J Mol Model (2017) 23: 304
https://doi.org/10.1007/s00894-017-3475-9

SOFTWARE REPORT

LiGRO: a graphical user interface for protein–ligand molecular dynamics

Luciano Porto Kagami [1] · Gustavo Machado das Neves [1] · Alan Wilter Sousa da Silva [2] · Rafael Andrade Caceres [3,4] · Daniel Fábio Kawano [5,6] · Vera Lucia Eifler-Lima [1]

Received: 14 July 2017 / Accepted: 17 September 2017 / Published online: 4 October 2017

Abstract To speed up the drug-discovery process, molecular dynamics (MD) calculations performed in GROMACS can be coupled to docking simulations for the post-screening analyses of large compound libraries. This requires generating the topology of the ligands in different software, some basic knowledge of Linux command lines, and a certain familiarity in handling the output files. LiGRO—the python-based graphical interface introduced here—was designed to overcome these protein–ligand parameterization challenges by allowing the graphical (non command line-based) control of GROMACS (MD and analysis), ACPYPE (ligand topology builder) and PLIP (protein-binder interactions monitor)—programs that can be used together to fully perform and analyze the outputs of complex MD simulations (including energy minimization and NVT/NPT equilibration). By allowing the calculation of linear interaction energies in a simple and quick fashion, LiGRO can be used in the drug-discovery pipeline to select compounds with a better protein-binding interaction profile. The design of LiGRO allows researchers to freely download and modify the software, with the source code being available under the terms of a GPLv3 license from http://www.ufrgs.br/lasomfarmacia/ligro/.

Keywords Gromacs · Graphical user interface · Protein-ligand · Molecular dynamics and ACPYPE

✉ Luciano Porto Kagami
luciano.kagami@ufrgs.br

✉ Vera Lucia Eifler-Lima
veraeifler@ufrgs.br

[1] Laboratory of Medicinal Organic Synthesis (LaSOM), Faculty of Pharmacy, Federal University of Rio Grande do Sul, Ipiranga Avenue, n° 2752, Porto Alegre, RS 90610-000, Brazil

[2] Universal Protein Resource (UniProt), EMBL-EBI, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

[3] Pharmacosciences Department of Federal University of Health Sciences of Porto Alegre (UFCSPA), Porto Alegre, Brazil

[4] Health Sciences Postgraduate Program of Federal University of Health Sciences of Porto Alegre (UFCSPA), Sarmento Leite Street, n° 245, 2nd Floor, room 200C, Porto Alegre City, Rio Grande do Sul State, Brazil

[5] Faculty of Pharmaceutical Sciences, University of Campinas (UNICAMP), Cândido Portinari Street, n° 200, Campinas, SP 13083-871, Brazil

[6] Institute of Chemistry, University of Campinas (UNICAMP), Josué de Castro Street, Campinas, SP 13083-970, Brazil

Introduction

Background

Molecular dynamics (MD) simulations are valuable techniques used in chemistry, physics and material science. MD tools enable the natural evolution of molecular systems to be studied over time, allowing the prediction of static and dynamic properties directly from the underlying interactions. MD simulations are frequently used to explore conformational aspects of proteins and nucleic acids, providing valuable information concerning the fluctuations of these systems. Additionally, they also offer a significant scope in protein–ligand interactions, speeding up and lowering the costs of the drug development pipeline by aiding the identification and optimization of lead compounds [1].

In this context, MD can be coupled to molecular docking simulations (which provide a faster initial approach to explore the vast ligand conformational space) for the post-screening of large compound libraries, enabling the analyses of the

different target conformations, the refinement of the final complexes structures and the calculation of more accurate ligand binding energies, critical issues in the rational drug-design process. By treating both ligands and proteins as flexible entities, MD simulations are proper tools for studying the induced fitting of the target around the docked ligand, also allowing the assessment of the role of water molecules in these interactions through the calculation of binding free energies [2].

