JOURNAL OF CHEMICAL INFORMATION AND MODELING

Application Note

pubs.acs.org/jcim

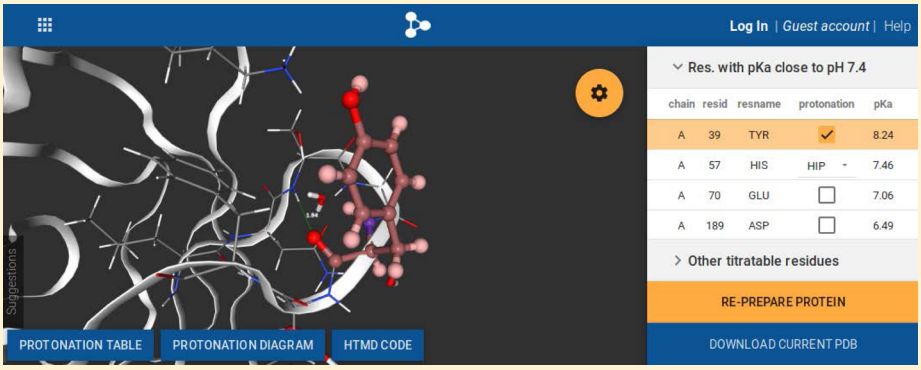
PlayMolecule ProteinPrepare: A Web Application for Protein Preparation for Molecular Dynamics Simulations

Gerard Martínez-Rosell,[†] Toni Giorgino,[*,‡] and Gianni De Fabritiis[*,§,†]

[†]Computational Biophysics Laboratory (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/Doctor Aiguader 88, Barcelona 08003, Spain

[‡]Institute of Neurosciences, National Research Council, I-35127 Padua, Italy

[§]Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, Barcelona 08010, Spain



ABSTRACT: Protein preparation is a critical step in molecular simulations that consists of refining a Protein Data Bank (PDB) structure by assigning titration states and optimizing the hydrogen-bonding network. In this application note, we describe ProteinPrepare, a web application designed to interactively support the preparation of protein structures. Users can upload a PDB file, choose the solvent pH value, and inspect the resulting protonated residues and hydrogen-bonding network within a 3D web interface. Protonation states are suggested automatically but can be manually changed using the visual aid of the hydrogen-bonding network. Tables and diagrams provide estimated pK_a values and charge states, with visual indication for cases where review is required. We expect the graphical interface to be a useful instrument to assess the validity of the preparation, but nevertheless, a script to execute the preparation offline with the High-Throughput Molecular Dynamics (HTMD) environment is also provided for noninteractive operations.

INTRODUCTION

The increase in sampling capabilities of modern molecular dynamics (MD) simulations has opened the way to a corresponding increase in the number of systems that research and industrial groups can study computationally.[1−5] The term *high-throughput*, in particular, has become applicable to large-scale MD simulations, indicating the possibility to build, simulate, and analyze multiple replicas of the same system or multiple systems (e.g., series of drug candidates) in an automated fashion.[6]

One step of the simulation workflow, however, has remained relatively underaddressed, namely, the preparation steps preliminary to the building of a molecular system. Proper accounting of titration states is important in protein−protein and protein−ligand recognition, enzymatic catalysis, and folding.[7] While there is software that performs this preparation task, generally from the command line, the evaluation of chemical groups has a heuristic nature that relies on visual intuition of three-dimensional (3D) structures and iteration.

For this reason, we devised ProteinPrepare, a protein preparation web application that allows the estimation and visualization of charge states, optimization of the hydrogen-bonding network of protonated structures, thorough visual inspection of the results, and rapid iteration of changes. The application provides an evaluation of the titration states of a target protein's residues on the basis of their local environment and the optimization of its hydrogen-bonding network through the placement of missing hydrogen atoms and flipping of side chains. Special consideration is given to residues whose pK_a is close to the solvent pH because they are more prone to be misclassified by the customary binary (protonated vs unprotonated) assignments; the user can interact with the results, force their chosen titration states, and have the application reoptimize the structure.

