

Collaborative and Parallelized Immersive Molecular Docking

Teeroumanee Nadan¹, Adrian Haffegge¹, and Kimberly Watson²

¹ Advanced Computing and Emerging Technologies Centre

² School of Biology Unit
University of Reading
Reading, RG6 6BX, United Kingdom
t.nadan@reading.ac.uk

Abstract. During docking, protein molecules and other small molecules interact together to form transient macromolecular complexes. Docking is an integral part of structure-based drug design and various docking programs are used for *in-silico* docking. Although these programs have powerful docking algorithms, they have limitations in the three-dimensional visualization of molecules. An immersive environment would bring additional advantages in understanding the molecules being docked. It would enable scientists to fully visualize molecules to be docked, manipulate their structures and manually dock them before sending to new conformations to a docking algorithm. This could greatly reduce docking time and resource consumption. Being an exhaustive process, parallelization of docking is of utmost importance for faster processing. This paper proposes the use of a collaborative and immersive environment for initially hand docking molecules and which then uses powerful algorithms in existing parallelized docking programs to decrease computational docking time and resources.

Keywords: Virtual Reality, molecular visualization, parallelization, collaborative, immersion.

1 Introduction

A receptor is a large protein molecule with a cavity/pocket/active site in which a smaller molecule, called a ligand, fits into and interacts with to form a complex molecule having complex biological functions. This interaction between a receptor and a ligand is known as a docking event. Docking can include interactions of various complexities ranging from less to more complex, such as protein-protein, protein-DNA, protein-metal and protein-ligand interactions. Various *in-silico* docking programs have been developed, each with powerful docking algorithms, to best dock the ligand in a receptor. Due to the complexities involved, the docking process is time consuming. It is therefore important to parallelize docking in order to speed up the process. However, to best reduce docking time, it is helpful to initially hand dock the ligand into the receptor's active site. Experts have

insight and can 'see' potential docking possibilities. Thus, a proper visualization medium must be available to enable experts to better understand three dimensional (3D) molecular structures and to provide intelligent guesses for starting the docking process.

This paper proposes the use of a parallelized docking algorithm in an immersive environment to aid the docking process. In an effort to make maximum use of scientists' knowledge, it has been decided to visualize and hand dock molecules in a Cave Automatic Virtual Environment (CAVE) [1]. Thus, instead of performing docking directly (i.e. in the absence of user input), users will be able to first manually dock the molecules and then computationally dock them by using docking programs. The proposed system will make use of robust algorithms from parallelized desktop-based docking programs to generate the docking results. Collaboration will be introduced to allow multiple users to view/manually dock molecules and also to view the docking results. This will allow multiple users to share their knowledge and optimally hand dock the molecules.

2 Molecular Docking Challenges and Related Work

The motivation behind this project is three-fold, namely: visualization, collaboration and parallelization.

2.1 Limited Visualization

In-silico docking programs generally are limited to desktop use, hindering scientists from fully visualizing structures in 3D and limiting the visual information. The user interface for desktop-based computational programs usually uses a mouse, reducing interaction with the molecules being visualized. However, it is crucial to be able to rotate, translate and manipulate molecules in multiple dimensions to emphasise on hidden details. For better docking, it is imperative to have a good visualization of the receptor's and the ligand's three dimensional positions and interaction with them. To overcome these limitations, it is important to provide an immersive environment which can bring all or most of the advantages of such an environment to the user. This can best be achieved in a Virtual Reality (VR) environment, enabling a better visualization of the molecules, and hence a better understanding of the receptor-ligand interactions.

By definition, VR is a computer simulation of a real or imaginary system that allows a user to perform operations on the virtual (or simulated) system with the effects being rendered in real time. The CAVE is a room-size, high resolution environment with translucent walls, onto which stereo images are projected. Shutter glasses are worn which are synchronised with the projectors to rapidly alternate between the right and left eye, enabling the wearer to perceive depth. The CAVE also has trackers which monitor the user's head position and orientation, and updates the virtual image to the current view. Within a CAVE, navigation can be achieved by using a hand-held wand. The latter is a small hardware device with a pressure-sensitive joystick and trigger buttons. The wand

allows navigation within the environment and interaction with the virtual objects themselves. Gloves can be used for interaction but can also be extended to simulate a sense of touch. Since the CAVE provides full immersion and a large field of view, it is appropriate for the visualization and manipulation of large complex biological molecules.