The GROMACS package developed by Lindahl et al. [3] is probably the most common suite used to perform MD simulations with protein–ligand complexes, although the software is still not able to construct a topology for non-amino acid molecules. For building the topology of the ligands, the researcher must use another program such as ACPYPE (AnteChamber PYthon Parser interfacE), a tool based on ANTECHAMBER employed to calculate partial charges, generate topologies and parameters in different formats for non-amino acid compounds. A test with > 17,000 small molecules from the Protein Data Bank attested the robustness of the method, ascribing to ACPYPE an efficiency of > 76% [4].

Although GROMACS may be considered a very robust package, being a command line-based software, it requires some basic knowledge of the UNIX shell. This particular aspect, associated with the not so straightforward handling of its output files, may represent an extra hurdle for pure chemists, pharmacists and biologists, then preventing these researchers from using GROMACS more efficiently. To overcome these limitations, several graphical user interfaces (GUIs) have been designed for GROMACS in the last decade [5–8], but none is specific for complex protein–ligand parameterization. Herein, we report the development of LiGRO, aiming to merge the protein/ligand parameterization and MD tools in a single, easy-to-use, python-based GUI. Its design takes advantage of the extensive set of commands of the Linux platform, which, when organized into so-called "shell scripts," provide the user with a set of options to manipulate and interact with the system. The development of the GUI involved the use of python program language [9] and the Tk GUI toolkit [10], as python is one of the most popular languages for scientific computing, and, when associated with proper scientific libraries, can produce high-level interactivity tools [11].

Methods

Implementation

LiGRO for GROMACS, a Linux-based software, was written in python programming language, version 2.7 while Tkinter was used for displaying the program graphics. The software is installed using the "LiGRO_Install.py" tool, which will install all the necessary dependencies for the appropriate program operation. GROMACS, ANTECHAMBER (AmberTools [12], ACPYPE, PLIP [13] and python libraries (Numpy