The computation is executed on the server taking advantage of High-Throughput Molecular Dynamics (HTMD),[6] a Python framework for simple molecular-simulation-based discovery. In particular, we used the `proteinPrepare()` functionality of HTMD, currently based on PROPKA 3.1 and PDB2PQR 2.1.[8,9] As

Received: April 3, 2017
Published: June 8, 2017

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516

such, in contrast to most other graphical tools, this web application also provides the short HTMD Python code required to perform the same task offline for the specific structure. The web application is publicly available at www.playmolecule.org for use on the web as part of the PlayMolecule web platform.

A number of other software packages and websites address parts of the preparation procedures of ProteinPrepare. Of note, the H++ server[10] includes the REDUCE software for hydrogen-bond optimization,[11] which is similar to the PDB2PQR optimization step in ProteinPrepare. An analogous feature is implemented in WHAT-IF.[12] The CHARMMing server computes PROPKA-based titration states and generates CHARMM topologies, without the hydrogen-bond optimization step.[13] PROPKA is also available as an interactive plugin for the Visual Molecular Dynamics (VMD) software.[14,15] The MDWeb server provides a Java-based interactive environment to inspect and manipulate hydrogen placement, but it does not take into account titration states;[16] similarly, the CHARMM-GUI server provides an input generator facility for several molecular formats, allowing the manual selection of protonation states.[17] Several packages and web interfaces compute pK_a values on the basis of continuum electrostatics calculations[18−20] (the underlying computation provided by, e.g., DelPhi,[21] APBS,[22] and MCCE[23]). Such treatment is more accurate, but it is also more computationally intensive and therefore has a lower feedback speed. Finally, some commercial software packages for drug discovery embed similar functions, e.g., Biovia Discovery Studio, which provides CHARMM-based pK_a calculations, and Schrödinger Maestro, whose interactive Protein Preparation Wizard, like ProteinPrepare, uses PROPKA 3.1 for pK_a calculations.

FEATURES

Input Selection. The ProteinPrepare web application is designed to provide a streamlined process (Figure 1) starting from a molecular structure provided as a PDB file, which can be retrieved from the RCSB Protein Data Bank (PDB)[24] or uploaded by the user. After its selection, the number of polypeptide chains present in the structure is counted, and an option is provided to select one of them or all of them together.

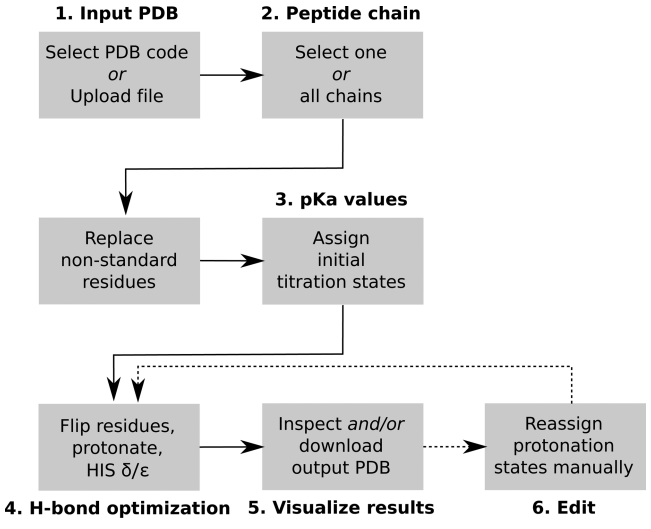


Figure 1. Sequence diagram of ProteinPrepare usage.

A checkbox also provides the option to remove crystallographic waters.

Processed structures are usually intended for later system building through force fields, such as AMBER[25] or CHARMM.[26] Such classical force fields support a predefined set of preparametrized residues, which generally encompasses the 20 standard amino acids and most of their titration states, but do not support arbitrary peptide units, e.g., engineered residues. For this reason, if nonstandard amino acids are detected in the structure, an option is provided to either (a) mutate them to any standard residue; (b) delete the residue, possibly leaving a gap in the peptide chain; or (c) leave the residue unchanged, in which case processing can continue but the residue will not be titrated nor optimized. The *Submit* button concludes the structure selection and starts the analysis.