During drug design, *in-silico* docking is being used for screening various protein and ligand interactions for the most stable and least energy interaction. To further help reduce the time taken for the drug discovery process, it is advantageous for experts to hand dock molecules before feeding the new molecular orientations to docking programs. An immersive environment provides better insight for docking and also exploits the user's intuition. Using experience and intuition, expert scientists can easily identify cavities in a receptor and manually place a ligand into the cavity. If this hand docked orientation is fed to a docking program, the search algorithm will ultimately take less time to find the optimal orientation, hence reducing computational time.

Some immersive programs have been developed for molecular docking, such as Vibe [2] and Stalk [3]. Vibe uses a High Performance Parallel Interface (HIPPI) [4] for a high-speed transfer, providing a real-time display of molecules in a virtual environment. Stalk, on the other hand, uses parallel, distributed, heterogeneous supercomputers with high-speed networks. However, one common problem is that these programs tend to focus more processing and data transmission.

2.2 Limited Collaboration

Another criterion worth considering while dealing with molecular visualization and docking, is the element of collaboration. Within the scientific community, there is often sharing of information and experience through collaboration in order to maximise the use of resources or expediate the generation of new ideas. Many scientists are not co-located and therefore collaboration becomes difficult. For instance, if two remote collaborators are working on the same molecular visualization, it would be difficult for each other to manipulate the molecule simultaneously. This is where collaborative environments would be useful. A collaborative session would enable multiple participants to simultaneously visualize or manipulate the same molecule. Similarly, collaborative docking of molecules could greatly enhance collaborative research. Few real-time collaborative computational programs exist. This is mainly because collaboration proves pointless when users are limited on visual information. More research is done on improving those programs' docking implementation rather than adding collaboration.

AMMP-Vis [5], [6] and AMMP-EXTN [7], [8] are among the few programs that have been developed to provide for an immersive visualization and protein modeling along with a collaborative session. AMMP-Vis is a collaborative multi-view virtual environment for molecular visualization and modelling. The latter takes AMMP-Vis to a higher level by providing management of user privacy and cooperation. Unlike AMMP-Vis, AMMP-EXTN implements multiple parallel shared view sessions with different access policies. Thus, a user can create master

session which other participants can join. The problem with AMMP-Vis and AMMP-Extn is that both focus mostly on collaboration to the detriment of the docking algorithm being used.

2.3 Lack of Parallelization

Docking algorithms are implemented in two phases, namely: search and evaluation. The first step scans for the best configurational and conformational space, while the second implements a scoring function which rates and evaluates the different conformations. The search algorithms often include Monte Carlo (MC) [9], Genetic Algorithm (GA) [10] and Tabu search [11], among others, while the scoring functions often include force fields for the energies of the receptor-ligand interactions or the ligand's internal energy. For effective docking results, the search algorithm must cover 3D conformational spaces very quickly while the scoring function must quickly evaluate the different conformations among a huge population of poses (individual docked results).

In-silico docking algorithms consider docking to be either rigid or semi-flexible or flexible. In rigid docking, the bonds in both the ligand and the receptor have no freedom of movement. Semi-flexible docking algorithms, on the other hand, apply some flexibilities in the ligand's bonds, while flexible algorithms mimic more the way docking occurs in nature by allowing degrees of freedom for bonds in both the ligand and the receptor. One major problem with *in-silico* docking programs is the limitation of the docking algorithm. This is due to the fact that the use of complex algorithms such as MC and GA and the incorporation of more flexibilities in the algorithms involve a high level of computation and the use of massive computational resources to run the algorithms. To be able to deal with the computationally intensive docking algorithms and generate the results in feasible time, it is important to parallelize the whole process.