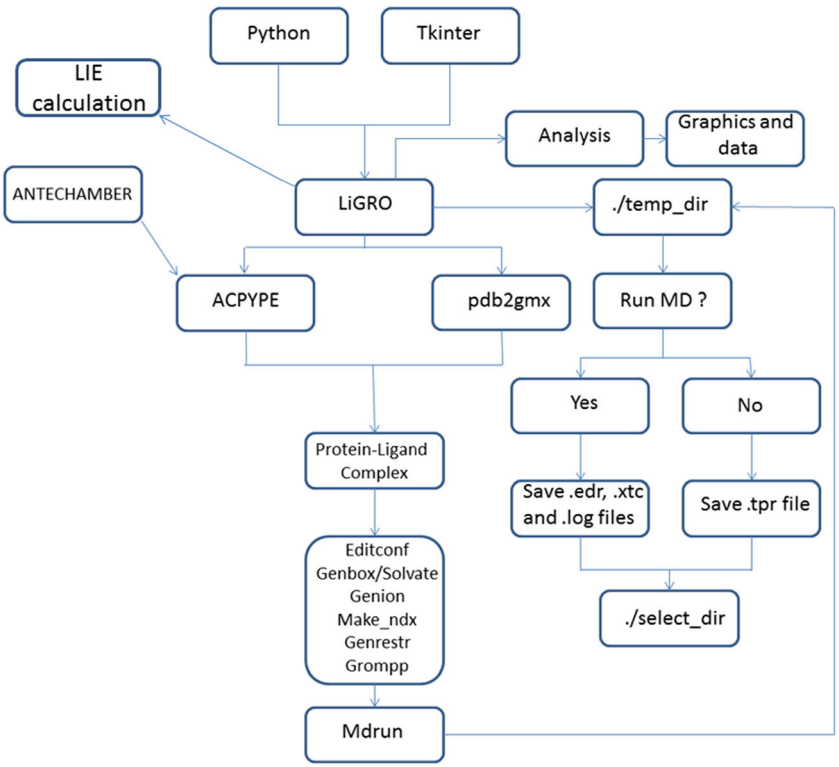

Fig. 1 LiGRO workflow

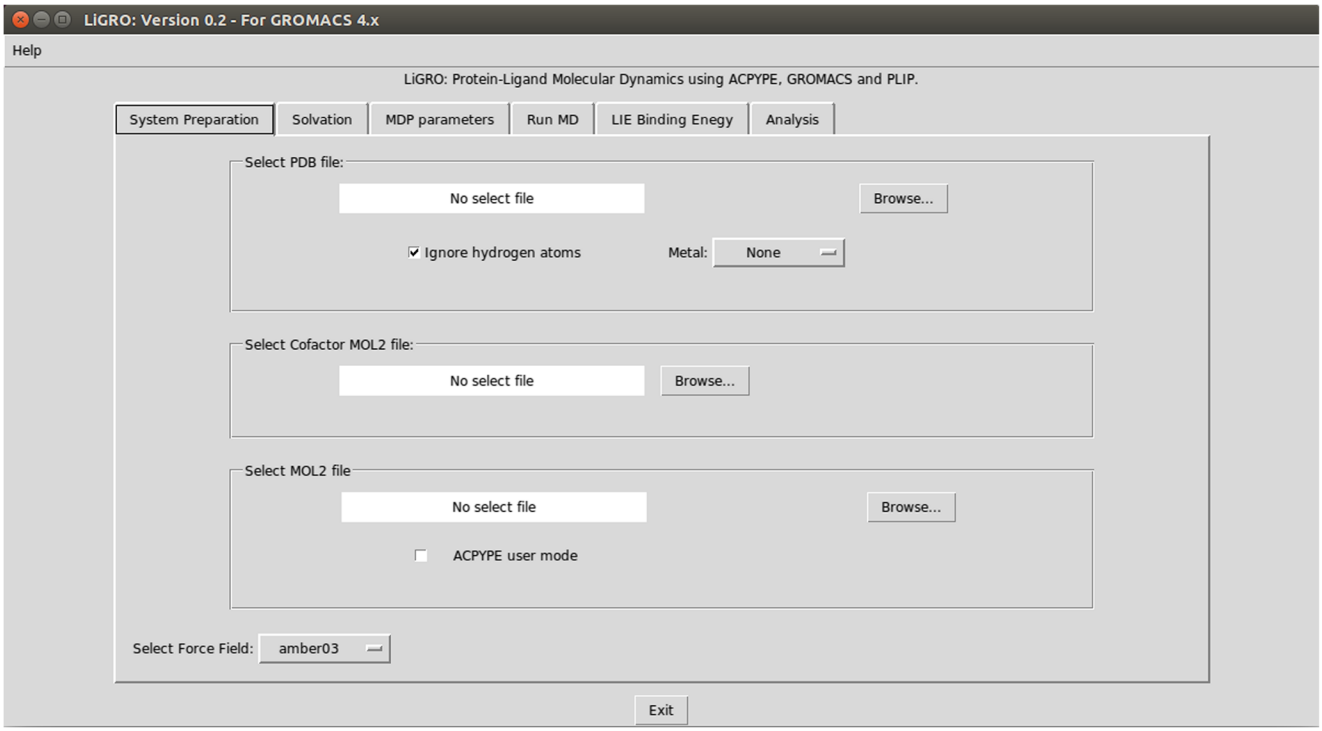

Fig. 2 LiGRO graphical user interface (GUI)

[14], PANDAS [15] and Matplotlib [16]) are installed automatically using a user-friendly graphical interface.

