



Zinovjev, K., & Van Der Kamp, M. W. (2020). Enlighten2: molecular dynamics simulations of protein-ligand systems made accessible. *Bioinformatics*, 36(20), 5104-5106. [btaa643].
<https://doi.org/10.1093/bioinformatics/btaa643>

Peer reviewed version

Link to published version (if available):
[10.1093/bioinformatics/btaa643](https://doi.org/10.1093/bioinformatics/btaa643)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via Oxford University Press at <https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btaa643/5873591?searchresult=1>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: <http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Enlighten2: molecular dynamics simulations of protein-ligand systems made accessible

Kirill Zinovjev¹ and Marc W. van der Kamp^{1,*}

¹School of Biochemistry, University of Bristol, Biomedical Sciences Building, University Walk, Bristol BS8 1TD, UK

*To whom correspondence should be addressed.

Abstract

Motivation: Experimental structural data can allow detailed insight into protein structure and protein-ligand interactions, which is crucial for many areas of bioscience, including drug design and enzyme engineering. Typically, however, little more than a static picture of protein-ligand interactions is obtained, whereas dynamical information is often required for deeper understanding and to assess the effect of mutations. Molecular dynamics (MD) simulations can provide such information, but setting up and running these simulations is not straightforward and requires expert knowledge. There is thus a need for a tool that makes protein-ligand simulation easily accessible to non-expert users.

Results: We present Enlighten2: efficient simulation protocols for protein-ligand systems alongside a user-friendly plugin to the popular visualization program PyMOL. With Enlighten2, non-expert users can straightforwardly run and visualize MD simulations on protein-ligand models of interest. There is no need to learn new programs and all underlying tools are free and open source.

Availability: The Enlighten2 Python package and PyMOL plugin are free to use under the GPL3.0 licence and can be found at <https://enlighten2.github.io>. We also provide a lightweight Docker image via DockerHub that includes Enlighten2 with all the required utilities.

CONTACT: marc.vanderkamp@bristol.ac.uk

1 Introduction

Biomolecular simulation, and especially molecular dynamics (MD) simulation, is increasingly used for understanding and designing biological molecules and their interactions (Huggins, et al., 2019). Atomically detailed, physics-based MD simulation methods are demonstrating growing impact in areas as diverse as biocatalysis, biomaterials, biotechnology, and drug design. Such simulations offer uniquely detailed insights at high time- and space- resolution into mechanisms, dynamics, and processes. Particularly important for the (re)design of enzymes as biocatalysts is assessment of the effect of mutations, for which static structural modelling alone is of limited use.

Although automated MD simulations are available in other visualization programs, such as YASARA Structure (Land and Humble, 2018), the QwikMD plugin to VMD (Ribeiro, et al., 2016) or commercial program packages such as MOE or Discovery Studio, these still require users to obtain and get used to a new, unfamiliar program and interface. Other options, such as the GROMACS Plugin to PyMOL (pymol.org, Schrödinger, LLC, 2018) and the Molecular Dynamics Simulation option in UCSF

Chimera (Pettersen, et al., 2004), still require expert knowledge (for ligand parameterization and MD simulation) and/or the non-trivial installation of MD software. There is thus a need for user-friendly, low-barrier access to MD simulation of protein-ligand complexes through a visualization program familiar to most structural biologists.

As a first attempt to meet this need, we initially developed efficient protocols (implemented as shell-scripts) and a basic PyMOL plugin (for the legacy Tkinter GUI) as Enlighten (<https://github.com/marcvanderkamp/enlighten>). These protocols have already been used to analyze enzyme-substrate interactions and dynamics (Rice, et al., 2019; Wang, et al., 2018) as well as for preparation for quantum mechanical / molecular mechanical (QM/MM) simulations of enzyme reactions (Byrne, et al., 2016; Drulyte, et al., 2019).

Now, we introduce Enlighten2, which includes a Python package for the protocols, a new user-friendly PyMOL plugin and, notably, a Docker image with all the required software that allows seamless running of

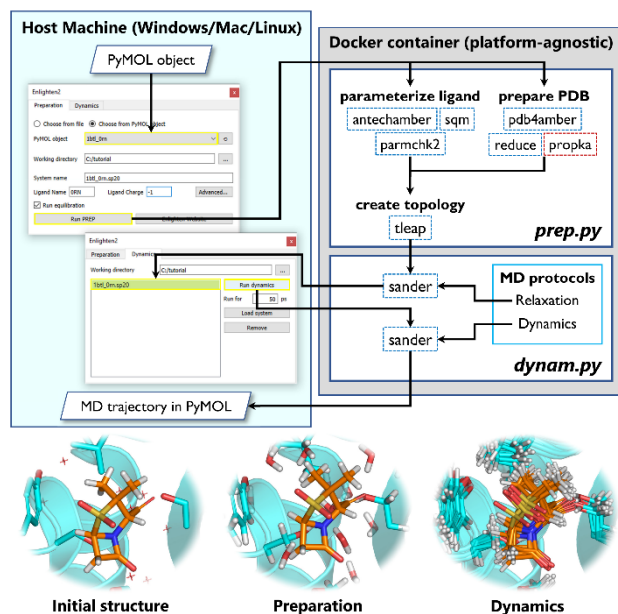


Fig. 1. Overview of the Enlighten2 PyMOL plugin and tools. Top: workflow for setting up and running protein-ligand MD simulation, with PyMOL plugin GUI windows on the right, and utilities run inside the Docker container on the left. *prep.py* and *dynam.py* are the Python wrappers. Bottom: example views going from protein-ligand structure (TEM-1 β -lactamase [PDB ID: 1BTL] with sulbactam from Enlighten2 tutorial 1) to a system prepared for MD simulation and dynamics output.