AutoDock [12] is a common program among the scientific community. According to a study carried out by Sousa et al. [13], in 2005 AutoDock was used by 27% of the scientists and since 2001 to 2005 AutoDock was ranked first among the five most commonly used docking programs, with an increased use from 36% in 2001 to 48% in 2005. Due to these various reasons and the fact that AutoDock is freely available, this docking program has been chosen for the initial phase of this project.

AutoDock can be used either for docking or virtual screening of substances. It comprises two different modules: AutoGrid and AutoDock. AutoGrid pre-calculates a set of grids describing the target protein, resulting in faster docking. These can be differentiated as grid maps and electrostatic map. While the AutoDock module performs the docking of the ligand to a set of grids. Configurational exploration is then performed. AutoDock uses both a MC algorithm and a modified version of GA, known as the Lamarckian Genetic Algorithm (LGA). Prior to running a docking job it is important to prepare a ligand and a receptor by adding hydrogens and associating atomic charges to atoms. This data is stored in a PDBQT format which is similar to the PDB [14], [15] format with an extra column to store partial atomic charges. Once AutoDock completes

a docking process, it generates a .dlg file which has the configuration of different poses ranked in order of energy interactions.

AutoDock is by default a sequential program, that is, the docking runs are executed one after the other. According to Khodade et al.[16], AutoDock takes 30 minutes to execute a docking run of 100-150 cycles. Thus, parallelization will significantly reduce the time taken for a docking task to complete. Different programs have been developed to perform parallelization of AutoDock, including the work of Khodade et al. and the program DOVIS [17].

Khodade et al. has modified the LGA algorithm in AutoDock to run in parallel. A population size of 50 with 100 and 200 runs resulted in a decrease from 81 minutes on an IBM Power-5 processor to only 1 minute on an IBM cluster of 96 processors. DOVIS, on the other hand, has a large-scale and high-throughput virtual screening technology. It uses the docking algorithm in AutoDock version 3 and runs parallel docking jobs through a queuing system on a Linux cluster and processes 500-1000 small compounds per processor on a daily basis.

3 System Description

In relation to the above problems identified, the proposed system uses immersive visualization and hand docking functionalities, along with robust docking algorithms from existing non-immersive programs, within a collaborative environment. Figure 1 depicts the system components and the interaction steps between them. The system comprises of different interfaces, namely: File Interface (FI), Visualization and Manipulation Interface (VMI), Docking Interface (DI) and Collaboration Interface(CI). The system components and the interaction steps between them are depicted in Figure 1.

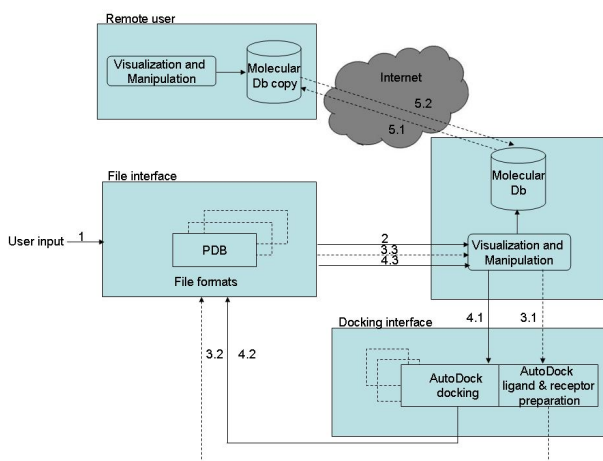


Fig. 1. System architecture

1. The system starts by accepting files from the user. Accepted file formats are PDB, PDBQT and .dlg. The PDB file has spatial data of molecules, while PDBQT and .dlg are file formats generated at an intermediate and final stage of AutoDock. The user also has the option to load multiple files in order to compare the complementarity of different ligands into the receptor's binding site. The FI reads the files which the user wants to load and also provides functionalities for saving the files in different formats.
2. VMI reads data from the file interface and projects the molecules in the CAVE. This module is being developed on VieGen [18]. Once files are loaded, the user can swap between different molecular representations, namely: Wire-frame, Stick, Ball-and-Stick and Space Filling/Corey-Pauling-Koltun (CPK). Different libraries have been created for this purpose. Manipulations can be done on the molecule, such as bond breaking and bond making. By using a wand, the user can easily rotate and move the receptor to find the cavity. Multiple ligands can be superimposed at the binding site for a clearer selection of the most appropriate ligand. Libraries of ligand fragments can be loaded and selected. These fragments can be manipulated and used for bond making purposes at the active site. Present work focuses on VMI and figure 2 shows the visualization of a CPK structural representations of the Histocompatibility Antigen (PDB id: 1HHI) in the CAVE. By rotating the structure, cavities can be easily spotted and ligands can be superimposed into a cavity.

