

Computer-aided drug design platform using PyMOL

Markus A. Lill · Matthew L. Danielson

Received: 13 August 2010 / Accepted: 20 October 2010 / Published online: 30 October 2010
© Springer Science+Business Media B.V. 2010

Abstract The understanding and optimization of protein-ligand interactions are instrumental to medicinal chemists investigating potential drug candidates. Over the past couple of decades, many powerful standalone tools for computer-aided drug discovery have been developed in academia providing insight into protein-ligand interactions. As programs are developed by various research groups, a consistent user-friendly graphical working environment combining computational techniques such as docking, scoring, molecular dynamics simulations, and free energy calculations is needed. Utilizing PyMOL we have developed such a graphical user interface incorporating individual academic packages designed for protein preparation (AMBER package and Reduce), molecular mechanics applications (AMBER package), and docking and scoring (AutoDock Vina and SLIDE). In addition to amassing several computational tools under one interface, the computational platform also provides a user-friendly combination of different programs. For example, utilizing a molecular dynamics (MD) simulation performed with AMBER as input for ensemble docking with AutoDock Vina. The overarching goal of this work was to provide a computational platform that facilitates medicinal chemists, many who are not experts in computational methodologies, to utilize several common computational techniques germane to drug discovery. Furthermore, our software is open source and is aimed to initiate collaborative efforts among computational researchers to combine other open source

computational methods under a single, easily understandable graphical user interface.

Keywords Computer-aided drug discovery · Molecular modeling · Docking · Scoring · Free energies · PyMOL

Introduction

Computer-aided drug design (CADD) has become an indispensable tool in the pharmaceutical industry and academia over the last decades [1–4]. Two of the most commonly used methodologies in structure-based CADD are docking and molecular mechanics techniques. Docking has been predominantly applied to the problems of lead identification and binding mode prediction during the drug development process. Molecular mechanics methods such as minimization and molecular dynamics (MD) simulations are primarily used to refine protein-ligand complexes, analyze the complexes' dynamics, and predict free energies of binding.

A huge body of CADD software has been developed over the years in many different academic research groups [5–13]. The software is often open source and usually provided by the authors at zero cost or a nominal fee to academic institutions. While this body of easily accessible software provides a great opportunity to perform research in CADD, two main problems are often encountered by medicinal chemists trying to use the software. First, in many instances the authors of CADD software focus on scientific details rather than the usability of their software by other researchers. Graphical user interfaces (GUIs) are seldom associated with the programs, making the software difficult to use for other researchers. This is particularly true for medicinal chemists, structural biologists, and other

M. A. Lill (✉) · M. L. Danielson
Department of Medicinal Chemistry and Molecular
Pharmacology, Purdue University, 575 Stadium Mall Drive,
West Lafayette, IN 47907, USA
e-mail: mlill@purdue.edu

researchers who may use software to understand the structure-function relationship of proteins, design small molecular probes and lead candidates, and conduct structure-activity relationship studies but are not specialists in computational approaches. Second, even if an excellent GUI exists, e.g. for AutoDock [6], each individual program usually requires a specific input structure and produces a specific output format not recognized by other computational programs. When this situation occurs, novice users are not capable of combining different programs such as using MD simulations to refine docking poses [14–17] or to perform ensemble docking based on a MD trajectory [18–20].

We aimed to develop a single computational framework, or GUI, that allows novice computational users to easily apply and combine molecular modeling methods designed to aid in drug discovery. Our framework is based on PyMOL, an open-source and widely used biomolecular visualization program. The choice to utilize PyMOL was based on the fact that it is user friendly, well documented, widely used, has high-quality rendering capacities, and in particular has the powerful utility to allow for access and manipulation of the data structure of molecules stored in PyMOL using Python plugins.

In this publication we focus on the integration of different structure-based design methods into the PyMOL environment. In particular, we developed software that facilitates running docking simulations using AutoDock Vina [5] and SLIDE [13] in addition to molecular mechanics simulations using the Amber package [7, 10]. Preparation of the biomolecule, monitoring the simulations, data analysis, and post-processing features are included and the combination of docking with molecular mechanics approaches is made easily accessible. The details of each individual Python plugin for molecular mechanics and the two docking programs are discussed in the Methods section followed by demonstrating how molecular mechanics and docking is interfaced in our computational framework. As most academic programs in CADD have been developed in a Linux environment, our computational framework is currently supported under this operating system only.

