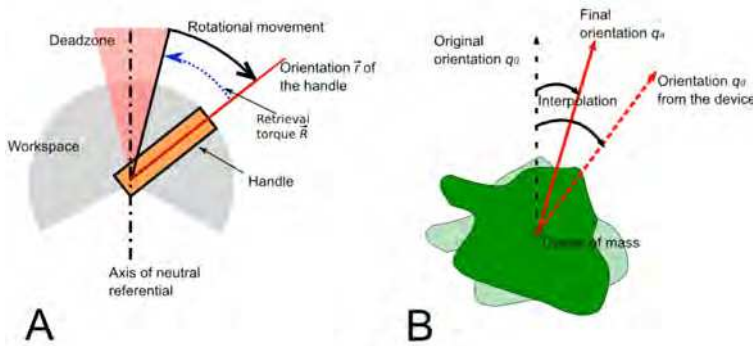
$$q_a = \text{slerp}(q_o, q_s, t) \tag{16}$$

Fig. 7. **Results of a device rotation.** A, the device is rotated from the axis of the neutral referential to the orientation $\vec{r}$, taking into account the deadzone. The retrieval torque $\vec{R}$ is thus equal to $-\vec{r}$. B, the motion of the device is mapped on the object, q_o representing the neutral referential. The final orientation q_a is obtained by applying the *SLERP* interpolation

3.2.4 External feedback

The interaction between the manipulated protein and the other one during the docking task produces biophysical interactions provided by the rigid body based biophysical simulation. These forces are not directly applied to the 3D object in the scene, but are rendered by a haptic force-feedback summed with the elastic feedback used to retrieve the neutral referential.

3.2.5 Conclusion

One of the challenges lies in the protein interaction paradigm simulated by rigid body biophysical simulation. We developed a new method to provide a fine control of the protein with a 6DoF force-feedback device, providing simultaneously biophysical feedback coming from rigid body based simulation. According to the results of an ergonomic study, our technique provides at least the same precision (RMSD) and performance (task time) as direct manipulation with clutching/declutching and successfully overcomes the physical limitations of the device. Moreover subjective results show that users feel more comfortable with our method which avoids the clutching mechanism. We suspect that these results come from the fact that the user is more focused on the assembly task, instead of spending time in

clutching/declutching. Further evaluation must be lead in this way. Participants found our technique less disturbing than clutching, appreciating the fact that there is no button to press to manipulate the object. Furthermore, their arm was never in an uncomfortable posture. They furthermore liked the adaptive interpolation. The slowness of the interaction when the two objects are very close was judged pertinent in order to accurately assemble the objects. Another interesting observation is that the most negative comments were not about the manipulation technique itself, but concerned difficulties with the 3D visual perception of a complex protein surface (see figure 6 D). Our approach could thus be an alternative to classical ones and provide at least the same efficiency. We are working on improving the precision of our approach by dynamically tuning the scaling factor used to control rotational and translational velocity. This could be done using the minimal distance between the two objects during the assembly. Finally, we highlight the fact that our approach addresses most problems of the physical limitations of haptic devices (workspace size, precision, mechanical constraints), avoids the use of a clutching/declutching mechanism, is well-adpated to both manipulation and navigation, and could be applied to other 6DoF devices, and does not require complementary visual feedback.

4. Multimodal rendering models

Given the large quantity of biophysical or geometrical information provided by IMS and conveyed to the user in real time, it seems relevant to supplement visual feedback with audio and haptic feedback. Haptic rendering is known to improve the quality of operator interactivity in an immersive environment, as well as his perception of the objects handled or data analyzed [Seeger & Chen (1997)]. Likewise, audio renderings may improve communication of complex information [Barass & Zehner (2000)]. Furthermore, substitutions and redundancy between these channels of communication may have beneficial results on user performance, as long as the choice of modalities is relevant to the task at hand. Richard et al. (2006) and Kitagawa et al. (2005) showed that specific audio and visual renderings can effectively convey information that is presented using haptic modalities. In this part, we provide some examples of haptic audio rendering especially dedicated to study protein interactions using rigid body based biophysical simulation.

4.1 Visual rendering

To represent protein structures, the community of biologists uses standard representations that any specialist can understand 8. They range from a per-atom representation 8(A, B, C) to molecular surfaces 8(H). Some high-level metaphors with ribbons and arrows 8(E,F,G) can describe the secondary structure in a schematic way. Color schemes for atoms respect different standards to simplify the distinction between the different elements of the molecule.

IMS can act at different scales, from whole proteins to precise atomistic interactions, sometimes in the same simulation run. The visuals must then follow the user needs. Three main features have to be fulfilled : interactive frame rate, display of potentially huge molecules and coherent visual information.

For rigid-body docking, pre-computed triangulated surfaces of the proteins and secondary structure representations can be used. But if the atoms are allowed to move inside the structure, computing their surface in real-time is too time consuming in most of the cases.
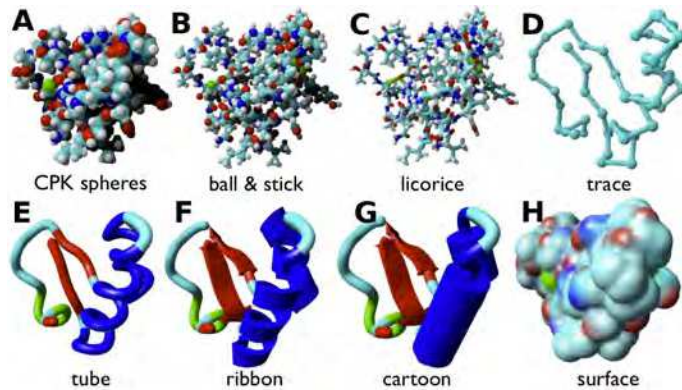
Fig. 8. Standard representation of protein structures. A-D : atomistic representation with spheres and bonds. Colors depend on the atom element. D : Backbone of the protein. E-G : secondary structure visualisation with high-level metaphor objects. Beta-sheets are in red, helices in blue and turns in green

So when it comes to more precise interaction and soft docking, spheres and bonds are more tractable.

Representation of proteins by spheres and bonds using common graphical primitives is easy to implement but generally not appropriate to reach an interactive frame rate. Each primitive is composed of many triangles, then displaying spheres or cylinders consumes a lot of computation time.

Other methods use Graphics Processing Unit (GPU) programming capabilities to draw the spheres and bonds directly on the GPU with no other information than the size and position of the particles.

Different textures and effects can be applied to emphasize interesting locations, collisions or other physical properties.

4.1.1 GPU shaders and HyperBalls

The computer visualization field evolves very quickly due to continuously renewed graphics hardware capabilities. So, the latest contributions from this domain of research has clearly helped scientists to display more and more complex systems. The latest graphics techniques can provide an improved visual perception which could drastically impact the way to visualize molecular structures [Chavent, Lévy, Krone, Bidmon, Nominé, Ertl & Baaden (2011)]. For example, using GPU shaders, i.e. code used to directly program the GPU, it is possible to accelerate and enhance the quality of well known molecular representations such as Molecular Surfaces (figure 9 A), Ball & Stick (figure 9 B), Van der Waals (figure 9 D and E) or protein Secondary Structure (figure 9 C). It is also possible to add lighting effects in real time in order to improve the perception of molecular shape or highlight molecular contours (figure 9 D and E). Furthermore, one can add effects such as blur to depict protein flexibility (figure 9 B). All these graphics techniques, available in real time, will be a great help for the users to interact in a wiser manner with their molecular structures.
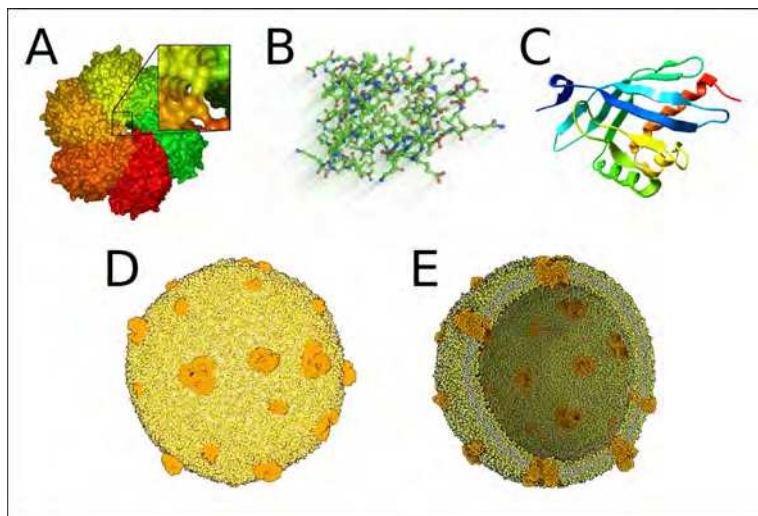
Fig. 9. New and revived molecular metaphors. (A-D) several molecular surface representations: (E) illustrate lighting effect to enhance molecular structure perception.