simulations on Windows, Mac and Linux operating systems without non-trivial software installation. Therefore, apart from PyMOL itself, the plugin only requires Docker to be installed and running. In addition, both the Python package and the plugin give users increased flexibility over the original Enlighten implementation. Enlighten2 thereby enables experimentalists to easily perform basic MD simulations of their protein-ligand system of interest through a familiar and open-source interface, without requiring the knowledge for using or manual installation of an MD software package or utilities.

2 Methods and implementation

Enlighten2 consists of three main parts (see Fig. 1):

1. A Python package with wrappers for AmberTools (AMBER 2018, University of California, San Francisco; Case *et al.*, 2018) programs and pre-defined protocols for enzyme-ligand system preparation, relaxation and production dynamics. The package automates the usage of external programs for system preparation (AmberTools and propka3.1 (Sondergaard, *et al.*, 2011); see Fig. 1). Details can be found on the project GitHub repository page (<https://github.com/vanderkamp/enlighten2>). It also includes protocols for two types of MD simulations: relaxation (consists of restrained gradual minimization, simulated annealing and final minimization) and dynamics (consists of heating, production MD and final minimization). Relaxation must be run prior to dynamics. The protocols are time-efficient, because only the dynamics of a (user-defined) sphere around the ligand is included – sufficient for protein-ligand interactions but not for e.g. large-scale conformational change.
2. A Docker image (<https://hub.docker.com/r/kzinovjev/enlighten2>) with the preinstalled Enlighten2 package and a reduced version of AmberTools including antechamber, parmchk2 and sqm – for parameterization of the ligand (Wang *et al.*, 2006) with the second version of the General Amber Force Field (GAFF2) (Wang, *et al.*, 2004); pdb4amber, reduce (Word, *et*

al., 1999) and tleap – for preparation of Amber topology and coordinate files; sander – for MD simulations. It also includes propka3.1 for assignment of protonation states for titratable residues. While it is possible to use the Docker image directly to access the command line tools provided by the Enlighten2 Python package, its main role is to provide access to these tools through the PyMOL plugin. It allows to run Enlighten2 on any platform without the need to install any software other than Docker.

3. A GUI plugin for PyMOL that consists of two tabs: “Preparation” and “Dynamics” (Fig. 1). The “Preparation” tab allows the user to prepare and relax a protein-ligand model system based on a PyMOL object or a PDB file. The advanced options allow for fine tuning of the preparation protocol (setting size of the sphere, pH of the simulation etc.). The “Dynamics” tab is dedicated to running dynamics of specified duration as well as to management of previously run simulations (including loading trajectories to PyMOL). The user is required to provide a protein-ligand model with hydrogens added to the ligand. When the Enlighten2 plugin is used for the first time, it will automatically instruct Docker to fetch the Docker image, such that the user does not need to install MD software.

3 Features

Enlighten2 provides the following functionality:

- Automatic ligand parameterization (through Antechamber) – the user must only specify the ligand name and charge.
- Run relaxation (to remove initial clashes etc.) and MD simulation of a desired length. Resulting trajectories are loaded in PyMOL when finished.
- Support for cofactors by providing additional parameter files as required (explained in Tutorial 2, see below).
- Add a solvation sphere with arbitrary size and position, in addition to any solvent atoms already present in the starting model. The center of the sphere can be picked through the PyMOL GUI.
- Set a specific pH value for the simulation (by using propka to determine the protonation states of titratable residues at a desired pH).

Both the Enlighten2 PyMOL plugin and the Python package are implemented such that new features can be added easily. We expect future users to provide feedback on what additional functionality is most desirable. The input and output files for all the tools used during system preparation and molecular dynamics are stored, allowing more experienced users to introduce arbitrary modifications and reuse the prepared system for modified or more demanding simulations, e.g. on HPC clusters.

Enlighten2 is specifically designed for basic and efficient protein-ligand simulations and therefore currently has some limitations. At least one non-covalently bound ligand is required. Automatic parameterization of (additional) covalently bound ligands, cofactors or custom residues is not supported, but the corresponding parameters can be provided manually. (Currently, covalent bonds between protein and non-protein, e.g. cofactor, residues cannot be added through the plugin.) The solvation and MD simulation are limited to a sphere, with all atoms outside the sphere fixed (to avoid the computational cost of simulating fully solvated systems with periodic boundary conditions). This means that large conformational changes or full ligand dissociation cannot be studied. Finally, since Enlighten2 ligand parameterization relies on Antechamber, Enlighten2 only supports the ligand chemistries covered by the General Amber Force Field, GAFF.