The LiGRO workflow (Fig. 1) illustrates how python and Tkinter (low system libraries responsible for providing graphical interactivity to the user) were linked to the LiGRO script—the program layer that controls MD functions. All files created by the LiGRO program are saved into a temporary directory, and, subsequently, the user can save these files into a selected directory. ACPYPE is an ANTECHAMBER-dependent program, and, together with pdb2gmx, can be used

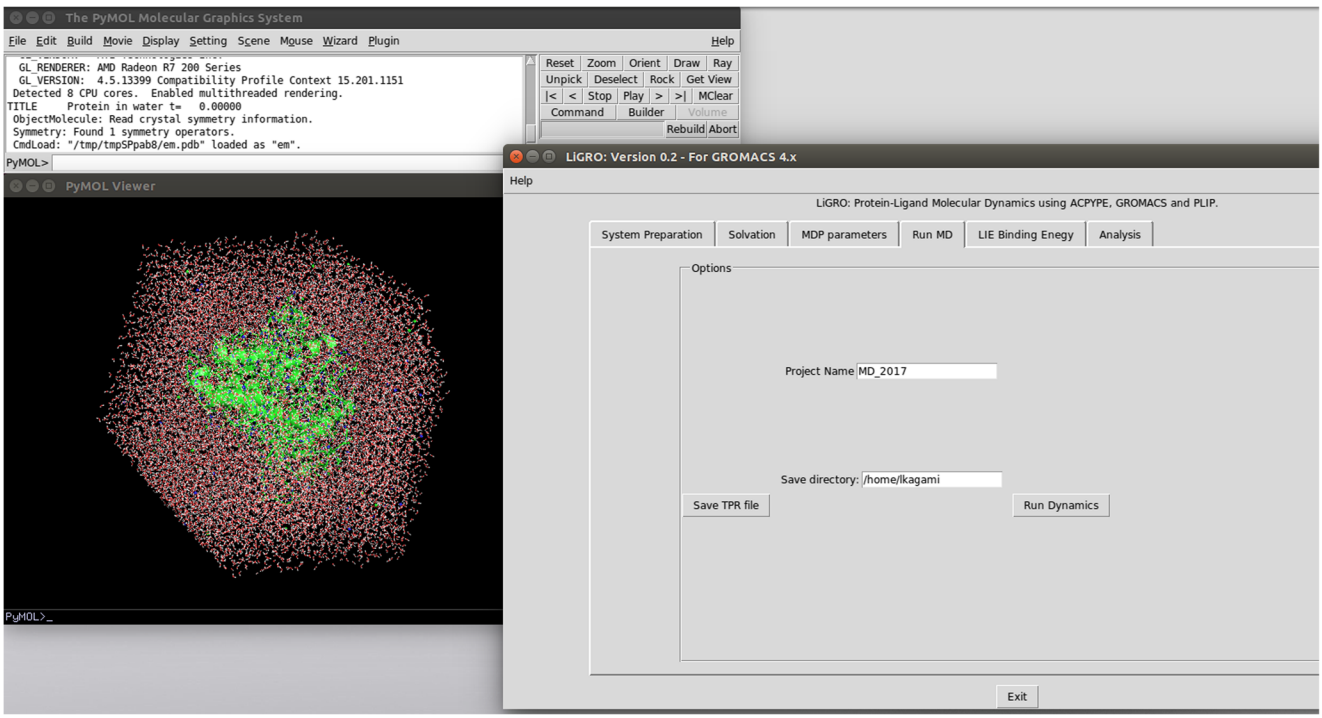

Fig. 3 Molecular dynamics (MD) simulation of *Leishmania major* pteridine reductase 1 in complex with NADP and biopterin (PDB-ID:2BF7)