We have recently developed our implementation of molecular representations on the GPU [Chavent, Vanel, Tek, Lévy, Robert, Raffin & Baaden (2011)]. In this work, we introduced a visual molecular model, the HyperBalls representation, that offers a continuous representation smoothly connecting between classical representations such as licorice or ball-and-stick (figure 10). This representation takes benefit of a GPU ray-casting implementation to visualize molecular systems efficiently. The proposed implementation of the HyperBalls method is efficient for both static and dynamic visualizations of a large number of molecules and is particularly well adapted to visualize huge molecular systems. At present, without further optimization, we can smoothly and interactively render systems with more than 560,000 atoms, reaching some limits for systems comprising a few million spheres. We can expect that our implementation will benefit from the future GPU architectures, where performance increases drastically from a generation to another. This HyperBalls implementation is clearly well suited for an interactive and immersive approach due to the quality rendering and the display efficiency. Furthermore, it is possible to see in real time atomic bond evolution that can be beneficial for interactive docking (see figure 10).
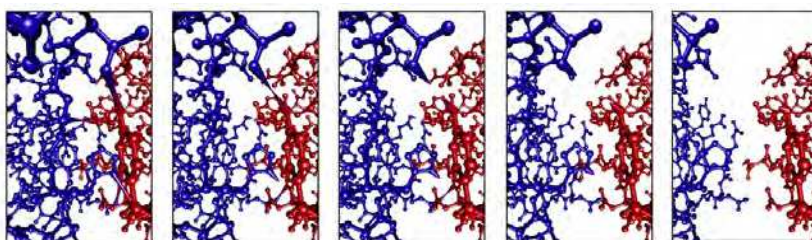

Fig. 10. HyperBalls representation to depict hydrogen bond disruption at a protein interface

4.1.2 Point-sprites

A simple way to represent molecular structure is to depict it as a collection of spheres. To represent spheres, it is possible to use only one square per sphere, always oriented perpendicular to the screen plane. Then an image (also called sprite) of a sphere is pasted on this square (figure 11). This method is usually used to depict visual effects such as flames, smoke or dust where one needs to display a big amount of animated particles. This method is really efficient and commonly implemented in 3D graphics libraries. The main drawback is that sprites are superimposed on each other, so there is no intersection between the individual spheres and it implies to sort the particles along the depth axis.
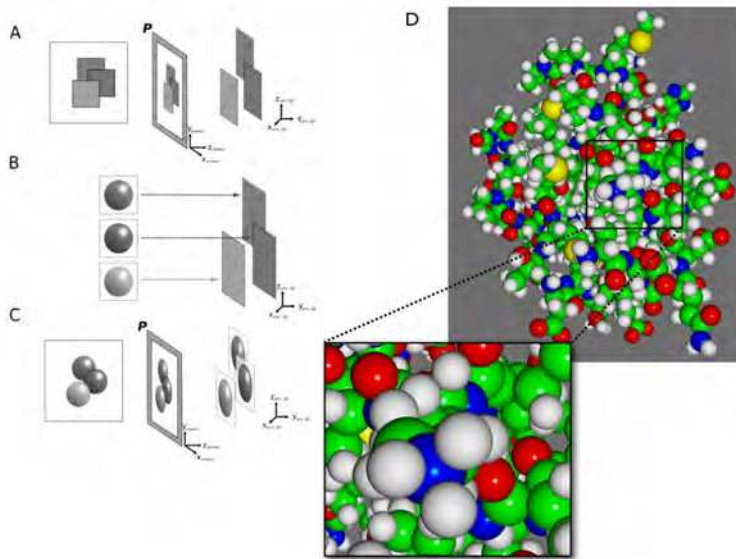

Fig. 11. Point-Sprites method to represent atoms of a protein

4.1.3 Benchmarks

These methods, as well as HyperBalls, have been implemented in an Unity3D (http://unity3d.com/) application and evaluated in terms of frame rate. The HyperBalls GPU shaders were adapted to fit the constraints of Unity3D but the performance was not as good as the initial implementation. The benchmarks show that the point-sprites method is far more efficient than the others. However, the frame rate is not constant when the camera is moving. In fact, the particles must be sorted to be displayed correctly which takes some time when there is a huge number of particles. Domain decomposition can be used to reduce this effect but then some visual glitches at the frontiers of the domains can occur.

The visual result is quite different depending on the methods. Point-sprites can be confusing as the bonds are missing and the spheres are superimposed (figure 11 D). But from a far point of view, the general form of big proteins is kept and using a good color scheme helps to distinguish the interesting areas of the molecules. The traditional primitives can be used

along with visual effects such as ambient occlusion, shading or texturing that are often pre-implemented for triangulated objects.

So what we suggest is to combine those methods in an interactive way, according to the size of the system and the user actions. Triangulated primitives are easy to implement for quick development and nice visual effects. For big systems and when the proteins are far away from the camera, particles are particularly suited. When the user zooms into a specific area, a more precise representation such as HyperBalls is adapted, especially to depict interactions.

4.1.4 Visual effects

To allow an accurate interaction with the particles, the position of each one in space must be easily discriminated by the user. The best option is to use stereoscopy but 3D display devices are not common yet. However it is possible to add some visual effects on the objects to overcome this problem. Shading, depth-cueing and ambient-occlusion are commonly used to add realistic lighting and depth perception to a 2D image (9 E). Also, texturing and contouring can help to highlight particular areas and particles and blurring effects can be used to emphasize some movements (9 B).

4.2 Haptic rendering

Currently, there are very few IMS frameworks that include large-scale haptic feedback (force/tactile feedback). This is mainly due to the complexity of computing operations of molecular simulation, which makes it difficult to comply with constraints in terms of refresh rates for real time haptic feedback (from 200 Hz to 1 kHz). Another difficulty is to render various kinds of physico-chemical interactions such as steric or electrostatic interaction. In order to obtain a consistent haptic feedback, only one type of rendering is provided to the user at a time. However one should note that at the perceptual level, steric interaction rendered using haptic feedback are similar to surface collision renderings since it prevents molecular interpenetration.

Most of haptic feedback presented in this section are computed using the rigid body simulation model described in section 2.1. In all rendering, the haptic-device controlled protein, which we will call ligand, can be considered as a big probe against the other protein, which we will call receptor.

4.2.1 Steric and electrostatic interactions

This rendering is used to provide haptic feedback of non-bonded interaction. Haptic rendering of physicochemical interactions consists in feeding the haptic device with the resultant forces computed as described in section 2.1. Forces can be computed and rendered independently or summed up to obtain a total resultant force. Exploration of the receptor by the ligand thus aims at finding stable areas. When the two proteins are in an unstable conformation it renders an unsteady feedback, thus leading the user to drag the ligand towards the surface of the receptor to find a better position and orientation. However the complexity of the force fields induce very irregular directional forces affecting the precision of the manipulation. It appears especially with steric interactions because of the non-linearity in the *Lennard-Jones* potential used to model these forces.