Methods

Software interfacing PyMOL with the molecular mechanics program Amber [7, 10] and the docking programs AutoDock Vina [5] and SLIDE [13] was developed using the Python programming language. The plugins are added to the startup folder of PyMOL, allowing it to automatically load the software and display our interface as three additional submenus titled Amber, AutoDock Vina, and

SLIDE. All software, except the Amber package, is free of charge for academic users but must be downloaded separately from the corresponding authors websites and installed (see [Necessary molecular modeling software](#)). Although not completely free, an affordable license for Amber can be obtained for academic, non-profit, or government users.

Molecular mechanics using Amber

Figure 1 (left) displays the workflow implemented in the PyMOL plugin, titled Amber, to perform molecular mechanics simulations. A protein structure file (usually a structure from the PDB database [21, 22] but other file formats are possible as well) is loaded into PyMOL. A typical protein structure usually contains several rotamer states of His, Asn, or Gln side chains that cannot be unambiguously identified based on the electron density of the X-ray experiment alone [23, 24]. The program Reduce [23] is called automatically from PyMOL to analyze the hydrogen bond network of the protein, determine the protonation states of His residues, and optimize rotamer states of His, Asn, and Gln side chains. The coordinates of the modified side chain rotamers are automatically updated in the PyMOL screen and the His protonation states are altered in PyMOL matching the Amber naming convention (HID for δ -protonated, HIE for ϵ -protonated, and HIP for doubly protonated His residues).

This procedure is automatically performed in the background without any need of user intervention and the results are shown in a separate text window. Histidine residue naming, and therefore protonation state, can be manually changed using an additional submenu of the Amber plugin. Following protein preparation, the Amber package [7, 10] is utilized to perform molecular mechanics applications initiated from PyMOL. Two different forms of communication between PyMOL and Amber are implemented: an interactive mode designed to rapidly test new drug design ideas and a background mode to perform long MD simulations to accurately estimate binding affinities. In the interactive mode, the protein is first prepared by calling the Amber module `tleap`. Missing residue types, generally cofactors, need user-defined force field parameters in the form of standard `.lib` and `.frcmod` files. Such files are centrally stored in a subdirectory, titled `AMBER_library`, and after definition are automatically recognized by the preparation procedure for any protein containing this cofactor. Located within the `AMBER_library` directory, the files `cofactors.txt` and `bonding.txt` define all existing cofactors and possible bonds between atoms of the cofactors and protein residues. The `bonding.txt` file also contains the definition of disulfide bridges. Disulfide bridges are then automatically recognized and the disulfide cysteine