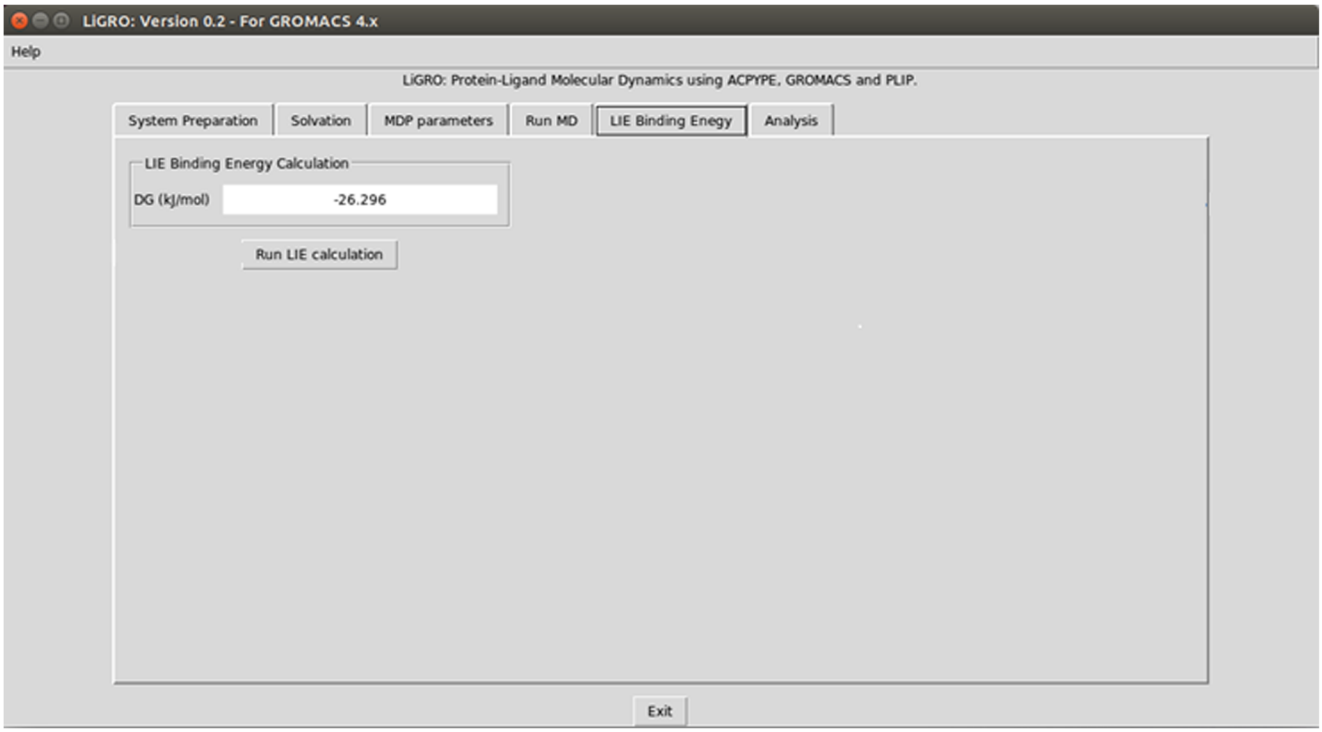

Fig. 4 LiGRO linear interaction energy (LIE) calculation module

to build complex topologies. Integration of LiGRO and ACPYPE allows the user to choose between AMBER03, AMBER94, AMBER96, AMBER99, AMBER99SB, AMBER99SB-ILDN, AMBERGS and OPLS-AA / L force fields for MD simulations.

The LiGRO script for simulations (Fig. 2) includes nine programs from the GROMACS package (pdb2gmx, editconf, genbox/solvate, grompp, genion, mdrun, genrestr, make_ndx and trjconv). By using this script, it is possible to run a complete MD simulation (minimization, NVT, NPT and MD simulation) with an easy-to-use input from the researcher. The user just needs to provide some parameters: select a protein PDB file and a Ligand MOL2 file; select a water model and a box type; set the salt concentration to neutralize the system (or input the desired amount of sodium/chlorine ions); and parameterize the MDP files (simulation time, number of steps and temperature). The LiGRO GUI allows the user to impute two non-amino acid structures, one cofactor and one ligand. LiGRO will consider only the structure imputed as the 'ligand' for trajectory analysis.