4.2.2 Surface collision

Two approaches were explored to render collisions between both proteins considering their surfaces. The first consists in computing a repulsive force. The direction of this force is the opposite of the direction provided and the module is proportional to the number of colliding triangles determined by the *RAPID* computation as explained in section 2.1. This force can also be weighed by a distance or a volume of interpenetration. Therefore the feedback is more relevant, but the complexity of the computation induces lower refresh rates which could lead to lags in feedback. Rather than repulse the two molecules from each other, the second approach, also based on distance computation, aims to prevent collisions locally by modeling contacts points as springs. The method is introduced by Johnson & Willemsen (2003) and allows fast computation of local minimum distances based on the geometry of the model as well as resulting force and torque. Interestingly the spring model described can be easily adapted to model atomic clashes, such as steric ones in our case. Instead of using the complex Lennard-Jones potential to render the resulting force, interactions are modeled through this more simple spring model with realistic cutoffs (2.5 Angstroms). As the atomic distance computation is already optimized to take into account only surface and opposite atoms, the refresh rate is sufficient and allows a very precise rendering of the contacts, allowing users to feel holes and bumps at the surface. Hence computation speed and consistent feedback constraints are observed ensuring a biological relevance. Current research aims to determine how the size of the proteins affects computing time. It will also be interesting to compare this atomic clashes-based approach with the geometric one which could provide faster computation.

4.3 Audio rendering

Sonification is the use of non-speech audio to convey information. Due to the high temporal resolution and wide bandwidth, the use of auditory stimuli seems highly suitable for time-varying parameters (very high temporal definition when compared to other modalities such as video and haptics), concurrent streams (overlapping of multiple audio renderings for various parameters is possible and easily understandable if these are properly designed), and spatial information (lower definition if compared to visual stimuli, but perceptible over the 360 degree sphere, therefore allowing true 3D rendering).

A large variety of sonification techniques exist and are used in various applications [Walker & Lane (1994)]. One sonification technique is referred to as "parameter mapping" [Hermann & Ritter (1999)], and it is this technique we used to study protein interaction. Parameter mapping sonification is based on creating a link between the data to be rendered and the parameters of a synthesizer (or of any other device which generates or plays back sound). In this particular sonification typology, three elements need to be carefully considered [Walker & Lane (1994)]:

- The nature of the mapping: which data dimension (*i.e.* temperature, pressure, velocity...) is mapped onto, or represented by, each acoustic parameter (*i.e.* frequency, loudness, tempo...). As an example, for a sonification task the temperature might be linked with the frequency of a sound, therefore as the temperature increases, the frequency of the corresponding sonification increases.
- Mapping polarity: in the event of an increase in the sonified data, the sonification parameter can decrease or increase. In the case of temperature-frequency mapping, it

is common to use an increasing-TO-increasing (up-up) polarity. An alternate example could be the size of an object being mapped to frequency: the polarity would likely be increasing-TO-decreasing such that large objects are linked to low sounds and vice versa.

- Mapping scale: in response to a specific increase of the data to be sonified, how much should the sonification parameter increase or decrease. One must take into account the possible range of the data, and the percentage of the usable audible range which is to be exploited. Human hearing is more sensitive to small frequency changes at low frequencies, rather than higher, following an exponential scale. In the case of temperature-frequency mapping the temperature could be exponentially linked to the frequency.

In our application, sound spatialization is used in two different ways: firstly, for local parameters the sonification is spatialised in the specific position where the parameter is calculated, in accordance with visual or haptic rendering, to provide additional information in the protein coordinate system (*i.e.* if the task is to sonify the collision between two different atoms on both proteins, sonification is spatialised at the position of the collision). Then, multiple concurrent sonifications can be spatially distributed in order to give a better intelligibility of the sonifications themselves (*i.e.* stream segregation, selective attention in auditory perception, cocktail party effect studied by Moore (2003)). In 2007, a set up a test for the validation of different sonification methods for object manipulation. Within this test, the subject was asked to change the orientation of a simplified 3D chemical compound in order to be the same as that of a given reference. To do this, the subject used an orientation tracking device. Three approaches for data parameter sonification were tested to improve the speed and accuracy of this manipulation: manipulation speed, angular distance from the reference configuration, and guidance towards the reference position.

Regarding the protein-protein docking task, the following biophysical information has been selected for the sonification:

4.3.1 Molecular surface collision and complementarity

Atomic surface complementarity is estimated essentially as a calculation of the variance of the inter-atomic distances on the two proteins surfaces. This parameter is used to control the variance of a randomly applied pitch to different grains of a granular synthesis process. Granular synthesis has been applied using a spoken word as audio sample (for this particular application, the french word "complementaire" has been recorded and used), repeated cyclically within the granular engine. In this instance, the word is unintelligible if the geometrical complementarity parameter is low, becoming more intelligible as the parameter increases. The rendered audio stream is doubled and associated to each of the two proteins, in preparation for further processing.

The collision parameter represents the number of collisions computed between the two surfaces. The employed method for atomic collision sonification uses a modulation of the phase of a sinusoidal wave whose parameters (carrier and modulator) are controlled by the global number of collisions. Starting with a continuous 400 Hz sinusoidal wave modulated by a 1 Hz signal, the frequency of the modulation increases as the global collision score gets higher, and with it the number of modulating waves, going from 1 to 4, when the two proteins are completely superposed. A second method developed is based on the individual association of every collision with a broadband noise processed with subtractive synthesis (the result is similar to wind noise). The noise is specifically filtered for every collision, adding a

controlled randomization of the filtering parameters, so that every "noise generator" sounds different from the others, and spatialised according to its proper position in space. Both of these sonification methods are based on the principle that the signal produced becomes more and more annoying as the number of collisions increases, encouraging the user to change the position and distance of the proteins in order to reduce the number of collisions, and as such stopping the annoying sound. Regarding the second sonification method, sound spatialization helps the listener to localize the part of the protein surface where the collision is taking place, and to guide him/her towards an orientation of the protein for which no collisions are present.

4.3.2 Electrostatic energy

This electrostatic parameter is computed from electrostatic interaction energies between charged particles. Electrostatic energy sonification is performed through the alternation of two sounds, generated using additive synthesis, whose pitch and timbre vary as a function of the global value of this specific force (scalar value). The electrostatic force value is highly variable, and there is not a direct linear relationship between this parameter and a quality judgement of it being good or bad for the docking condition. The link between the parameter and the quality of its specific value has therefore been traced in a two dimensional Cartesian diagram, with the value of the parameter on the X axis, and the quality (being good or bad) on the Y axis. At a given electrostatic force value, the correspondent value on the Y axis has been sonified with the method previously described. For good values, the frequencies of the two sounds are coincident, and their spectra are perfectly harmonic, whilst as the value worsens, the two frequencies become more distant, and the spectra more inharmonic.

4.3.3 Steric energy

This parameter is computed from van der Waals interaction energies between particles. The sonification of the van der Waals energy is based on the principle of the beatings between two sounds frequentially close. As with the electrostatic force, for the van der Waals force value there is not a linear relationship between the parameter and a quality judgement (being good or bad). A mapping similar to the one described for the previous sonification method (electrostatic force) has been employed, with the Y axis value being sonified. Two intermittent sinusoidal pulses are played back simultaneously: if the quality value for the van der Waals force is good, then the two waves have the same frequency, whilst as it becomes worse, one of the two pulses reduces in frequency by up to 20 Hz from the other. This processing results in the creation of beatings between the two frequencies. If there are no beatings, then the score can be considered to be good. In contrast, if the beatings become more frequent (more rapid beat frequency indicates greater frequency separation between the two pulses) the score is becoming worse.

4.3.4 Hotspots at the interface

The number of "hotspots" at the complex interface refers to the list of amino acids present within the current interface region, previously identified using experimental methods as being important actors for protein-protein interaction. Finding hotspots at the protein-protein interface is an important part in judging the quality of solutions. In a second stage, the two audio streams are processed with a low-pass filter with the cutoff frequency controlled by the percentage of protein hotspots which are situated on the interface region. If none of the

hotspots are present on the interface the low-pass filter frequency is set at 200 Hz, making the sound nearly inaudible. The cutoff frequency of the filter increases with the number of hotspots present at the interface, making the sound clearer and brighter until, in the optimal position, the frequency filtering is completely deactivated. The two audio streams are rendered stereophonically, associating the left and right channels respectively to the first and second protein.