Charge State Assignment. The first part of the analysis consists of the computation of pK_a values for all of the titratable side chains. This computation is performed by the PROPKA software, version 3.1.[8] The approach of PROPKA is to start with "model" pK_a values of ionizable residues, which are then shifted to account for their local environment: in particular, it includes empirical terms with contributions by (a) solvent exposure (buried ratio or desolvation penalty), (b) hydrogen bonds formed in the main chain and side chains, and (c) interactions with charged groups. Each of these terms shifts the pK_a values with respect to the model ones.[14,27] Different approaches exist for the determination of pK_a values[28] and have been the subject of recent blind prediction challenges,[29,30] whose comparison is outside of the scope of this paper. The current version of ProteinPrepare adopts PROPKA because it is at the same time fairly accurate[31] and virtually instantaneous (a requirement for interactive use).

The computed pK_a values are compared with the chosen pH value, and a heuristic is applied as follows: the residue is protonated if pH < pK_a and deprotonated otherwise. However, if |pH − pK_a| < 1, the residue is also marked as *dubious*. This is related to the connection between pK_a and the equilibrium probability of protonation:

$$p = \frac{1}{1 + 10^{\mathrm{p}K_a - \mathrm{pH}}} \quad (1)$$

In other words, we conventionally assume that residues whose charge states are assigned with probabilities p between approximately 0.1 and 0.9 (i.e., at pK_a ≈ pH ± 1) are not sufficiently certain and thus are marked for further inspection.

Optimization. Once charge states are assigned for titratable residues, optimization of the geometry of side chains occurs. This step is necessary because structures resolved by crystallographic methods often do not have sufficient resolution to distinguish, e.g., nitrogen from oxygen or carbon atoms in amide side chains, implying that the orientation of side chains in the model may not reflect a stable hydrogen-bonding network required to keep the protein folded.[12]

The hydrogen-bonding optimization is performed by the PDB2PQR software, version 2.1.[9] Here we only provide a summary of the algorithm, as the details are available from the relevant literature.[32] First, PDB2PQR fills in any missing heavy atoms. It then adds hydrogen atoms according to the titration states computed at the previous step, debumping residues in the case of steric clashes. The orientations of the side chains of the completed residues are then combinatorially optimized via a Monte Carlo procedure: groups of interacting residues are identified, and combinations of orientations (possibly flipping