The user can check if the protein is properly placed in the simulation box with the Pymol molecular viewer (Fig. 3). The script also accounts for long-range electrostatic interactions by setting the Coulomb energies to particle mesh Ewald (PME) to stabilize the trajectory during the simulation [17]. Finally, the user can save the .tpr file to run in a GROMACS cluster.

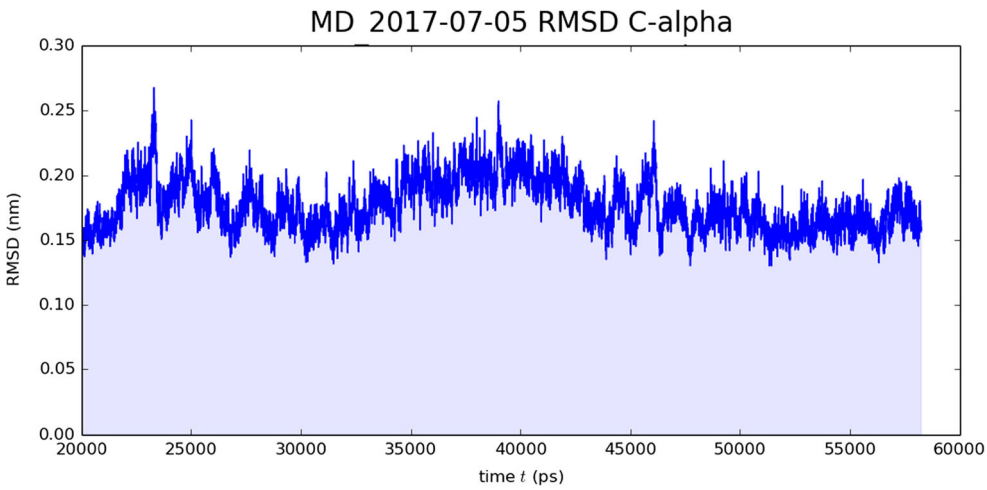

Fig. 5 LiGRO analysis of a root mean square deviation (RMSD) C-alpha plot

In the linear interaction energy (LIE) binding energy calculation module (Fig. 4), it is possible to calculate the binding affinity prediction using the LIE method, which provides values that can be used to select compounds with a better protein-binding interaction profile [18]. The GROMACS LIE requires preliminary calculations of the Lennard-Jones (LJ) and Coulomb non-bonded energies in two simulations: one with the molecule of interest coupled to its receptor and one with the solvated molecule; in LiGRO all these operations can be performed through a single button click.

The analysis module of LiGRO generates several plots (Fig. 5) necessary for the evaluation of the simulation outputs. Three GROMACS package programs compose this script: rmsdist, msd, h_bond, energy and gyrate. With these programs, it is possible to generate the RMSD, MSD, radius of gyration and hydrogen bond number plots, while the user can visualize these graphics through a simple button click. The short-range electrostatic contributions (LJ and Coulomb) can be calculated by choosing the LJSR-CoulSR IE in the analysis module. In addition, LiGRO provides the minimum, maximum and average values for these analyses using the Numpy python library. The integration with protein–ligand interaction profiler (PLIP) confers to LiGRO a complete tool to describe a protein-ligand profiler.

LiGRO also enables the analyses of protein-ligand interaction profiles through the course of the simulation by using the PLIP program. This software provides the ability to monitor ligand interactions, including hydrogen/halogen bonds, hydrophobic contacts, π-stacking/π-cation interactions and salt/water bridges, at any time during the simulation.