The correlation between VMI and DI is simply to allow a user to hand dock molecules by manipulating the molecules and then feed the data to the docking algorithm.

3. Prior to the docking interface, there is an optional module which is the preparation of the ligand and the receptor. This step depends on the type of docking program the user wants to run and is therefore optional. For instance, if the user wants to use AutoDock, the user can send the hand docked molecular conformation from the VMI to the DI (step 3.1 - please refer to Figure 1) and a PDBQT file will be generated, which can then be read by the FI (3.2) and visualized by the VMI (3.3).
4. DI reads data from VMI and runs the docking algorithm (4.1). As shown in Figure 1, initial emphasis will be laid on incorporating parallel AutoDock into the CAVE environment and visualizing the results in the CAVE (4.2 and 4.3). As both parallelized versions of AutoDock, described in the previous section, are freely available, it is planned to start with the program developed by Khodade et al. Later on this interface should enable plugins for configuring the different available docking programs to be used. The challenge involved is to be able to cater for all the different input file formats required by these docking programs and then incorporate the different stages that these docking programs might have prior to the actual docking procedure, such as steps 3.1 - 3.3 for AutoDock.
5. CI will be implemented using VieGen's networking functionalities. This interface has three points of contact with VMI, allowing remote users to visualize molecules loaded by the main user. The remote user can also request from

the server a copy of molecules prepared for docking (such as step 3.3 for AutoDock), hand docked configurations and the docking results generated by the chosen docking program(5.1). The remote user application makes a copy of the molecular data and also implements the FI. For manipulation purposes, a second copy of the molecular data is done. If the remote user modifies the spatial configuration of a molecule, the new molecular configuration data are compared with the first copy stored on the local database and only changed configuration data are sent back to the server. At the server's side, a copy of the modified molecular data is saved, without overwriting the original data (5.2). At any time the main user can view the different configuration changes done by the remote user. This is done by loading unchanged configurational data from the main user's molecular data and also by loading the modified configurational data from the data received from the remote user.

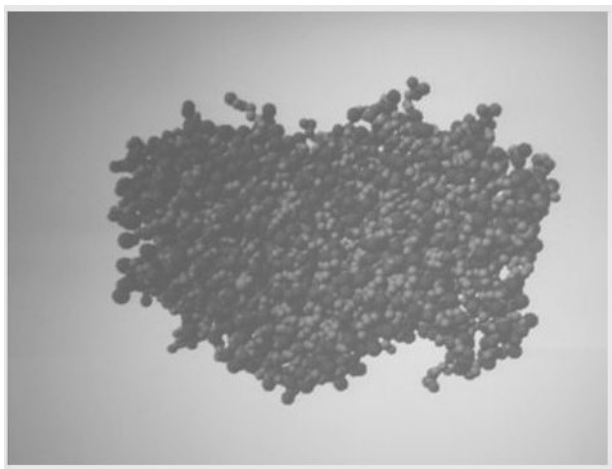


Fig. 2. CPK molecular visualization in CAVE

4 Conclusion

An immersive collaborative environment provides a high quality visualization which is beneficial for a thorough understanding and analysis of the 3D structure of molecules, particularly with respect to finding cavities and active sites in receptor molecules and trying to optimally hand dock ligands into those cavities. The CAVE provides an ideal environment to achieve such quality visualization. Once visualization and collaborative hand docking is achieved in the CAVE, robust algorithms in existing docking programs can be utilized to dock the new configuration from the user(s). Parallelizing docking programs cuts short docking algorithms running time. In this paper, a system has been described, which can

use hand docked configurations of a receptor and a ligand in a CAVE immersive environment as inputs to existing docking programs having robust and efficient docking algorithms. Molecular visualization in a CAVE has been achieved so far and integration of AutoDock is expected to be follow, along with a collaboration session of the system. Such a system would be useful in assisting drug design.