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516

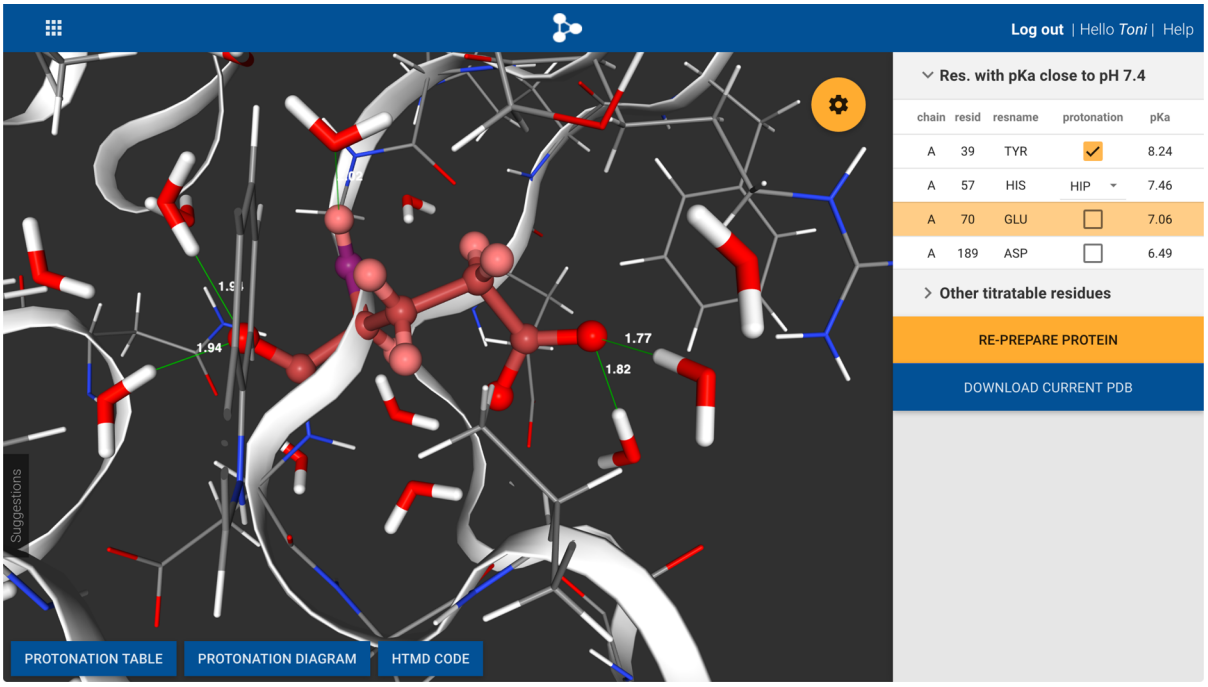


Figure 2. Main screen with the results of the preparation. The molecular structure of bovine pancreatic trypsin (PDB identifier 3PTB) at pH 7.4 is shown. Residue Glu 70 of chain A (yellow background in the right pane; orange balls-and-sticks side chain in the 3D view) and its neighbors (thin side chains in the 3D view) are highlighted. The right-hand side of the window enables manual protonation adjustments. Residues considered "dubious" (pH − 1 < pK_a < pH + 1) are shown by default, while the other titratable residues are accessible on the *Other titratable residues* drop-down list. Clicking on a residue highlights it in the 3D view; its protonation can be assigned by modifying the input checkbox or, in the case of histidines, by selecting the δ- (HID), ϵ- (HIE), or doubly protonated (HIP) state from the drop-down menu. The *Re-prepare protein button* repeats the hydrogen-bond optimization steps with the manually selected states; *Download current PDB* retrieves the current structure.

Gln, Asn, and His residues) and placements of hydrogens on the carboxy groups (Glu and Asp) and on nitrogens in neutral His (N_δ vs N_ϵ) are evaluated in search of the configuration providing the highest number of hydrogen bonds. If the location of crystallographic waters is known (and the user has not elected to remove them), protein−water and water−water interactions are finally considered, polarizing the water molecules accordingly.

ProteinPrepare's *main window* is finally opened to display the optimized configuration, which can be inspected and manipulated. The detailed list of modifications performed, such as the list of atoms added at each residue, is available in the *protonation table*, which is accessible at the bottom of the screen and described in a later section.

Main Window. The main window (Figure 2) contains a WebGL protein structure viewer,[33] a site navigation sidebar on the left, the main option menu on the right, and a number of tabs at the bottom containing extra information about the system. Since solvent exposure is one of the residue characteristics affecting pK_a (e.g., it is explicitly included in PROPKA's prediction model), a button is provided to display the solvent-inaccessible molecular surface (shown as a transparent shape superimposed on the current molecular structure view).

The right-hand side of the main window lists the titratable residues, starting with the ones whose protonation states are more likely to require manual review. The residue list provides pK_a values as well as controls to override any titration state chosen by the program. In particular, histidines can be forced in the δ-, ϵ-, or positively charged state. Clicking on one of the listed residues also highlights it in the 3D view (with a red outline), together with its neighboring residues (shown as thin lines) and the hydrogen-bonding network identified around it (shown as green lines).

After setting residues to the desired states, users can press the *Re-prepare protein* button to apply the changes. The optimization step is then re-executed respecting the assigned states.