Conclusions

LiGRO provides a novel user-friendly graphical interface for protein-ligand MD simulations using GROMACS. This program eliminates the need for users to write command-lines and to manually construct complex topology files. For the ligand setting, sLiGRO uses ACPYPE—a tool-based on the ANTECHAMBER software that generates automatic topologies, parameters and partial charges for small molecules for use in a variety of MD programs. By using LiGRO, the user needs to enter only a few inputs to generate a complete protein-ligand MD simulation (complex energy minimization and NVT/NPT equilibration). LiGRO automatically unwrap the trajectory output, leaving it ready for the trajectory analysis.

With LiGRO, it is less likely for the user to encounter parameterization errors or to break the MD simulation. Currently, LiGRO is the only GUI capable of running protein-ligand simulations in GROMACS that is fully compatible with all the available versions. The LiGRO script follows the Bevan laboratory tutorial Protein-Ligand Complex [19] as its workflow. Users must remember that the default settings are not always the best choice for a MD simulation since these simulations are often complex operations that require more parameters to obtain better results. Researchers may freely download and modify the software, with the source code being available under the terms of the GPLv3 license from http://www.ufrgs.br/lasomfarmacia/ligro/.

Acknowledgments The authors thank the Brazilian funding agencies CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul), FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and INCT-IF (Instituto Nacional de Ciência e Tecnologia para Inovação Farmacêutica) for their financial support.

References

1. Zoete V, Cuendet MA, Grosdidier A, Michielin O (2011) SwissParam: a fast force field generation tool for small organic molecules. J Comput Chem 32(11):2359–2368
2. Alonso H, Bliznyuk AA, Gready JE (2006) Combining docking and molecular dynamic simulations in drug design. Med Res Rev 26(5):531–568
3. Van Der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJ (2005) GROMACS: fast, flexible, and free. J Comput Chem 26(16):1701–1718
4. da Silva AWS, Vranken WF (2012) ACPYPE-Antechamber python parser interface. BMC Res Notes 5(1):367
5. Sellis D, Vlachakis D, Vlassi M (2009) Gromita: a fully integrated graphical user interface to gromacs 4. Bioinform Biol Insights 3:99
6. Kota P (2007) GUIMACS—a Java based front end for GROMACS. In Silico Biol 7(1):95–99
7. Makarewicz T, Kaźmierkiewicz R (2016) Improvements in GROMACS plugin for PyMOL including implicit solvent simulations and displaying results of PCA analysis. J Mol Model 22:109
8. Makarewicz T, Kaźmierkiewicz R (2013) Molecular dynamics simulation by GROMACS using GUI plugin for PyMOL. J Chem Inf Model 53(5):1229–1234
9. van Rossum G, de Boer J (1991) Interactively testing remote servers using the Python programming language. CWi Quart 4(4): 283–303
10. Ousterhout JK (1994) Tcl and the Tk toolkit. Professional computing series. Addison-Wesley, Boston
11. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12(Oct):2825–2830
12. Salomon-Ferrer R, Case DA, Walker RC (2013) An overview of the Amber biomolecular simulation package. Wiley Interdiscip Rev Comput Molec Sci 3(2):198–210
13. Salentin S, Schreiber S, Haupt VJ, Adasme MF, Schroeder M (2015) PLIP: fully automated protein–ligand interaction profiler. Nucleic Acids Res 43(W1):W443–W447
14. Walt SVD, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. Comput Sci Eng 13(2):22–30
15. McKinney W (2014) Pandas, Python Data Analysis Library. 2015 Reference Source. http://pandas.pydata.org/

16. Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9(3):90–95
17. Cheatham TI, Miller J, Fox T, Darden T, Kollman P (1995) Molecular dynamics simulations on solvated biomolecular systems: the particle mesh Ewald method leads to stable trajectories of DNA, RNA, and proteins. J Am Chem Soc 117(14):4193–4194
18. Gutiérrez-de-Terán H, Aqvist J (2011) Linear interaction energy: method and applications in drug design. Methods Mol Biol 819: 305–323
19. Lemkul JA (2015) GROMACS Tutorials. http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/ Accessed 20 February 2017