BiKi Life Sciences

Decherchi, Sergio; Bottegoni, Giovanni; Spitaleri, Andrea; Rocchia, Walter; Cavalli, Andrea

DOI:
10.1021/acs.jcim.7b00680

License:
Other (please specify with Rights Statement)

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Decherchi, S, Bottegoni, G, Spitaleri, A, Rocchia, W & Cavalli, A 2018, 'BiKi Life Sciences: A New Suite for Molecular Dynamics and Related Methods in Drug Discovery', *Journal of Chemical Information and Modeling*, vol. 58, no. 2, pp. 219-224. https://doi.org/10.1021/acs.jcim.7b00680

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Application Note

Cite This: J. Chem. Inf. Model. 2018, 58, 219−224

pubs.acs.org/jcim

BiKi Life Sciences: A New Suite for Molecular Dynamics and Related Methods in Drug Discovery

Sergio Decherchi,*[,†,‡] Giovanni Bottegoni,[§] Andrea Spitaleri,[∥,‡] Walter Rocchia,*[,∥] and Andrea Cavalli*[,†,⊥]

[†]Computational Sciences, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

[‡]BiKi Technologies s.r.l, Via XX Settembre, n. 33/10, 16121 Genova, Italy

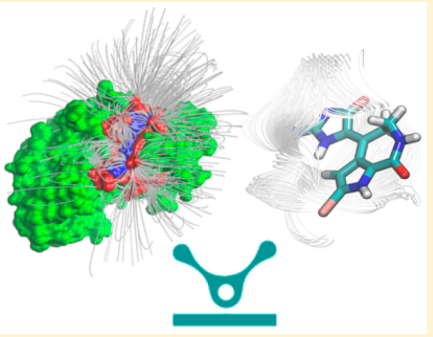
[§]Heptares Therapeutics Ltd, BioPark, Broadwater Road, AL73AX, Welwyn Garden City, U.K.

[∥]Concept Lab, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

[⊥]Department of Pharmacy and Biotechnology, University of Bologna, Via Belmeloro 6, 40126 Bologna, Italy

Supporting Information

ABSTRACT: In this paper, we introduce the BiKi Life Sciences suite. This software makes it easy for computational medicinal chemists to run ad hoc molecular dynamics protocols in a novel and task-oriented environment; as a notebook, BiKi (acronym of Binding Kinetics) keeps memory of any activity together with dependencies among them. It offers unique accelerated protein−ligand binding/unbinding methods and other useful tools to gain actionable knowledge from molecular dynamics simulations and to simplify the drug discovery process.



INTRODUCTION

Computational chemistry has played an increasingly central role in drug discovery and pharmaceutical research, becoming an established tool in the search for novel bioactive compounds.[1] Compared to other computational methods, molecular dynamics (MD) has, until recently, played a marginal role in prospective drug discovery projects, especially those carried out in industry. Three elements have hampered the widespread exploitation of MD for pharmaceutical research. First, MD has traditionally been perceived as a simulative technique that is difficult to implement and automatize because it relies on a complex theoretical background. Second, the analysis of long MD trajectories can be a daunting task that ultimately fails to return clear and specific insights into how to design and synthesize new and more effective compounds. Finally, MD is regarded as very time-consuming and therefore incompatible with the stringent time requirements of most drug discovery projects.

Recently, thanks to the advent of novel hardware architectures, including graphical processor units (GPUs) and purposely designed machines,[2] and the development of innovative algorithms,[3] it is becoming possible to compute MD trajectories of pharmaceutically relevant systems on a time scale approaching that of biological events. Some published studies have used plain MD simulations, encompassing several microsecond-long trajectories, to investigate the spontaneous binding of ligands to their biological counterparts.[4−6] However, such studies are still very computationally demanding, even for one single compound, which drastically limits their applicability. Furthermore, some events, including the spontaneous unbinding of molecules with long residence times, are still beyond the reach of plain MD simulations even in combination with purpose-built hardware architectures. To overcome these limitations, several physical approaches aimed at enhancing the sampling of rare events have been developed over the last 20 years.[1] These advanced sampling schemes[7] may have addressed the issue of speed. However, the applicability of MD and related methods is still limited by difficulties in simulation setup and analysis, particularly when dealing with accelerated sampling prospectively applied to drug discovery projects. We recently developed several specific dedicated algorithms for fully flexible dynamic docking[3] and undocking.[8,9] These algorithms require a limited number of free parameters, thus transforming valuable but previously impractical algorithms into a new technology, which can be exploited by a wider community of users in industry and academia. Here, we report on a new software suite, named BiKi Life Sciences: this suite includes the above algorithms via a set of modules in a coherent graphical and scripting environment aimed at facilitating the use of MD in drug discovery with particular focus to the hit-to-lead and lead optimization phases.