Protonation Table. The *Protonation table* tab, located below the main window, opens a table summarizing data on all of the prepared molecules's residues. The table contains some of the data computed internally by PROPKA and PDB2PQR (Table 1). It lists all of the residues and provides additional information with respect to the one shown in the main window, such as the `buried` fraction and whether the last optimization `flipped` a residue's side chain as part of the optimization process. The *Download* button allows the user to

Table 1. Information Provided by the Protonation Table (Here "Titratable" Indicates Residues Asp, Glu, Cys, Tyr, His, Lys, and Arg

column	description	residues	source
`resname`	residue name	all	PDB
`resid`	residue ID	all	PDB
`insertion`	insertion code	all	PDB
`chain`	peptide chain	all	PDB
`buried`	fraction of residue buried in the core	all	PROPKA
pK_a	pK_a value	titratable	PROPKA
`protonation`	assigned protonation state	titratable	PDB2PQR
`flipped`	whether the side chain was flipped	Asn, Gln, His	PDB2PQR

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516

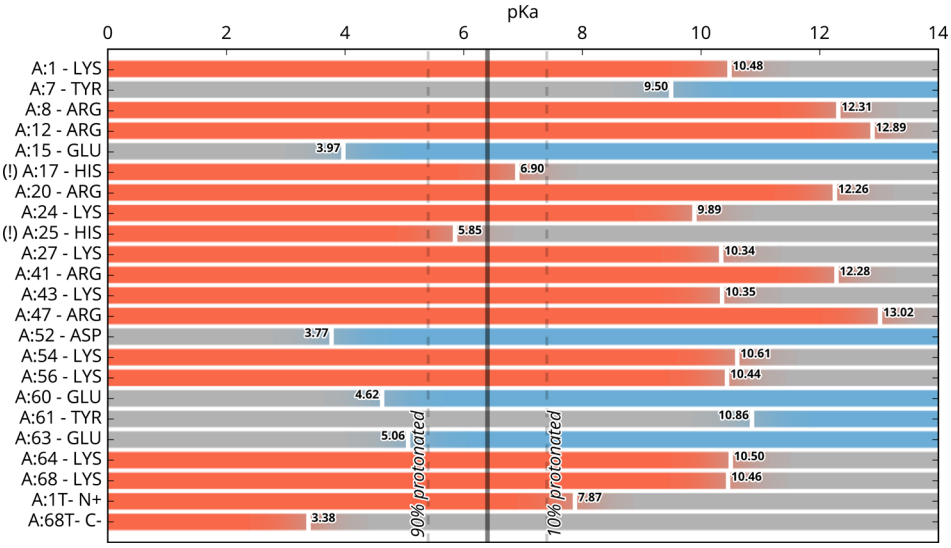
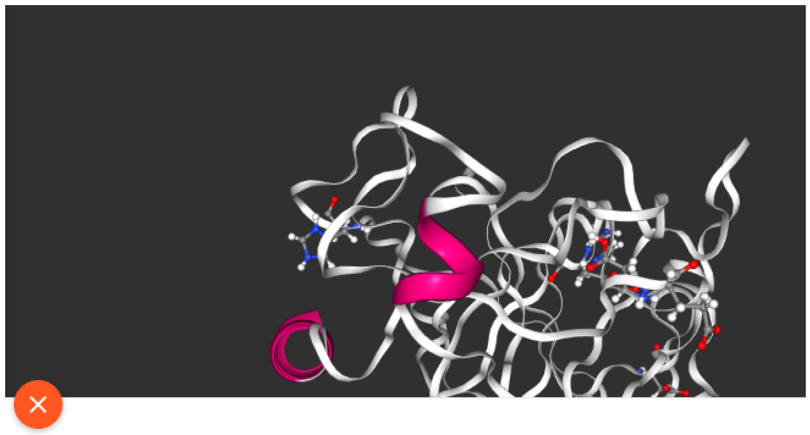