5. Coupling simulation and interaction codes

Molecular simulation engines previously described provide time-dependent atomic or particle positions, velocities and system energies according to biophysical models at different scale. These models can now compute a molecular dynamics trajectory of interesting biological systems in interactive time. This progress allows to control and visualise a molecular simulation in progress. We have developed a generic library, called *MDDriver*, in order to facilitate the implementation of such interactive simulations. It allows to easily create a network connection between a molecular user interface and a physically-based simulation. We use this library in order to study a real biomolecular system, simulated by various interaction-enabled molecular engines and models. We use a classical molecular visualisation tool and a haptic device to control the dynamic behavior of the molecule. This approach provides encouraging results for interacting with a biomolecule and understanding its dynamics. Our goal is to extend IMS approach to a broader range of simulation engines, as the use of a specific simulation sofware or model often depends on the studied biological system. We have thus developed a generic and independent library, called *MDDriver*, which allows us to easily interface molecular simulation engines with molecular visualisation tools through a network connection. As a first step, we have rendered the calculation modules easily interchangeable while keeping the existing *VMD* user interface as MUI.

5.1 *MDDriver* : a library to coupling molecular simulations codes and molecular user interfaces

In the *VMD/NAMD* architecture, the IMD network protocol [Stone et al. (2001)] was developed in order to interface the Molecular User Interface (MUI) with the MD engine. However, the use of a specific simulation engine and MUI strongly depends on the studied biological system and on user habits. Adding IMD capabilities to other simulation engines and molecular models as well as to a variety of MUIs in addition to *VMD* and *NAMD* enables a whole range of new possibilities in interactive molecular simulations. This approach allows us to address a larger user community working on molecular modeling and simulations, sometimes based on their own home-made simulation engines. Following these motivations, we developed a generic and independent library, called *MDDriver*, inspired by the *VMD/NAMD* approach.

5.1.1 Software architecture

We have thus encapsulated the IMD protocol in the *MDDriver* library, allowing a developer to easily adapt MUI code and MD code in order to extend them with IMD features. This interface provides functions for the exchange of specific data structures over a network: atom positions and system energies, computed for each simulation step by the MD engine (server part), and user-applied forces on a selected atom set.
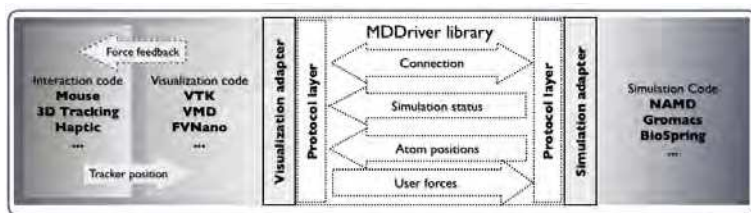
Fig. 12. *MDDriver* library for interfacing a Molecular Dynamics simulation with a Molecular User Interface (Interaction and Visualization code)

5.1.2 Molecular simulation *MDDriver* wrapper

This approach was tested, applied and improved by integrating calls to the *MDDriver* library into the *GROMACS* simulation engine [Hess et al. (2008)], thus rendering the simulation interactive via a MUI. We have used *VMD* as MUI in order to study the molecular behavior of Guanylate Kinase (GK) using an all-atom model and a coarse-grained representation [Baaden & Lavery (2007)] with *GROMACS*. Then we have tested a home-made simulation engine dedicated to molecular docking, which was also IMD-enabled.

MDDriver module offers a simple, modular and generic solution to combine any coordinates-based calculation code with various visualization programs. IMD simulation, this powerful tool for exploration of biomolecules structure in large biological system, is now accessible in a easier way to desktop or virtual reality computational environment. We insist on the fact that the *MDDriver* library was designed for easy integration into any molecular simulation engine providing time series of particle positions. Indeed there are many approaches capable of simulating the dynamic behavior of biomolecules, such as lattice simulations, elastic networks, coarse grain models or even quantum mechanical and semi-empirical methods.

5.1.3 Performances

We will only briefly comment here the desktop use performances obtained for the *MDDriver* library implementation to the *GROMACS* code. The data (coordinates, status and forces) transfer rate between calculation and visualization modules essentially depends on the size of the simulated system.Force application component alters slightly more IMD performances for small systems, depending essentially on selected/total particles ratio (increasing data exchange). In the context of large computing infrastructure deployment for *GROMACS* IMD using *MDDriver*, similar performances have been observed. This confirming robustness of the *MDDriver* library coupled to parallelized applications, performance of the display/interaction installation being the main limitation for IMD simulations of large molecular systems.

6. Applications

We propose in this last section to illustrate the previous simulation, interaction and rendering concepts especially designed for IMS, with several applications. In the first application, these concepts was used to designed new approach and methodology for docking. In the three next ones, these concepts was used in a research context to study some biostructural phenomena. In the two last one, we present two cutting edge scientific sofware that used and included all the innovative concepts presented in this chapter.

6.1 CoRSAIRe : a multimodal and immersive molecular docking project

6.1.1 Main focus

The main focus of the *CoRSAIRe* project [**?**]   is to design a new methodology in that field based on advanced interaction and rendering possibilities, that Virtual Reality (VR) technologies may offer. With respect to other works on docking, we are specifically studying multi-sensorimotor rendering during an interactive docking task.

Usually user participation during the computationnal docking process was very limited, since it only involved configuring docking scripts and choosing one result amongst the computer-generated solutions to the studied problem. Indeed classic approaches to docking provide large numbers of complex configuration based on 3D data describing partner proteins. These algorithms take a long time to produce results, since they test all possible geometric configurations to dock the two proteins. These configurations are then filtered according to energy and physicochemical criteria. Finally, the scientist selects, in this set of results, a smaller set of possible solutions that can be tested against each other experimentally.

Relying on user expertise before applying automatic docking algorithms interactive context allows the user to use natural abilities for the detection of surface complementarity, as well as prior implicit or literature based knowledge regarding for example the nature of the protein-protein interface, what hotspots are present, etc.

It seems thus relevant to develop complementary or alternative approaches to docking. In project *CoRSAIRe* (Combination of Sensorimotor Renderings for the Immersive Analysis of Results) our hypothesis is that using Virtual Reality (VR) technologies and related interactions, which rely on multiple sensory and motor channels, may help experts in this docking task. There are several reasons for this. Firstly, stereoscopy, especially when it is adaptative, may improve perception of 3D protein models. Furthermore, direct manipulation of several proteins at the same time, as afforded by peripherals commonly used today for such tasks (*e.g.* 3D mouse, force-feedback interfaces, etc.) may be more intuitive and efficient than traditional, desktop WIMP[1]-type interfaces. Additionally, multimodal management of sensorimotor feedbacks (based on an approach aiming to dynamically specify adaptation of visual, haptic and audio renderings to the characteristics of the information in use) is one possible answer to the problems related to the simultaneous presentation of large amounts of data. Finally, a strongly interactive approach of VR docking allows the docking expert to be placed on the forefront of the work, rather than giving an automatic algorithm a complete control over the generation of possible sets of solutions. We believe our approach, which combines benefits of multimodal interaction with the capitalization of docking experts' occupational skills (in biology, crystallography, bioinformatics) in modelling will allow improvements in the speed of predictions for the structure of protein-protein complexes, as well as in overall search efficiency and in the quality of results obtained when analyzing possible solutions.

6.1.2 Discussion and results

This project allow us to define the multimodal allocation space ("modal allocation" [André (2000)]) that refers to the specific use of one or more sensory modalities to display an information. It is preferable for users to use optimal modal allocation considering both technical constraint, task (*e.g.* characteristics of information relevant to scientists) and

---

[1] Acronym of Window, Icon, Menu, and Pointing device.