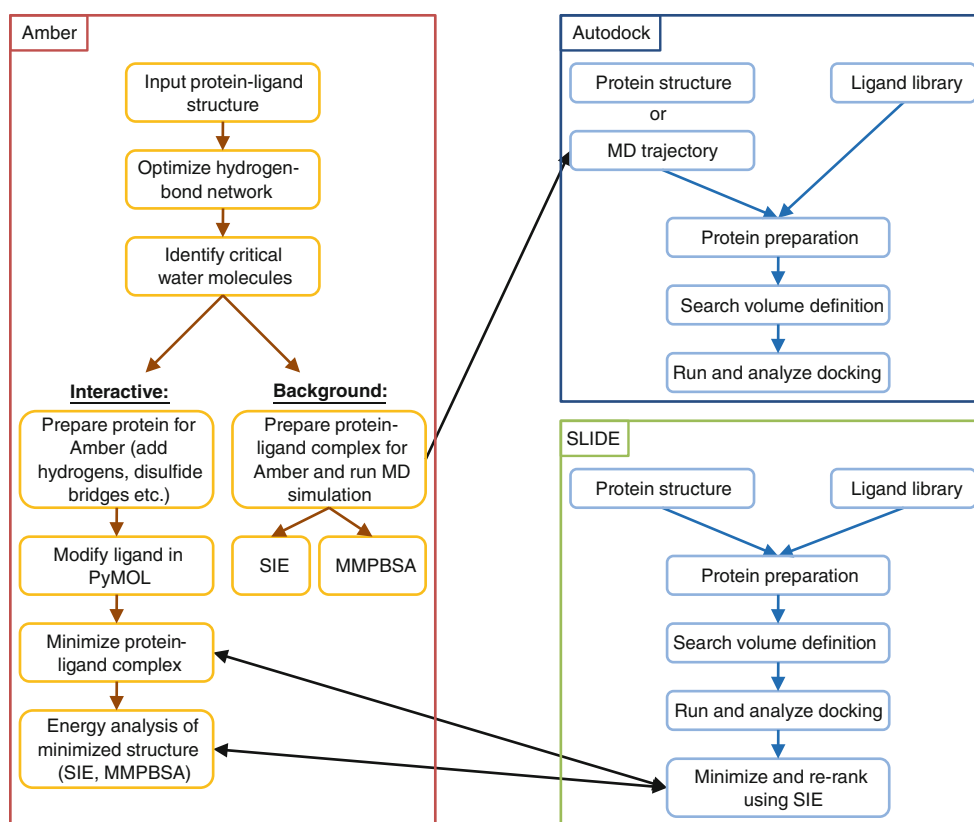


Fig. 1 Workflow of PyMOL scripts automating molecular modeling using AMBER and docking with AutoDock Vina and SLIDE. Interaction between the individual software is displayed by *open black arrows*

residues are renamed to the Amber-specific CYX nomenclature, thus generating a disulfide bond. Bonds between cofactors and protein residues, i.e. cysteine-heme contacts, are generated in a similar manner. Hydrogens missing in the protein structure are added by Amber and the protein is automatically updated within PyMOL while conserving the original residue numbering. Following protein preparation, the co-crystallized or docked ligand can be visually modified and the resulting ligand-protein system can be minimized interactively. Free energy estimates of the refined protein-ligand structure can be automatically estimated using the sum of van der Waals, electrostatic interactions, and Poisson–Boltzmann Surface Area calculations (MMPBSA) [25, 26] or alternatively using the Solvated Interaction Energies method (SIE) [27] (Fig. 2). The interactive feature is well-suited to quickly estimate binding affinity increase or decrease due to small modifications made to the scaffold of a ligand.

To more accurately predict binding energies, the Molecular-Mechanics Poisson–Boltzmann Surface Area method (MMPBSA) combined with an improved entropy estimate [28] and SIE can be automatically applied to a molecular dynamics trajectory ran in the background mode. First, the protein-ligand system is prepared similar to the

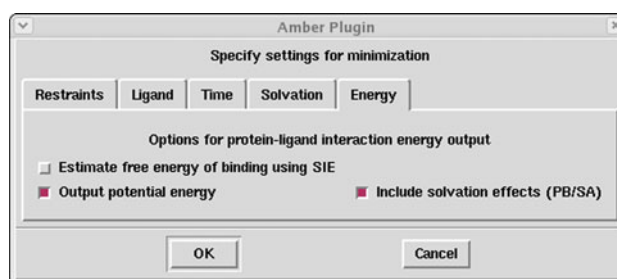
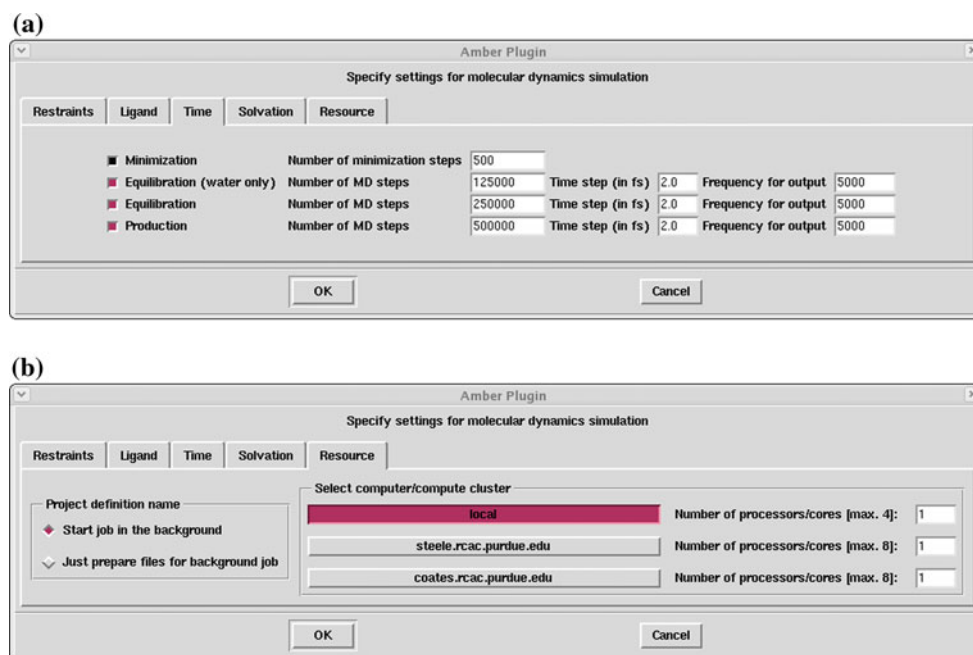