4 Usage examples

Two detailed tutorials for novice users are available on the project website (<https://enlighten2.github.io>). Both expect no previous experience with PyMOL or MD simulation. Tutorial 1 features running simulations on a model system constructed from two PDB structures, where one structure contains the protein-ligand complex and the other is an apo-structure of the homolog to be simulated. This tutorial familiarizes the user with basic PyMOL operations such as atom selection, addition of hydrogens to the ligand and deletion of undesirable co-crystallized molecules required prior to running an MD simulation. Tutorial 2 shows how to include a common co-factor (NADPH) alongside a ligand. The parameters and topology for NADPH are also provided on the tutorial webpage (along with their original source).

For more experienced users that would like to use the Enlighten2 Python package directly (e.g. to set-up simulations for HPC systems), usage examples are provided in the GitHub repository. Further, all intermediate files are conveniently stored in separate folders, allowing an experienced user to introduce any input modifications prior to rerunning the corresponding codes manually (e.g. for setting up periodic boundary simulations or introducing covalent bonds between proteins and cofactors) or upload the input files to run simulations on HPC systems.

5 Availability

The Enlighten2 PyMOL plugin code can be downloaded directly from the project website (<https://enlighten2.github.io>) and then added to PyMOL through the PyMOL Plugin Manager. With Docker running in the background, no further software installation is required. The website also features an overview of the package, the installation instructions and the two tutorials described above. The source code for the Enlighten2 Python package and the Dockerfile used to create the Docker image are publicly available as a GitHub repository (<https://github.com/vanderkamp/enlighten2>). The Python package is aimed at more experienced users; most users are expected to use Enlighten2 through the PyMOL plugin. The code for the plugin is available as a separate GitHub repository (<https://github.com/vanderkamp/enlighten2-pymol>).

6 Conclusions

Enlighten2 provides non-expert users easy access to efficient, automated MD simulations of protein-ligand complexes, through a familiar interface and without assuming knowledge of setting up and running biomolecular simulations. The usage of Docker guarantees that the code runs and behaves identically on any machine and, apart from PyMOL and Docker, no software installation is required. It is therefore easy to use, also for classroom teaching or online workshops. In addition, by using Enlighten2 from the command-line or as a Python package, one can systematically run MD simulations for a set of ligands or enzyme variants as part of *in silico* workflows.

Acknowledgements

M.W.v.d.K. thanks Simon Bennie, Saad Raza, Kara Ranaghan and Adrian Mulholland for their contributions to the initial Enlighten tools, plugin and tutorials (see <https://github.com/marcvanderkamp/enlighten>).

Funding

This work has been supported by the by the Biotechnology and Biological Sciences Research Council (BB/L018756/1 and BB/M026280/1) and the UK Catalysis Hub (EPSRC grant EP/M013219/1).

Conflict of Interest: none declared.

References

- Byrne, M.J. et al. (2016) The Catalytic Mechanism of a Natural Diels–Alderase Revealed in Molecular Detail. *J. Am. Chem. Soc.*, **138**, 6095–6098.
- Case, D.A. et al. (2018) AMBER 2018, University of California, San Francisco.
- Drulyte, I. et al. (2019) Crystal structure of the putative cyclase IdmH from the indanomycin nonribosomal peptide synthase/polyketide synthase. *IUCr*, **6**, 1120–1133.
- Huggins, D.J. et al. (2019) Biomolecular simulations: From dynamics and mechanisms to computational assays of biological activity. *Wires. Comput. Mol. Sci.*, **9**, e1393.
- Land, H. and Humble, M.S. (2018) YASARA: A Tool to Obtain Structural Guidance in Biocatalytic Investigations. In: Bornscheuer, U.T. and Höhne, M., editors, *Protein Engineering: Methods and Protocols*, Springer New York, 43–67.
- Pettersen, E.F. et al. (2004) UCSF Chimera—a Visualization System for Exploratory Research and Analysis. *J. Comput. Chem.*, **25**, 1605–12.
- Ribeiro, J.V. et al. (2016) QwikMD—Integrative Molecular Dynamics Toolkit for Novices and Experts. *Sci. Rep.*, **6**, 26536.
- Rice, K. et al. (2019) The predominance of nucleotidyl activation in bacterial phosphonate biosynthesis. *Nat. Comm.*, **10**, 3698.
- Søndergaard, C.R. et al. (2011) Improved Treatment of Ligands and Coupling Effects in Empirical Calculation and Rationalization of pKa Values. *J. Chem. Theory Comput.*, **7**, 2284–2295.
- Wang, J. et al. (2006) Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graph. Model.*, **25**, 247–260.
- Wang, J. et al. (2004) Development and testing of a general amber force field. *J. Comp. Chem.*, **25**, 1157–1174.
- Wang, L. et al. (2018) A Rieske oxygenase/epoxide hydrolase-catalysed reaction cascade creates oxygen heterocycles in mupirocin biosynthesis. *Nat. Catal.*, **1**, 968–976.
- Word, J.M. et al. (1999) Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation, *J. Mol. Biol.*, **285**, 1735–1747.