	Visual	Auditive	Haptic
Surface representation	ok		
Surface collisions	ok	ok	ok
Surface complementarity	ok	ok	ok
Electrostatic interactions			ok
Electrostatic energy	ok	ok	ok
Steric interactions			ok
Steric energy	ok	ok	ok
Hydrophobic patchs	ok		
Hotspots at the interface	ok	ok	

Table 1. Restricting the modal allocation space

operator-related constraints (*e.g.* characteristics of perception, of expertise, etc.). The results of the project allow us to formulate the following principles for the design of a multimodal application for molecular docking, summed up in table 1.
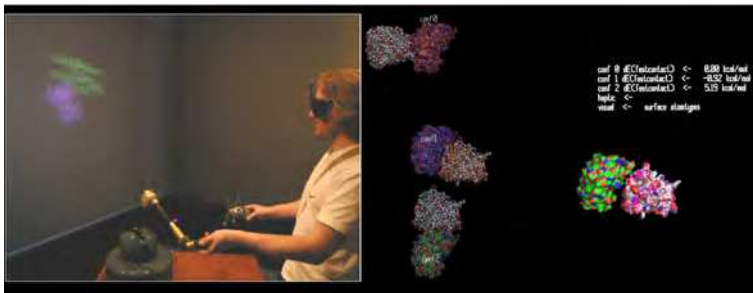
Fig. 13. A user immersed in the docking application (left). On the right, a sample screen capture following selection of three conformations by the user.

The interactive process dedicated to protein protein docking designed into the CoRSAIre project, allows significant reduction of the number of configurations to be tested by algorithms used afterwards, and we maximize the use of the user expertise. Our approach could also be reused in the design of future docking software, integrating factors such as protein flexibility, based on the premise that many docking problems involve flexible partners. Furthermore, this work should also focus on defining future situations of use of such tools. Indeed, our interactions with future users identified several possible avenues for the use of docking tools, *e.g.* teaching, scientific discovery, collaborative work, etc.

6.2 Interactively locating ion binding sites by steering particles into electrostatic potential maps

Interactively locating ion binding sites by steering particles into electrostatic potential maps Metal ions drive important parts of biology, yet it remains experimentally challenging to locate their binding sites into biomolecules (protein, DNA). With the MyPal method (Molecular scrutinY of PotentiALs), implemented in the *BioSpring* program, we use interactive steering of charged ions in an electrostatic potential map in order to identify their potential binding sites [Delalande et al. (2010)]. We use this method in order to facilitate the discovery of new relevant

ion binding sites by successfully retrieving the location of cation binding sites in DNase I enzyme and assessing their selectivity, combining atomic and coarse-grained resolutions.
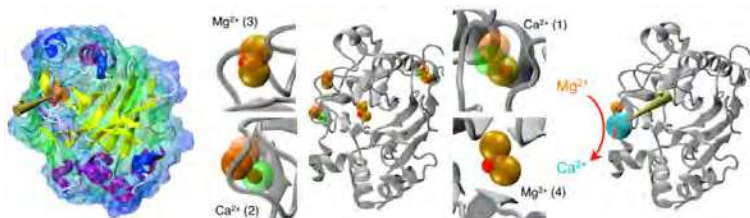

Fig. 14. Visual summary of the interactive experiments. On the left, interactive potential exploration of the DNase I enzyme using the MyPal method. On the middle, results of experiments for detecting a priori unknown ion binding sites. The reference position of each binding pocket is shown as a red sphere and MyPal predictions for a potential map with (orange) or without (green) ionic strength are displayed by transparent spheres. On the right, an ion substitution experiment ("molecular-billiard") at site 2 is depicted. Such an experiment probes the selectivity of a given ionic pocket for different ions

We interactively scanned the electrostatic potential of DNase I by using Na+, Ca2+ and Mg2+ as ionic probes. For the binding sites detection, Mg2+ cation was chosen as its double charge facilitates long-range electrostatic steering towards the binding pockets and its small size increases the accuracy for sensing the rough and detailed molecular surface at atomic resolution. Taking into account ionic strength for calculating the electrostatic potential leads to more accurate maps. However, without considering ionic strength we achieved comparable predictions and more easily detect binding sites thanks long-range driving forces (Figure 16). All four ion binding sites identified were retrieved by the MyPal approach. To assess the selectivity of identified binding sites, we start with different ions (Na+, Ca2+, Mg2+ and Cl-) at a given site and tried to interactively substitute this initial ion by another. Figure 16 and Table 2 illustrate and summarize the results for these ion substitution "molecular-billiard" simulations. As might be expected, chloride as an anion cannot be stabilized within any of the four cation binding pockets, nor can it displace a bound cation. Sites that are magnesium selective are well characterized by our approach. Less efficient substitution experiments may be related to the simplicity of our model in which selectivity depends on the shape of the pocket itself and the pathway for accessing it. Generally speaking, buried and narrow sites are unreachable for large ions, whereas sites localized at the enzyme surface are readily subject to ion exchange. In the latter case, haptic feedback helps the user to distinguish between favourable and unfavourable substitutions.

The current implementation of MyPal/BioSpring was not designed in order to provide precise quantitative binding affinity estimates, but to be capable of distinguishing in real time between non-existing, weak and strong ion binding sites and assess the relative selectivity of significantly different ionic probes. Despite the approximations made in the choice of the model representation it should remain possible to quantify the strength of binding by calculating the work required by the user to extract an ion from its binding site.

6.3 Interactive study of Guanylate Kynase opening and closure

In this study, we have worked on an intensive studied biomolecular system, the Guanylate Kinase (GK) enzyme. Structures for this molecule are provided by experimental methods

Probe (Site)	Ca^{2+} (1)	Ca^{2+} (2)	Mg^{2+} or Ca^{2+}(3)	Mg^{2+} (4)
Mg^{2+}/Ca^{2+}	$Ca \mapsto Mg$	$Ca \mapsto Mg$	$Mg \dashv Ca$	$Mg \dashv Ca$
	$Mg \dashv Ca$	$Mg \mapsto Ca$	$Ca \mapsto Mg$	-
Na^+	$Ca \dashv Na$	$Ca \mapsto Na$	$Mg \dashv Na$	$Mg \dashv Na$
	$Na \mapsto Ca$	$Na \mapsto Ca$	$Na \mapsto Mg$	-
Cl^-	$Ca \dashv Cl$	$Ca \dashv Cl$	$Mg \dashv Cl$	$Mg \dashv Cl$
	-	-	-	-

Table 2. **Ion substitution interactive simulation results.** The table indicates whether exchange from X to Y is possible ($\mapsto$) or impossible ($\dashv$). For instance, $Ca \dashv Cl$ means that Ca^{2+} cannot be displaced by Cl^-. A minus sign indicates that initial positioning of the chosen probe ion at the given binding pocket was not possible *via* our approach.

such as Nuclear Magnetic Resonance or X Ray cristallography. The molecule has a U shape with either a closed or an open conformation (see 15). The closure mechanism of GK consists in increasing the proximity of two substrate binding sites, for GMP and ATP, both essential for the enzymatic reaction. The goal of our study is to understand which parts of this system are involved in the closure mechanism. This mechanism has been investigated using our *MDDriver* framework (VMD/MDDriver/GROMACS) at two levels of detail. The first level corresponds to an all-atom model (18098 atoms), the second to a lower resolution coarse-grain model (1900 beads), and the third to a augmented spring network model. Prospective tests using coarse-grain simulations allowed for the efficient exploration of a broad range of possibilities to close the enzyme, trying to reach a closed conformation similar to the available experimental structures.
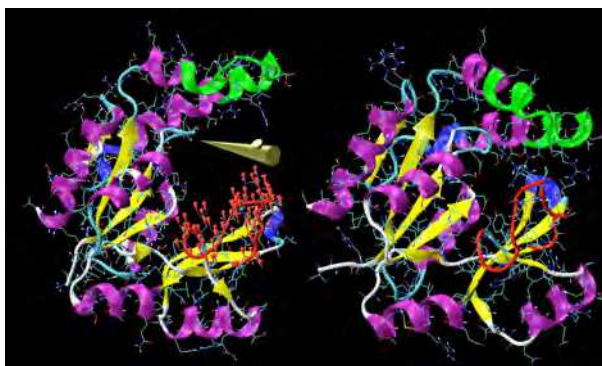

Fig. 15. Haptic control (red arrows in the red loop) of Guanylate Kinase closure. Secondary structure cartoon representation of the open state (left) and the closed state (right)

Figure 15 shows a secondary structure representation of the protein, considering specific architectural units such as the loops (white tubes), the helices (purple ribbons) and the beta sheets (yellow arrows). The crucial role of one loop (highlighted in red in Figure 15) in the initiation of GK's closure could thus be identified. It was then confirmed in a second phase using more detailed all-atom simulations. Understanding the features of this early intermediate state occurring as an impulse for the closure mechanism allows us to propose a novel mechanistic hypothesis. The loop move could be initiated by GMP docking, which may drive this loop via long range electrostatic interactions. When the loop draws closer to the

other side of the enzyme, conformational changes could be triggered, subsequently inducing a global closure of the enzyme. The interactive exploration of the simulation using the haptic modality lead us to this theoretical hypothesis. It also suggests that electrostatic interactions could be the main driving force for closure.