Fig. 2 Screenshot of the notebook dialog used to select options during interactive energy minimizations and free energy estimates

previously described procedure for the interactive molecular mechanics session but the background mode allows additional solvation options: cubic box with periodic boundary conditions, a solvation cap, and implicit solvation using the OBC model [29]. The MD simulation can be run on a local desktop or on any computer cluster using a PBS queuing system (Fig. 3). The queue and PBS settings are specified by the user in a global settings file.

The location of the MD output is stored in a monitoring file that allows a user to check the progress of the MD simulation (informing the user if the simulation has completed or not) and re-import the results back into PyMOL

Fig. 3 Screenshots of the notebook dialog utilized to select options for background MD simulations. The notebook pages defining (a) simulation times and (b) selecting computational resources are shown here



anytime after the MD simulation finishes. If the simulation was ran on a remote computer cluster and not locally on a desktop, the MD trajectory files are automatically transferred from the cluster and subsequent free energies can be estimated using SIE or MMPBSA.

Following the completion of a MD simulation and subsequent file transfer if needed, SIE energy analysis can be initiated by reading the monitoring file and subsequently selecting the starting and ending frames along with an interval between frames from the MD trajectory. The program *sietraj* is started in the background and energy information is displayed on screen in the form of a pop-up dialog after completion. When applying MMPBSA post-processing, all necessary input files for Amber's *mmpbsa* program are generated including a trajectory file where all water and user-selected non-amino acids, e.g. counter ions, are removed. To calculate the enthalpic energy contribution, the molecular mechanics energy combined with PBSA solvation is computed between the ligand and protein. Entropic contributions to binding affinity are estimated using normal mode (*nmode*) analysis [30–32]. To improve the accuracy of predicting the entropic energy contribution, we have implemented a fully automatic version of the recent *nmode* procedure of Ryde and co-workers [28].

Docking using AutoDock Vina

Recently, Seeliger et al. [11] have published a PyMOL plugin for AutoDock and AutoDock Vina. We will only briefly describe the features of our plugin, titled AutoDock Vina, as it possesses similar functionality to the plugin from Seeliger et al. We chose to incorporate AutoDock

Vina into our computational platform due to its computational efficiency, a key consideration when performing ensemble docking studies. The AutoDock Vina plugin generates ligand libraries by exporting all ligand objects present in a PyMOL session. The ligand objects can either be directly generated using the PyMOL builder or read in from other sources (e.g. the ZINC [33] and KEGG [34] ligand databases). A lexicon of exported libraries is stored and each library of compounds can later be modified, combined with other libraries, and imported for docking to different target proteins.

In AutoDock Vina, a box is placed over a section of the protein to define the docking search volume. Using our plugin the position and size of the box can be defined based on a user-defined selection of ligands or amino acids in PyMOL (Fig. 4) and visually modified as the box is graphically displayed in PyMOL. The user has the ability to select protein residues to remain flexible during docking visually in PyMOL. Flexible residues can be selected based on their position within the docking search volume, by an automatic selection of residues within a user-defined radius around a user selection (e.g. all protein residues within 5 Å of one or several ligands), or by selecting known critical amino acids important to binding.