Figure 3. Protonation diagram of titratable residues in the structure of SDF-1/CXCL12 (PDB identifier 2KEE). The vertical gray lines show the currently chosen pH = 6.4 ± 1.0. Colors indicate residues' predicted states, namely, red (positively charged), gray (neutral), or blue (negative).



```
from htmd import *
mol = Molecule('3ptb')
mol.filter('protein')
molPrep, prepData = proteinPrepare(mol, pH=5.5, returnDetails=True)
d = prepData.data
d.loc[(d.resid == 91) & (d.chain == 'A'), "forced_protonation"] = 'HIP'
molPrep_reprepare, prepData_reprepare = prepData.reprepare()
```

Figure 4. The *HTMD code* pane shows the Python code reproducing the current state of the optimization. The system being processed here is bovine pancreatic trypsin (PDB identifier 3PTB) at pH 5.5, as can be seen in the HTMD function calls. Furthermore, His 91 has been manually set to the positively charged state.

download an even more detailed version of the table in Excel format, including, e.g., the list of atoms missing from the input file (possibly not resolved) and consequently introduced at guessed positions.

Protonation Diagram. The *Protonation diagram* tab contains a diagram showing the sensitivity of the charge states of the protein's titratable residues to the buffer acidity (Figure 3). The plot provides an at-a-glance summary of how close the residues' predicted pK_a values are to the currently selected pH on the basis of eq 1. Each titratable residue is shown as a horizontal bar, its color matching the predicted titration state at varying pH values (on the horizontal axis). For acidic residues, the color of the bar blends between gray (neutral) and blue (negatively charged); for basic residues, it blends between red (positively charged) and gray (neutral). The actual pK_a value is shown as a tick and a label close by.

A vertical reference band is placed at the currently selected pH value plus or minus one unit. Thus, residues whose pK_a values fall outside the reference band can be considered relatively unambiguous, while those within the band are the

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516

ones for which manual review is recommended as discussed above because their predicted protonation probabilities fall between approximately 10% and 90% (assuming correct pK_a predictions).

HTMD Code. An important feature of ProteinPrepare is that it overcomes the "interactive versus scriptable" dilemma by providing ready-to-use code that reproduces the results currently shown in the interface. The *HTMD code* panel (Figure 4) contains a short script that can be executed in a Python 3 environment with the HTMD package installed[6] (freely available at www.htmd.org). The script reproduces the steps manually performed during the current session, namely, loading the PDB from the web, applying mutations of nonstandard residues, protonation overrides, and optimization.

Implementation. The ProteinPrepare web service relies on the HTMD package for processing molecular structures.[6] In particular, it provides classes and functions to edit molecular structures, query residue information, and programmatically perform the preparation steps (through its `proteinPrepare()` method).

The in-browser 3D interactive molecular representation is provided by the NGL viewer.[33] The viewer provides methods to define various types of representations; these are used by the ProteinPrepare service to highlight the currently selected residue (the "inspect" function) and its neighbors, with visually distinct styles (Figure 2). The view can be manipulated in three dimensions, and residue and atom identities can be read out with the mouse pointer from there. The viewer is based on the WebGL standard and thus works on all recent browsers (including mobile devices) without plug-ins.

It is possible to use the interface either as an anonymous *Guest user* or after authenticating oneself, in which case the jobs submitted will be kept private. Authentication is based on Google's OAuth standard and thus does not require a registration (i.e., users can use Google's identity).

The server was implemented using Flask microframework, an open-source framework supporting the development of Python-based web applications. This framework, together with extensions such as Flask-SQLAlchemy, provides an object-relational mapper that exposes database entities and relationships as Python objects as well as simple methods that enable the communication between the server and the client via JSON-formatted requests/responses.[34]

The front end was implemented using AngularJS 1.5, an open-source web application framework focused on the development of reactive single-page web applications. Furthermore, Angular Material 1.1 provided a set of reusable web components implementing the latest Material Design guidelines, a set of specifications that yield a consistent and intuitive visual language.