6.3.1 Modelling a transient stage of DNA repair by flexible docking of double stranded DNA to RecA nucleoprotein filaments

Homologous recombination is a fundamental process enabling the repair of double-strand breaks with a high degree of fidelity. In prokaryotes, it is carried out by RecA nucleofilaments formed on single-stranded DNA (ssDNA). These filaments incorporate genomic sequences that are homologous to the ssDNA and exchange the homologous strands. Due to the highly dynamic character of this process and its rapid propagation along the filament, the sequence recognition and strand exchange mechanism remains unknown at the structural level. By the interactive and flexible approach available from the BioSpring program, we investigated the possible geometries of association of the early encounter complex between RecA/ssDNA filament and double-stranded DNA (dsDNA) [Saladin et al. (2010)]. Due to the huge size of the system and its dense packing, we used a reduced representation for both protein and DNA. In this study, a systematic docking was also performed to associate dsDNA and the RecA/ssDNA complex, but this approach didn't enable the consideration of flexible regions of the nucleofilament RecA. BioSpring approach promoted to easily build a hybrid rigid-flexible representation of the molecular system by combining an Augmented Spring Network model (ASN) and a static molecular shape, and finally enabled to include very flexible L2 loops in the structure of RecA/ssDNA receptor. These flexible L2 loops constituted the only interactively controlled protein region, the rest of the RecA nucleofilament and the ssDNA were considered as static. Incoming dsDNA (ligand) was the second molecular fragment described by a flexible ASN model. During the interactive docking simulation, L2 loops moves were obtained by pulling user-selected atoms, while position and orientation of the dsDNA were controlled by acting on a fixed particles group (central nucleobases). Each single interactive simulation consisted in (i) moving L2 loops and simultaneously (ii) pulling the dsDNA toward the ssDNA, then (iii) allowing the relaxation of the system and finally (iv) saving the ligand and receptor positions.

Docking of curved dsDNA structures permitted to reach a more stable molecular complex than the one obtained from B-type DNA ligands. These simulations also demonstrate that it is possible for the double-stranded DNA to access the RecA-bound ssDNA while initially retaining its Watson-Crick pairing and emphasize the importance of RecA L2 loop mobility for both recognition and strand exchange.

6.4 ePMV : embedding molecular modeling software directly inside of professional 3D animation applications

ePMV, the Embedded Python Molecular Viewer [Johnson & Autin (2011)] is an open-source plug-in, that runs the molecular modeling software PMV [Sanner (1999)] directly inside of numerous professional 3D animation applications (hosts), to provide seamless access the capabilities of both systems and to simultaneously link the host software to other scientific algorithms. ePMV currently plugs into Maxon Cinema4D, Autodesk Maya, and Blender. Uniting host and scientific algorithms into a single interface allows users from
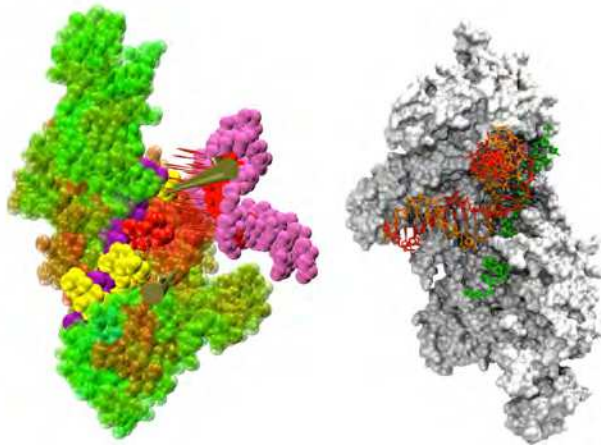
Fig. 16. Visual summary of the interactive experiments. On the left, interactive and flexible low-resolution docking of dsDNA to the RecA/ssDNA complex, using BioSpring. The two trackers enable the user to move at the same time (i) protein L2 loops (yellow or red, if selected) and (ii) dsDNA (pink and red, for selected nucleobases). Static fragments of RecA protein and ssDNA are shown in green/brown and purple spheres, respectively. On the right, all-atom model obtained after reconstruction from one of the best BioSpring prediction.

varied backgrounds to assemble professional quality visuals and to perform computational experiments with relative ease. The hybrid provides:

• high quality rendering with shadows, global illumination, ambient occlusion, etc
• intuitive GUI workflows that help users set up animations ranging from easy turntable rotations to sophisticated mechanism-of-action movies
• mesoscale modeling that allows users to illustrate or animate complex cell events in molecular detail by positioning objects with intuitive controls
• a common Python Platform that allows users to initiate sophisticated algorithms like molecular dynamics or docking energy calculations on the fly and to interoperate these algorithms with each other and with the host

The Interactive Molecular Driver [Stone et al. (2001)] and the callback action from Modeller [Eswar & Sali (2008)] enable real-time interactive molecular simulations with additional forces provided by the user. This interactive steering can operate at different levels, from selected atoms or residues, to selected curve points associated with molecular backbones. Mouse gestures and animated key frames can transmit forces or new coordinates to the simulation calculator that is linked to the host GUI via ePMV. Sophisticated host algorithms like inverse kinematics and efficient collision detection algorithms can operate on the same data as well. With this setup, a ligand can be hand-guided into a binding site with real-time docking scores provided by the Python modules of Autodock [Huey et al. (2007)]. Host-provided physics shortcuts (e.g., soft-body springs for bonds) enable interactive flexible docking with real-time scoring. At the cutting edge of molecular Augmented Reality, a user can interact with data via handheld markers tracked by a camera [Gillet et al. (2005)] to perform an interactive Rigid-body docking with intuitive midair hand gestures (see Figure 17 and http://epmv.scripps.edu/videos/structure2011).
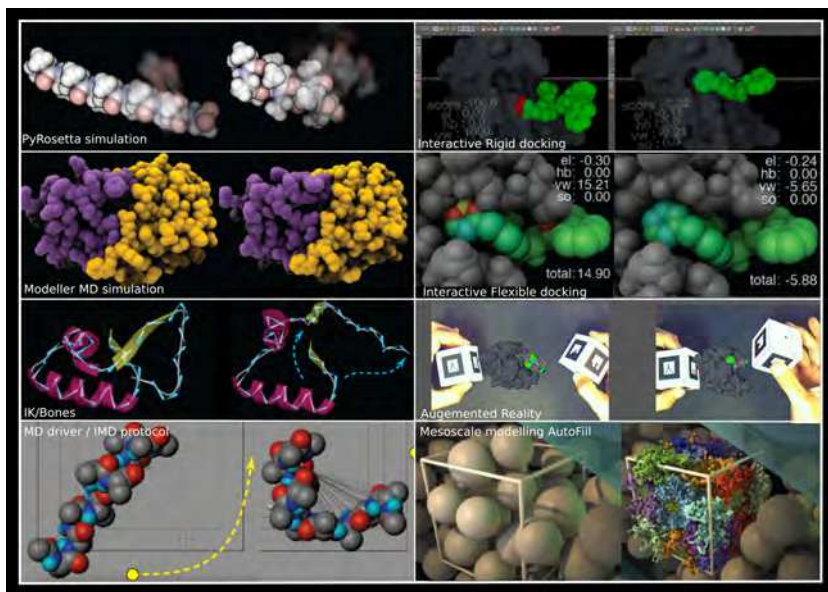
Fig. 17. ePMV features through 8 examples