All docking simulations are run in the background allowing the user to close PyMOL while the simulation progresses. Following completion of the docking run, the results are imported into PyMOL in a similar manner as previously described for MD simulations, and all docking poses are automatically displayed to the user. A separate dialog displays all docked solutions with their associated docking score.

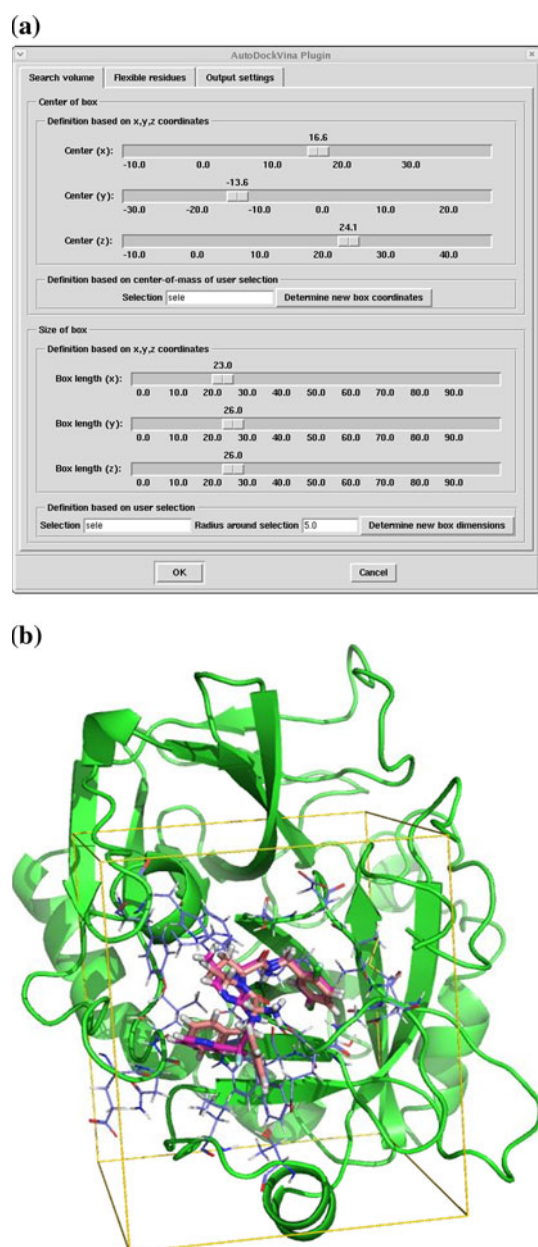


Fig. 4 Screenshot of the notebook dialog used to define the docking search volume (a), flexible residues, and output settings. The corresponding box defining the docking search volume for a protein is shown below (b)

Integrating Amber and AutoDock Vina

Although PyMOL plugins for individual molecular modeling programs such as Amber [7, 10] and AutoDock [6] are very valuable, our software has the additional capability to integrate different programs into one useable tool. This opens new opportunities for molecular modeling, such as performing ensemble docking and utilizing molecular dynamics simulations to refine binding poses and to more accurately predict binding affinities of compounds.

The latter is easily performed by utilizing our Amber plugin and PyMOL as a graphical element connecting docking to MD simulations.

Ensemble docking [18–20] has been put forward as an approach to include protein flexibility beyond the standard side-chain level during docking. The AutoDock Vina plugin allows the user to import an NMR style PDB file containing multiple protein structures or a single Amber trajectory file with multiple frames. If a trajectory is used as input, a QT-cluster algorithm [35] has been implemented to select the most diverse set of protein structures from the input trajectory. The user has the ability to predefine a target number of clusters (also known as protein templates or snapshots) and an initial minimal root mean square deviation (RMSD) between cluster representatives (center of clusters). The algorithm then automatically adjusts the RMSD value between cluster centers to obtain a total number of clusters within 50% of the user-defined target number of clusters (e.g. User defined 20 cluster representatives, the program will select between 10 and 30 structures according to the RMSD threshold specified). Subsequently, AutoDock Vina then automatically docks a ligand library into all cluster representatives. For each ligand in the library, the predicted binding poses from all templates are pooled and are automatically clustered based on a user-defined RMSD criterion to remove “similar” binding modes between protein templates. The resulting clustered binding poses are then visually displayed in PyMOL and a score analysis, including a histogram of scores and standard deviation, is displayed (Fig. 5).