References

1. Cruz-Neira, C., Sandin, D.J., Defanti, T.A., Kenyon, R.V., Hart, J.C.: The CAVE: Audio visual experience automatic virtual environment Subsequences. *Communications of ACM* 35, 64–72 (1992)
2. Cruz-Neira, C., Langley, R., Bash, P.A.: VIBE: a virtual biomolecular environment for interactive molecular modeling Subsequences. *Computers and Chemistry* 20, 469–477 (1996)
3. Levine, D., Facello, M., Hallstrom, P., Reeder, G., Walenz, B., Stevens, F.: Stalk: An interactive system for virtual molecular docking Subsequences. *IEEE Computations Science and Engineering*, 55–56 (1997)
4. Tolmie, D., Renwick, J.: HIPPI: simplicity yields success, Subsequences. *Network IEEE* 7, 28–32 (1993)
5. Chastin, J., Zhu, Y., Brooks, J., Owen, S., Harrison, R.: A collaborative multi-view virtual environment for molecular visualization and modeling. In: 3rd International Conference on Coordinated and Multiple Views in Exploratory Visualization (CVM), pp. 77–84. IEEE, London (2005)
6. Chastin, J., Zhu, Y., Brooks, J., Owen, S., Harrison, R.: AMMP-Vis: A collaborative virtual environment for molecular modeling. In: *ACM Symposium on Virtual Reality Software and Technology*, Monterey, pp. 8–15 (2005)
7. Ma, W., Zhu, Y., Harrison, R., Owen, G.S.: AMMP-EXTN: Managing user privacy and cooperation demand in a collaborative molecule modeling virtual system. In: *Virtual Reality Conference VR 2007*, pp. 301–302. IEEE, Los Alamitos (2007)
8. Ma, W.: AMMP-EXTN: a user privacy and collaboration control framework for a multi-user collaboratory virtual reality system. Masters Thesis, Georgia State University (December 2007)
9. Abagyan, R.A., Totrov, M.M.: Biased probability Monte Carlo conformation searches and electrostatic calculations and peptides and proteins Subsequences. *J. Mol. Biol.* 235, 983–1002 (1994)
10. Holland, J.: *Adaptation in Natural and Artificial Systems Infrastructure*. University of Michigan Press, Ann Arbor (1975)
11. Bland, J.A., Dawson, G.P.: Tabu search and design optimization Subsequences. *Computer Aided Design* 23, 195–201 (1991)
12. Goodsell, D.S., Olson, A.J.: Automated docking of substrates to proteins by simulated annealing Subsequences. *Proteins: Structure, Function and Genetics* 8, 195–202 (1990)
13. Sousa, S.F., Fernandes, P.A., Ramas, M.J.: Protein-ligand docking: current status and future challenges Subsequences. *Proteins: Structure, Function and Bioinformatics* 65, 15–26 (2006)
14. Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer Jr, E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., Tasumi, M.: Protein Data Bank: A computer-based archival file for macromolecular structures Subsequences. *J. Mol. Biol.* 112, 534–542 (1997)

15. Berman, H.M., Westbrook, J., Feng, Z., Gillilands, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank Subsequences. *Nucleic Acids Research* 28, 235–242 (2000)
16. Khodade, P., Prabhu, R., Chandra, N., Raha, S., Govindarajan, R.: Parallel Implementation of Autodock Subsequences. *J. Appl. Cryst.* 40, 598–599 (2007)
17. Zhang, S., Kumar, K., Jiang, X., Wallqvist, A., Reifman, J.: DOVIS: an implementation for high-throughput virtual screening using AutoDock Subsequences. *BMC Bioinformatics* 9, 126 (2008)
18. Haffegee, A.: VieGen: an accessible toolset for the configuration and control of virtual environments Ph.D Thesis, University of Reading (2008)