CONCLUSIONS

ProteinPrepare, together with the PlayMolecule platform, is a web application that fully leverages modern web technologies such as WebGL and the Material Design guidelines. Hence, it enables a rich and dynamic interaction, once only attainable by desktop applications, with the advantages of web technologies such as platform independence, being installation- and update-free, and so on. These advantages, together with steady improvements in web technologies and connectivity, suggest that rich web applications may become pivotal in academia and the pharmaceutical industry alike.

A limitation of the methods used in ProteinPrepare is that they rely on a static configuration for assigning titration states. Large structural rearrangements, for example, will change residues' solvent exposure, and the pK_a values will be shifted accordingly. However, this is a standard assumption to make until widespread use of constant-pH MD may address this problem in the future.[35,36] Furthermore, the software underlying ProteinPrepare does not support arbitrary peptides. Users are required to indicate replacements for any nonstandard residues encountered. Similarly, ligands and ions are not currently taken into account for optimizing the structure, although these limitations may be removed in forthcoming versions.

In our view, the protein preparation tool presented in this application note offers three main features: (1) help in deciding the protonation states and charges of residues while still retaining the capability to change them, (2) optimization of the hydrogen-bonding network as it might have been approximative from the crystal structure, and (3) both an interactive and scriptable way to perform these tasks. Deciding on protonation states in classical molecular dynamics is critical because a change in charge can have drastic effects and invalidate all of the results. The hydrogen-bonding network optimization is important to help the protein retain the original state of the crystal structure, especially when losing an active or inactive state can take milliseconds or more to recover. The third aspect is purely one of convenience: the use of the HTMD framework allows the reproduction of any result obtained through the ProteinPrepare web interface on local computing resources and the possibility to automate the preparation steps, e.g., repeating them for a large set of structures in a *high-throughput* context. Coding of HTMD-based analysis on local resources takes place through the Python language and can thus take advantage of any of its numerous libraries and facilities for reproducible research.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: toni.giorgino@cnr.it.

*E-mail: gianni.defabritiis@upf.edu.

ORCID

Gerard Martínez-Rosell: 0000-0001-6277-6769

Toni Giorgino: 0000-0001-6449-0596

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Acellera Ltd. for funding. G.D.F. aknowledges support from MINECO (BIO2014-53095-P) and FEDER. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 675451 (CompBioMed project).

REFERENCES

(1) Harvey, M. J.; Giupponi, G.; De Fabritiis, G. *J. Chem. Theory Comput.* **2009**, *5*, 1632−1639.

(2) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. *J. Comput. Chem.* **2005**, *26*, 1668−1688.

(3) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E. *Bioinformatics* **2013**, *29*, 845−854.