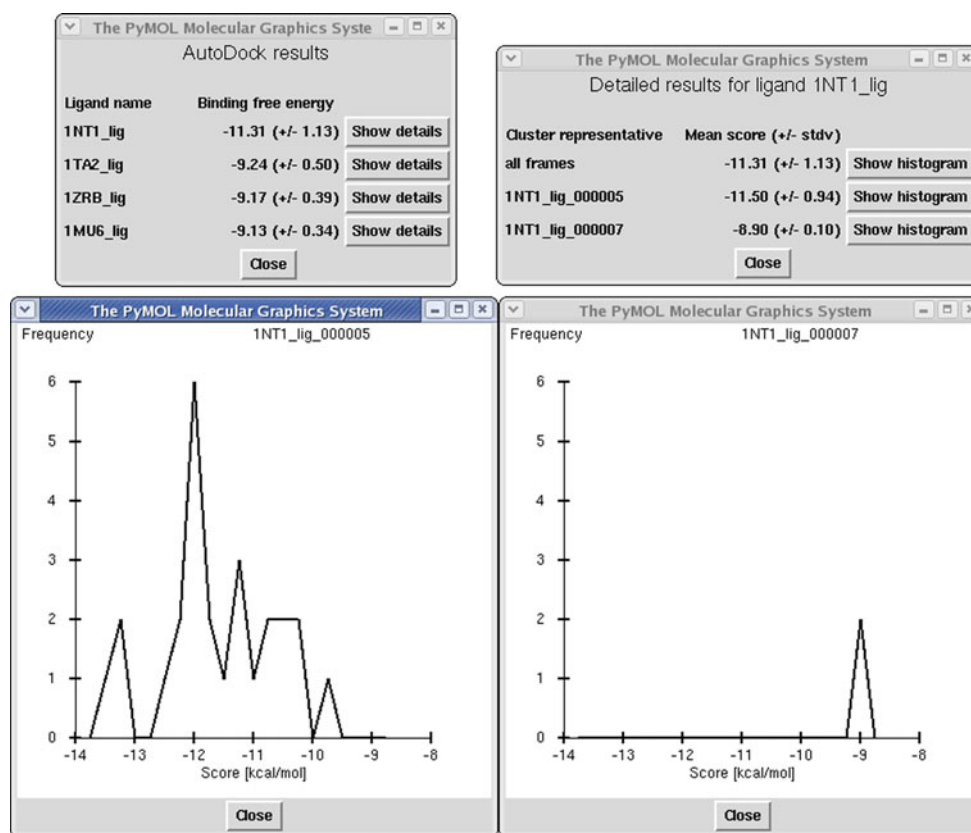
Docking using SLIDE

Titled Slide, the plugin integrating the docking program SLIDE [13] into PyMOL handles ligand libraries similar to the previously described AutoDock Vina plugin. The docking search volume can be interactively defined to be a sphere around a fixed point defined in Cartesian coordinates or around a user-defined selection of ligands. The inherent SLIDE pharmacophore selection can be biased by an analysis of the interaction between protein and a user-defined selection of ligands in PyMOL. SLIDE is then automatically started from PyMOL in the background and the results are analyzed as described above for AutoDock Vina.

Integrating SLIDE and SIE

The standard SLIDE implementation only outputs the top-ranked pose for each ligand. We have compiled a modified SLIDE version that outputs a user-defined number of poses. The ensemble of possible binding poses can be automatically post-processed using Amber minimization

Fig. 5 Results for four ligands docked to thrombin using ensemble docking with AutoDock Vina. *Top, left* Ligands sorted according to mean predicted free energy of binding with standard deviation. *Top, right* Details for ligand 1NT1_lig. Mean score with standard deviations for all frames and for the two identified clusters with cluster representatives (1NT1_lig_000005, 1NT1_lig_000007 representing frames 5 and 7). *Bottom* Histograms of identified scores of all members of individual clusters



and subsequent SIE [27] scoring including a Poisson-Boltzmann solvation model. The binding poses are thus re-ranked according to the SIE scoring function and detailed energy analysis is displayed to the user. Such a procedure allows users to rank binding poses and ligand sets using a more expensive and accurate scoring method that includes a detailed physical model of solvation effects.