6.5 FvNano: A virtual laboratory to manipulate and visualize molecular systems

The main goal of FvNano is to provide an easy to use program to manipulate molecular structures in "real time" on regular or high-performance computing (HPC) platforms. The idea is to combine molecular dynamics (MD) software with modern human-computer interaction (HCI) peripherals and GPU rendering. As previously told, combining MD with user interaction is crucial for a better understanding of the molecular motions inside a protein structure when a particular solvent is used or with an important number of active compounds. MD simulations require a lot of computing power. Hence, the ability to use high-performance computing platforms is mandatory when studying complex macromolecular systems. However, the difference between regular and massively parallel architectures can make the program hard to optimize for both platforms. This is solved by using a modular architecture based on the Flow-VR middleware **?**, http://flowvr.sourceforge.net/. In that case, MD simulation, interaction and visualization can be represented as modular blocks linked together by Flow-VR. Each of these blocks can then run on single or multiple threads according to the user's choice. Manipulating objects in a 3D environment with a 2D screen can be challenging, for that purpose, FvNano currently implements two types of HCI peripherals: SpaceBalls and haptic arms. SpaceBalls are used to move the viewpoint and haptic arms to manipulate the molecular structures with force feedback support. The visualization part is also modular, as of now two renderers are available: VMD and OpenGL. The OpenGL renderer uses the HyperBalls GPU shaders previously described. FvNano can also be used to visualize molecular trajectories computed by MD softwares with a simple user interface inspired by video players. Related work within the VMD software has recently been discussed [Stone et al. (2001)].
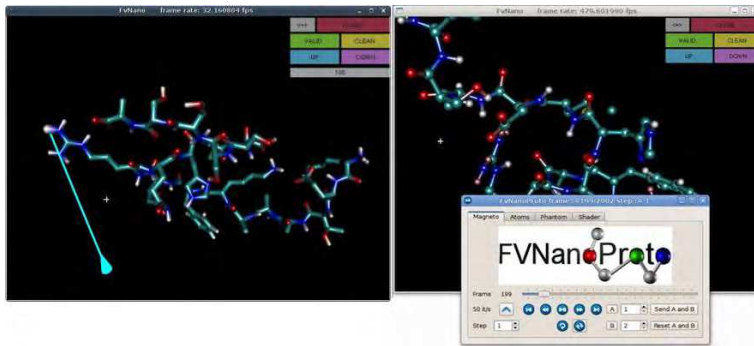
Fig. 18. Screenshot of the interactive molecular dynamics application (left). The cyan cone is the haptic arm avatar and the line shows the atom movement in progress. On the right a screenshot of the molecular trajectory reader.

7. Conclusion

Protein interactions are now routinely studied via computer simulation to understand aspects that cannot currently be studied by experiments. Recently, the Fold It! project [Cooper et al. (2010)] - a 3D-puzzle desktop game, in which the user's task is to fold proteins interactively and without any knowledge prerequisites - showed that using interactivity and insight of human minds can lead to more accurate results than pure computation. Removing false-positive results is done implicitly by users that intuitively avoid erroneous ways of molecule assembly using their experience and logic mind. More generally, molecular simulations can now benefit from this approach to reduce computation and analysis time.

In this document, we presented an interactive approach to assist scientists in their study of protein-protein docking phenomena using some advanced interaction and rendering features offered by a Virtual Reality or advanced human-computer-interaction environment. In such a context, it is important to take into account existing practices of domain experts used in their everyday work. By formalizing user needs and tasks in order to propose a limited set of design principles leading towards an appropriate tool such practices can be further improved, whilst leaving some room for them to evolve in new directions.

Through different examples, we have seen in this chapter that this goal requires efforts from many scientific domains. Experimental biologists describe the needs and validate the results that bioinformaticians extract from the analysis of simulations. Computer science experts are needed to provide efficient codes and graphics. Also, cognitive science helps to design suitable interaction paradigms and user interfaces. Interaction can be used in many applications, from rigid-body docking to accurate atomistic simulations, allowing the user to obtain a wide range of results. The novelty of our approach is that it strives to ensure continuous user participation in the process through direct manipulation of the protein models. In proposing such an approach in which users are involved both upstream and downstream from automatic docking procedures, we hope to maximize the use of their expertise. Hence, the interactive approach efficiently introduces a human element in the process and benefits from the user's experience and insight. The resemblance of this kind of applications with video games should not delude scientists to underestimate the scientific value of such techniques.

8. References

Anderson, J., Lorenz, C. & Travesset, A. (2008). General purpose molecular dynamics simulations fully implemented on graphics processing units, *J. Comp. Phys.* 227: 5342.

André, E. (2000). *Handbook of natural language processing*, chapter The generation of multimedia presentations, pp. 305–327.

Andrusier, N., Nussinov, R. & Wolfson, H. J. (2007). Firedock: fast interaction refinement in molecular docking, *Proteins* 69(1): 139–59.

Baaden, M. & Lavery, R. (2007). There's plenty of room in the middle: Multiscale Modelling of Biological Systems, *Recent Advances in Structural Bioinformatics* pp. 173–196.

Baker, N. A., Sept, D., Joseph, S., Holst, M. J. & McCammon, J. A. (2001). Electrostatics of nanosystems: Application to microtubules and the ribosome, *Proceedings of the National Academy of Science*, Vol. 98, pp. 10037–10041.

Barass, S. & Zehner, B. (2000). Responsive sonification of well-logs, *Proceedings of the International Conference on Auditory Display (ICAD'00)*.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. & Bourne, P. (2000). The protein data bank, *Nucleic Acids Research* 1(28): 235–242.

Borrelli, K., Vitalis, A., Raul Alcantara, R. & Guallar, V. (2005). Pele: Protein energy landscape exploration. a novel monte carlo based technique, *Journal of Chemical Theory and Computation* 6(1): 1304–1311.

Bourdot, P. & Touraine, D. (2002). Polyvalent display framework to control virtual navigations by 6DoF tracking, *Proceedings of the IEEE Virtual Reality International Conference (IEEE-VR'02)*, pp. 277–278.

Chavent, M., Lévy, B., Krone, M., Bidmon, K., Nominé, J., Ertl, T. & Baaden, M. (2011). Gpu-powered tools boost molecular visualization, *Briefings in Bioinformatics* .

Chavent, M., Vanel, A., Tek, A., Lévy, B., Robert, S., Raffin, B. & Baaden, M. (2011). Gpu-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids, *Journal of Computational Chemistry* 32(13): 2924–2935.

Comeau, S., Gatchell, W., Vajda, S. & Camacho, C. (2004). Cluspro: an automated docking and discrimination method for the prediction of protein complexes, *Bioinformatics* 20(1): 45–50.

Connolly, M. (1983). Analytical molecular surface calculation, *Journal of Applied Crystallography* 16: 548–558.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z. & Players, F. (2010). *Nature* 466(7307): 756–60.

Cui, Q. & Bahar, I. (2006).

Delalande, O., Ferey, N., Grasseau, G. & Baaden, M. (2009). Complex molecular assemblies at hand via interactive molecular simulations, *Journal of Computationnal Chemistry* 30(15): 2375–2387.

Delalande, O., Férey, N., Laurent, B., Gueroult, M., Hartmann, B. & Baaden, M. (2010). Multi-resolution approach for interactively locating functionally linked ion binding sites by steering small molecules into electrostatic potential maps using a haptic device, *Pacific Symposium on Biocomputing*, pp. 205–215.

Dominjon, L., Lécuyer, A., Burkhardt, J., Andrade-Barroso, G. & Richir, S. (2005). The "Bubble" Technique: Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace, *Proceedings of the World Haptics Conference (joint Eurohaptics Conference and Haptics Symposium)*.

Eswar, N., E.-D. W. B. S. M. & Sali, A. (2008). Protein structure modeling with modeller, *Methods Mol. Biol.* 426: 145?159.

Fernandez-Recio, J., Totrov, M. & Abagyan, R. (2003). Icm-disco docking by global energy optimization with fully flexible side-chains, *Cambridge, MA: Bradford Books / MIT Press* 1(52): 113–117.

Gillet, A., Sanner, M., Stoffler, D. & Olson, A. (2005). Tangible interfaces for structural molecular biology, *Structure* 13(3): 483–91.

Gottschalk, S., Lin, M. & Manocha, D. (1996). Obbtree: A hierarchical structure for rapid interference detection, *Proceedings of the 23rd Conference on Computer graphics and interactive techniques*, Vol. 30, pp. 171–180.