Received: November 28, 2017
Published: January 17, 2018

DOI: 10.1021/acs.jcim.7b00680
J. Chem. Inf. Model. 2018, 58, 219−224

BIKI LIFE SCIENCES

BiKi Life Sciences (hereafter referred to as BiKi, acronym of Binding Kinetics) is an integrated software environment, encompassing multiple modules:

- BiKi Basic, a core module dedicated to setting up classic MD simulations and related postprocessing and analysis methods;
- BiKi Netics to study unbinding kinetics via scaled MD simulations;[8]
- Pocketron for a dynamical analysis of MD trajectories in order to characterize pockets and potential binding sites and identify the allosteric connections between them;[10]
- BiKi MD-Binding for a fully solvated and dynamic docking based on a purpose-built bias;[3]
- BiKi Hydra to assess the relevance of water molecules within binding pockets based on their stability.[11]

The functions of each module can be accessed through a graphical user interface (GUI) or through the BiKi scripting engine. Figure 1 reports a schematic summary of the BiKi modules and their interactions. The BiKi GUI is written in Java and is built around the concept of "project". Each project has a one-to-one correspondence with a folder on disk and comprises a set of simulation and analysis items, together with input/output files. The BiKi scripting engine is an API coded in Java. It allows the functionalities offered by the GUI to be replicated in more detail. In particular, this API allows scripting in Python, through the system interpreter, or leveraging the internal Groovy interpreter. Both the GUI and the scripting engine are interfaced with high-performance parallel codes (C, C++, Fortran), partly developed by third parties such as plumed.[12]

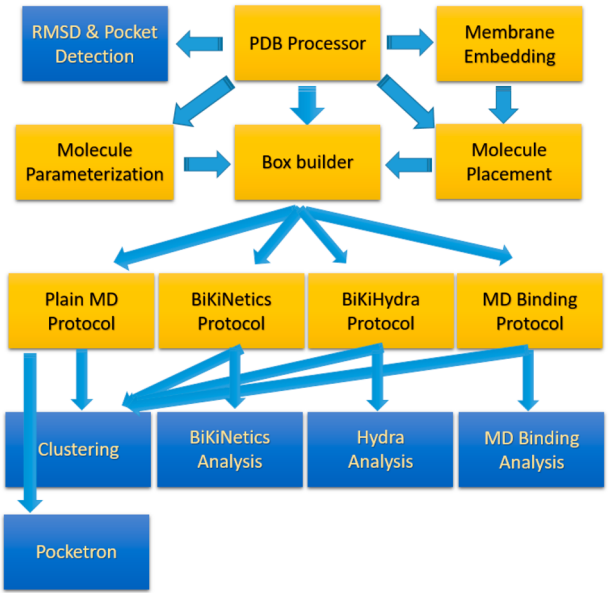


Figure 1. BiKi modules and their interactions: (yellow) builder modules, (blue) analysis modules.

A common aspect to all modules is that, for any selectable item, BiKi automatically provides default values representative of the corresponding typical simulative MD scenario. Often, these parameters reflect a trade-off between accuracy and execution speed. To make it more user-friendly for beginners, only a limited number of tunable parameters are usually displayed, while the rest can be found under the "advanced" tab.

Each project comprises different simulations. In turn, each simulation comprises multiple simulation items. Items are "stateful" objects that are stored inside a project. BiKi has also some utilities for one-shot activities not logically linked to a specific project. These are stateless objects that address specific tasks.

Items, identified by the .bkx extension, are XML files that store all the settings related to that specific aspect of the simulation. Examples of items include the following: structures files, periodic boxes, and input protocols. The name of every item can be customized by the user, allowing long-term traceability of any action.

A graphical project tree displays a project. All items pertinent to a project are located under the "Setting" leaf. Two more leaves are dedicated to input and output files (the "Input" and "Output" leaves, respectively). Objects can depend on each other through files, and in MD campaigns, structures are often manipulated in a complex flow. Therefore, BiKi has a built-in dependency-resolution system. The dependency between files coherently takes into account file changes as in a tool like "make". If an item/file depends on a file/item that has changed, then the item must be updated according to the contents of the modified file to restore consistency. This dependency mechanism is transparent to the user.