Conclusions

Three PyMOL plugins (Amber, AutoDock Vina, and SLIDE) to popular molecular modeling programs have been presented, allowing researchers with limited computational background in areas such as structural biology, medicinal chemistry, physics, and biochemistry to easily utilize and integrate each of these programs. Initiation and analysis of different simulations from a common powerful GUI not only removes a novice's hesitation to use novel molecular modeling tools but also makes it easier to control and derive valuable information from the simulations.

Special emphasis has been placed on fully integrating different molecular modeling programs (without requiring user manipulation) to perform new types of computational

experiments that each individual program does not provide. In this paper we have demonstrated examples of our integration process by post-processing AutoDock Vina [5] docking poses using more sophisticated scoring schemes (MMPBSA/SIE) [25–27] and performing ensemble docking on molecular dynamics trajectories. In the future, we plan to continue our automatic integration process and hope to initiate similar collaborative approaches between other computational researchers. The plugins discussed above will be made available free of charge from http://people.pnhs.purdue.edu/~mlill/software/pymol_plugins.

Necessary molecular modeling software

The following third-party programs are necessary for complete utilization of the described plugins: the Amber package [7, 10] (tested with Amber 9 and Amber 10) to perform molecular mechanics applications and MMPBSA calculations, *changepdb* and *changecrd* from U. Ryde to estimate entropic contributions to the free energy of binding, *sietraj* to calculate free energies of binding using the SIE [27] methodology, *babel* [36] to convert between different file formats, *Reduce* [23] to optimize the hydrogen-bond network in proteins, and AutoDock Vina [5] and SLIDE [13] to perform docking studies.

Acknowledgments We would like to thank Ulf Ryde for access to the programs changepdb and changecrd that facilitate the estimation of entropic contributions to the free energy of binding.