Grayson, P., Tajkhorshid, E. & Schulten, K. (n.d.).

Hermann, T. & Ritter, H. (1999). *Listen to your Data: Model-Based Sonification for Data Analysis*, pp. 189–194.

Hess, B., Kutzner, C. Vanderspoel, D. & Lindahl, E. (2008). Gromacs 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation., *Journal of Chemical Theory and Computation* 4(3): 435–447.

Hinsen, K. (2000). The molecular modeling toolkit: A new approach to molecular simulation, *Journal of Computational Chemistry* 21: 79–85.

Huey, R., Morris, G. M., Olson, A. J. & Goodsell, D. S. (2007). A semiempirical free energy force field with charge-based desolvation, *Journal of computational chemistry* 28(6): 1145–52.

Isralewitz, B., Baudry, J., Gullingsrud, J., Kosztin, D. & Schulten, K. (2001). Steered molecular dynamics investigations of protein function, *J. Mol. Graph.* 19: 13–25.

Johnson, D. & Willemsen, P. (2003). Six degree-of-freedom haptic rendering of complex polygonal models, *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*.

Johnson, G. & Autin, L., G. D. S. M. O. A. (2011). ePMV Embeds Molecular Modeling into Professional Animation Software Environments, *Structure* 19: 293–303.

Kitagawa, M., Dokko, D., Okamura, A. & Yuh, D. (2005). Effect of sensory substitution on suture-manipulation forces for robotic surgical systems, *Journal of Thoracic and Cardiovascular Surgery* 129(1): 151–158.

Knoll, P. & Mirzaei, S. (2003). Development of an interactive molecular dynamics simulation software package, *Rev. Sci. Instrum.* 74: 2483–2487.

LaViola, J. & Katzourin, M. (2007). An exploration of non-isomorphic 3d rotation in surround screen virtual environments, *Proceedings of the IEEE International Conference on 3D User Interfaces (IEEE-3DUI'07)*, pp. 49–54.

Leech, J., Prins, J. F. & Hermans, J. (1996). SMD : visual steering of molecular dynamics for protein design, *Computational Science and Engineering* 3: 38–45.

Leech, J., Prins, J. F. & Hermans, J. (1997). , *Physica A.* 240: 246–254.

Matthey, T., Cickovski, T., Hampton, S., Ko, A., Ma, Q., Nyerges, M., Raeder, T., Slabach, T. & Izaguirre, J. A. (2004). Protomol, an object-oriented framework for prototyping novel algorithms for molecular dynamics, *ACM Trans. Math. Softw.* 30-265(3): 237–265.

Moore, B. C. J. (2003). *An Introduction to the Psychology of Hearing*.

Nelson, M., Humphrey, W., Kufrin, R., Gursoy, A., Dalke, A., Kale, L., Skeel, R. & Schulten, K. (1995). MDscope - a visual computing environment for structural biology, *Comp. Phys. Comm.* 91: 111–133.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R. D., Kale, L. & Schulten, K. (2005). Scalable Molecular Dynamics with NAMD, *Journal of Computational Chemistry* 26: 1781–1802.

Pipe, S. (2008). Recombinant clotting factors, *Thromb Haemost* 99(5): 840–850.

Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics, *J. Comp. Phys.* 117: 1–19.

Prins, J. F., Hermans, J., Mann, G., Nyland, L. S. & Simons, M. (1999). A virtual environment for steered molecular dynamics., *Fut. Gen. Comp. Sys.* 15: 485–495.

Richard, P., Chamaret, D., Inglese, F.-X., Lucidarme, P. & Ferrier, J.-L. (2006). Human-scale haptic virtual environment for product design: Effect of sensory substitution, *International Journal of Virtual Reality* 5(2): 37–44.

Ritchie, D. (2003). Evaluation of protein docking predictions using hex 3.1 in capri rounds 1 and 2, *Proteins* 52(1): 98–106.

Saladin, A., Amourda, C., Poulain, P., Ferey, N., Baaden, M., Zacharias, M., Delalande, O. & Prevost, C. (2010). Modeling the early stage of dna sequence recognition within reca nucleoprotein filaments, *Nucleic Acid Research* 38(19): 6313–6323.

Sanner, M. F. (1999). Python: a programming language for software integration and development, *J. Mol. Graph. Model.* 17(1): 57–61.

Sanner, M., Olson, A. & Spehner, J.-C. (1996). Reduced surface: An efficient way to compute molecular surfaces, *Biopolymers* 38: 305–320.

Seeger, A. & Chen, J. (1997). Controlling force feedback over a network, *Proceedings of the Second PHANToM User's Group Workshop*.

Shoemake, K. (1985). Animating rotation with quaternion curves, *Proceedings of the 12th annual conference on Computer graphics and interactive techniques (SIGGRAPH'85)*, pp. 245–254.

Stam, J. (1999). Stable fluids, *In SIGGRAPH 99 Conference Proceedings, Annual Conference Series* 38: 121–128.

Stone, J. E., Gullingsrud, J. & Schulten, K. (2001). A system for interactive molecular dynamics simulation, *Proceedings of Interactive 3D Graphics* pp. 191–194.

Surles, M. C., Richardson, J. S., Richardson, D. C. & Brooks-JR., F. P. (1994). Sculpting proteins interactively : Continual energy minimization embedded in a graphical modelling system, *Protein Science* 3: 198–210.

Villoutreix, B., Bastard, K., Sperandio, O., Fahraeus, R., Poyet, J., Calvo, F., Deprez, B. & Miteva, M. (2008). In silico-in vitro screening of protein-protein interactions: towards the next generation of therapeutics, *Current Pharmaceutical Biotechnology* 9(2): 103–22.

Vormoor, O. (2001). Quick and easy interactive molecular dynamics using Java3D, *Comp. Sci. Eng.* 3: 98–104.

Walker, B. N. & Lane, D. M. (1994). *Auditory Display: Sonification, Audification, and Auditory Interfaces*, Westview Press.

Wang, R., Lu, Y. & Wang, S. (2003). Comparative evaluation of 11 scoring functions for molecular docking, *Journal of Medicinal Chemistry* 46: 2287–2303.

Wu, X. & Wang, S. (2002). Direct observation of the folding and unfolding of b-hairpin in explicit water through computer simulation, *J. Am. Chem. Soc.* 124: 5282–5283.

Zacharias, M. (2005). Attract: protein-protein docking in capri using a reduced protein model., *Proteins* 60(2): 252–6.
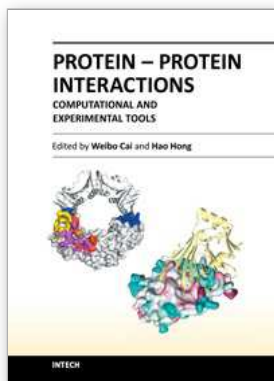
Protein-Protein Interactions - Computational and Experimental Tools
Edited by Dr. Weibo Cai

Proteins are indispensable players in virtually all biological events. The functions of proteins are coordinated through intricate regulatory networks of transient protein-protein interactions (PPIs). To predict and/or study PPIs, a wide variety of techniques have been developed over the last several decades. Many in vitro and in vivo assays have been implemented to explore the mechanism of these ubiquitous interactions. However, despite significant advances in these experimental approaches, many limitations exist such as false-positives/false-negatives, difficulty in obtaining crystal structures of proteins, challenges in the detection of transient PPI, among others. To overcome these limitations, many computational approaches have been developed which are becoming increasingly widely used to facilitate the investigation of PPIs. This book has gathered an ensemble of experts in the field, in 22 chapters, which have been broadly categorized into Computational Approaches, Experimental Approaches, and Others.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Alex Tek, Matthieu Chavent, Marc Baaden, Olivier Delalande, Patrick Bourdot and Nicolas Ferey (2012). Advances in Human-Protein Interaction - Interactive and Immersive Molecular Simulations, Protein-Protein Interactions - Computational and Experimental Tools, Dr. Weibo Cai (Ed.), ISBN: 978-953-51-0397-4, InTech, Available from: http://www.intechopen.com/books/protein-protein-interactions-computational-and-experimental-tools/advances-in-human-protein-interaction-interactive-and-immersive-molecular-simulations