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516

(4) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossvary, I.; Moraes, M. A.; Sacerdoti, F. D.; Salmon, J. K.; Shan, Y.; Shaw, D. E. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*; ACM: New York, 2006.
(5) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. *J. Comput. Chem.* **2005**, *26*, 1781−1802.
(6) Doerr, S.; Harvey, M. J.; Noé, F.; De Fabritiis, G. *J. Chem. Theory Comput.* **2016**, *12*, 1845−1852.
(7) Sadqi, M.; Lapidus, L. J.; Muñoz, V. *Proc. Natl. Acad. Sci. U. S. A.* **2003**, *100*, 12117−12122.
(8) Søndergaard, C. R.; Olsson, M. H. M.; Rostkowski, M.; Jensen, J. H. *J. Chem. Theory Comput.* **2011**, *7*, 2284−2295.
(9) Dolinsky, T. J.; Czodrowski, P.; Li, H.; Nielsen, J. E.; Jensen, J. H.; Klebe, G.; Baker, N. A. *Nucleic Acids Res.* **2007**, *35*, W522−W525.
(10) Anandakrishnan, R.; Aguilar, B.; Onufriev, A. V. *Nucleic Acids Res.* **2012**, *40*, W537−W541.
(11) Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. *J. Mol. Biol.* **1999**, *285*, 1735−1747.
(12) Hooft, R. W.; Sander, C.; Vriend, G. *Proteins: Struct., Funct., Genet.* **1996**, *26*, 363−376.
(13) Miller, B. T.; Singh, R. P.; Klauda, J. B.; Hodošček, M.; Brooks, B. R.; Woodcock, H. L. *J. Chem. Inf. Model.* **2008**, *48*, 1920−1929.
(14) Rostkowski, M.; Olsson, M. H.; Søndergaard, C. R.; Jensen, J. H. *BMC Struct. Biol.* **2011**, *11*, 6.
(15) Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graphics* **1996**, *14*, 33−38.
(16) Hospital, A.; Andrio, P.; Fenollosa, C.; Cicin-Sain, D.; Orozco, M.; Gelpí, J. L. *Bioinformatics* **2012**, *28*, 1278−1279.
(17) Lee, J.; et al. *J. Chem. Theory Comput.* **2016**, *12*, 405−413.
(18) Nielsen, J. E. *J. Mol. Graphics Modell.* **2007**, *25*, 691−699.
(19) Kieseritzky, G.; Knapp, E.-W. *Proteins: Struct., Funct., Genet.* **2008**, *71*, 1335−1348.
(20) Tan, K. P.; Nguyen, T. B.; Patel, S.; Varadarajan, R.; Madhusudhan, M. S. *Nucleic Acids Res.* **2013**, *41*, W314−W321.
(21) Li, L.; Li, C.; Sarkar, S.; Zhang, J.; Witham, S.; Zhang, Z.; Wang, L.; Smith, N.; Petukh, M.; Alexov, E. *BMC Biophys.* **2012**, *5*, 9.
(22) Baker, N. A.; Sept, D.; Joseph, S.; Holst, M. J.; McCammon, J. A. *Proc. Natl. Acad. Sci. U. S. A.* **2001**, *98*, 10037−10041.
(23) Georgescu, R. E.; Alexov, E. G.; Gunner, M. R. *Biophys. J.* **2002**, *83*, 1731−1748.
(24) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(25) Ponder, J. W.; Case, D. A. *Adv. Protein Chem.* **2003**, *66*, 27−85.
(26) MacKerell, A. D.; et al. *J. Phys. Chem. B* **1998**, *102*, 3586−3616.
(27) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M.; Jensen, J. H. *J. Chem. Theory Comput.* **2011**, *7*, 525−537.
(28) Gunner, M. R.; Baker, N. A. *Methods Enzymol.* **2016**, *578*, 1−20.
(29) Nielsen, J. E.; Gunner, M. R.; García-Moreno E., B. *Proteins: Struct., Funct., Bioinf.* **2011**, *79*, 3249−3259.
(30) Gosink, L. J.; Hogan, E. A.; Pulsipher, T. C.; Baker, N. A. *Proteins: Struct., Funct., Genet.* **2014**, *82*, 354−363.
(31) Davies, M. N.; Toseland, C. P.; Moss, D. S.; Flower, D. R. *BMC Biochem.* **2006**, *7*, 18.
(32) Dolinsky, T. J.; Nielsen, J. E.; McCammon, J. A.; Baker, N. A. *Nucleic Acids Res.* **2004**, *32*, W665−W667.
(33) Rose, A. S.; Hildebrand, P. W. *Nucleic Acids Res.* **2015**, *43*, W576−W579.
(34) Grinberg, M. *Flask Web Development: Developing Web Applications with Python*, 1st ed.; O'Reilly Media: Sebastopol, CA, 2014.
(35) Donnini, S.; Tegeler, F.; Groenhof, G.; Grubmüller, H. *J. Chem. Theory Comput.* **2011**, *7*, 1962−1978.
(36) Mongan, J.; Case, D. A.; McCammon, J. A. *J. Comput. Chem.* **2004**, *25*, 2038−2048.

DOI: 10.1021/acs.jcim.7b00190
J. Chem. Inf. Model. 2017, 57, 1511−1516