References

- Alvarez JC (2004) High-throughput docking as a source of novel drug leads. *Curr Opin Chem Biol* 8:365–370
- Green DV (2003) Virtual screening of virtual libraries. *Prog Med Chem* 41:61–97
- Ooms F (2000) Molecular modeling and computer aided drug design. Examples of their applications in medicinal chemistry. *Curr Med Chem* 7:141–158
- van de Waterbeemd H (2005) From in vivo to in vitro/in silico ADME: progress and challenges. *Expert Opin Drug Metab Toxicol* 1:1–4
- Trott O, Olson AJ (2010) Software news and update AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem* 31:455–461
- Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ (2009) AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J Comput Chem* 30:2785–2791
- Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, Onufriev A, Simmerling C, Wang B, Woods RJ (2005) The Amber biomolecular simulation programs. *J Comput Chem* 26:1668–1688
- Danielson ML, Lill MA (2010) New computational method for prediction of interacting protein loop regions. *Proteins* 78:1748–1759
- Hess B, Kutzner C, van der Spoel D, Lindahl E (2008) GROMACS 4: Algorithms for highly efficient, load-balanced and scalable molecular simulation. *J Chem Theory Comput* 4:435–447
- Pearlman DA, Case DA, Caldwell JW, Ross WS, Cheatham TE, Debolt S, Ferguson D, Seibel G, Kollman P (1995) Amber, a package of computer-programs for applying molecular mechanics, normal-mode analysis molecular-dynamics and free-energy calculations to simulate the structural and energetic properties of molecules. *Comput Phys Commun* 91:1–41
- Seeliger D, de Groot BL (2010) Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *J Comput Aided Mol Des* 24:417–422
- Soto CS, Fasnacht M, Zhu J, Forrest L, Honig B (2008) Loop modeling: sampling, filtering, and scoring. *Proteins* 70:834–843
- Zavodszky MI, Sanschagrin PC, Korde RS, Kuhn LA (2002) Distilling the essential features of a protein surface for improving protein-ligand docking, scoring, and virtual screening. *J Comput Aided Mol Des* 16:883–902
- de Molfetta FA, de Freitas RF, da Silva AB, Montanari CA (2009) Docking and molecular dynamics simulation of quinone compounds with trypanocidal activity. *J Mol Model* 15:1175–1184
- Graves AP, Shivakumar DM, Boyce SE, Jacobson MP, Case DA, Shoichet BK (2008) Rescoring docking hit lists for model cavity sites: predictions and experimental testing. *J Mol Biol* 377:914–934
- Manetti F, Locatelli GA, Maga G, Schenone S, Modugno M, Forli S, Corelli F, Botta M (2006) A combination of docking/dynamics simulations and pharmacophoric modeling to discover new dual c-Src/Abl kinase inhibitors. *J Med Chem* 49:3278–3286
- Okimoto N, Futatsugi N, Fujii H, Suenaga A, Morimoto G, Yanai R, Ohno Y, Narumi T, Taiji M (2009) High-performance drug discovery: computational screening by combining docking and molecular dynamics simulations. *PLoS Comput Biol* 5:e1000528
- Paulsen JL, Anderson AC (2009) Scoring ensembles of docked protein:ligand interactions for virtual lead optimization. *J Chem Inf Model* 49:2813–2819
- Park IH, Li C (2010) Dynamic ligand-induced-fit simulation via enhanced conformational samplings and ensemble dockings: a survivin example. *J. Phys. Chem. B* 114:5144–5153
- Hritz J, de Ruiter A, Ostenbrink C (2008) Impact of plasticity and flexibility on docking results for cytochrome P450 2D6: a combined approach of molecular dynamics and ligand docking. *J Med Chem* 51:7469–7477
- Berman H, Henrick K, Nakamura H, Markley JL (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single uniform archive of PDB data. *Nucleic Acids Res* 35:D301–D303
- Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. *Nucleic Acids Res* 28:235–242
- Word JM, Lovell SC, Richardson JS, Richardson DC (1999) Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *J Mol Biol* 285:1735–1747
- Weichenberger CX, Sippl MJ (2006) NQ-Flipper: validation and correction of asparagine/glutamine amide rotamers in protein crystal structures. *Bioinformatics* 22:1397–1398
- Srinivasan J, Cheatham TE, Cieplak P, Kollman PA, Case DA (1998) Continuum solvent studies of the stability of DNA, RNA and phosphoramidate—DNA helices. *J Am Chem Soc* 120:9401–9409
- Massova I, Kollman PA (1999) Computational alanine scanning to probe protein–protein interactions: a novel approach to evaluate binding free energies. *J Am Chem Soc* 121:8133–8143
- Naim M, Bhat S, Rankin KN, Dennis S, Chowdhury SF, Siddiqi I, Drabik P, Sulea T, Bayly CI, Jakalian A, Purisima EO (2007) Solvated interaction energy (SIE) for scoring protein–ligand binding affinities. 1. exploring the parameter space. *J Chem Inf Model* 47:122–133
- Kongsted J, Ryde U (2009) An improved method to predict the entropy term with the MM/PBSA approach. *J Comput Aided Mol Des* 23:63–71
- Onufriev A, Bashford D, Case DA (2004) Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins* 55:383–394
- Lamm G, Szabo A (1986) Langevin modes of macromolecules. *J Chem Phys* 85:7334–7348
- Kottalam J, Case DA (1990) Langevin modes of macromolecules—applications to Crambin and Dna hexamers. *Biopolymers* 29:1409–1421
- Case DA (1994) Normal-mode analysis of protein dynamics. *Curr Opin Struct Biol* 4:285–290
- Irwin JJ, Shoichet BK (2005) ZINC—a free database of commercially available compounds for virtual screening. *J Chem Inf Model* 45:177–182
- Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M (2010) KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res* 38:D355–D360
- Heyer LJ, Kruglyak S, Yooseph S (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res* 9:1106–1115
- Guha R, Howard MT, Hutchison GR, Murray-Rust P, Rzepa H, Steinbeck C, Wegner J, Willighagen EL (2006) The Blue Obelisk—interoperability in chemical informatics. *J Chem Inf Model* 46:991–998