BIKI BASICS

BiKi Basic can be used to perform all the relevant operations related to setting up, running, and analyzing plain MD trajectories. It is also the backbone on which the other modules have been developed.

For the MD simulation setup, BiKi Basic provides the following items/utilities:

- **Pdb processing**. PDB fetching, subselections with a VMD-like syntax,[13] residue manual protonation assignment, automated water-naming conversion, axis alignment, duplicated atoms removal, conversion to pqr/xyzr, chain termination, application of Amber naming conventions, and hydrogen-adding for proteins.
- **Molecule parameterization**. An essential component of the BiKi pipeline is the tool for parametrizing arbitrary molecules (e.g., ligands). For BiKi, the Amber force field is the reference force field and nomenclature. Therefore, it interfaces to AmberTools to place water molecules and to support charge computation.[14] Currently, BiKi supports ff99SBildn[15] and ff14SB.[16] Molecules can be parametrized by a specific module that uses the general Amber force field (GAFF)[17] and Antechamber.[18] Newly generated topologies can be stored in a local user library. Residues parametrized outside BiKi can be imported and added to the same library. RESP charge calculation is supported via a BiKi specific interface to the NwChem software.[19] Since NwChem is not currently compatible with Antechamber, BiKi provides a user-transparent conversion tool that converts the NwChem output into an Antechamber-compliant ESP file.
- **Protein membrane embedding**. The setup of a protein−membrane simulation involves two steps: (i) predicting the position/orientation of the protein within the bilayer and (ii) the actual plugging of the protein within the membrane lipids. BiKi's membrane setup tool is devised

DOI: 10.1021/acs.jcim.7b00680
J. Chem. Inf. Model. 2018, 58, 219−224

to automate the entire procedure. The posing algorithm works in a way similar to that defined in the work of Wolf et al.[20] A relevant exception is that our score is calculated according to a slightly modified approach. While BiKi still computes a "per slice" partial score based on the detection of a hydrophobic belt, we use an atomic-level description to define the score. We adopt an optimization procedure based on multiple cycles, again in line with the work of Wolf et al.,[20] where bisection steps are used to converge to a local maximum.

The second phase is the actual plugging of the protein into the membrane. The protein must be inserted while causing the least perturbation possible in the already equilibrated membrane template. To this end, in line with the work of Yesylevskyy,[21] BiKi generates a shrunk, nonphysical version of the protein around its main inertial axis. Using the values for shift and tilt obtained during the posing phase, the shrunk protein is lodged in the membrane, carving an initial hole in the bilayer by removing all the lipids and water molecules within 3 Å of the protein. The initial plugging can thus be carried out while minimizing the number of lipids removed. Then, several subsequent inflation steps gradually restore the protein to its original size. At each step, the protein structure is frozen while the protein−membrane interaction potential is minimized. The protein internal energy is ignored until the very last step, where the entire protein structure is harmonically restrained and relaxed. Alterations in local pressure can actually take place, but they can easily be fixed in the subsequent NPT anisotropic equilibration phase. BiKi ships with a set of pre-equilibrated membrane templates (users can store their own customized templates, such as those built via CharmmGUI,[22] to an internal membrane template database). Internal solvation is recovered via a purposely developed strategy: (i) we compute the solvent excluded surface (SES) of the protein via NanoShaper;[23,24] (ii) we solvate the protein structure; (iii) we discard all waters except those internal to the SES, which are added to the final protein−membrane system. Final poses were validated by comparison with structures in the OPM database[25] (see the SI).

- **Molecule placement**. When initializing molecular dynamics runs, it is often the case that some molecules must be placed at given positions, either in the solvent or in predetermined sites of a protein. This item allows, for instance, a ligand to be positioned in place of one without hydrogens, or randomly into the solvent. Another function, is the possibility to automatically place the ligand in front of the targeted binding pocket for the subsequent application of the MD binding protocol. See ref 3 for more details.
- **Simulation box setup**. Once the molecules have been parametrized if needed, the membrane has been included in the system, and the other molecules have been placed, the system topology can be assembled. At this stage, it is assumed that the structure is complete and there are no missing residues or atoms. The box/topology building tool supports Amber topology files. These can later be converted into Gromacs topologies, if needed. In this way, BiKi indirectly supports multiple MD engines including Amber,[14] Namd,[26] and Gromacs.[27]
- **Simulation Protocol**. Before performing the simulation, the final key step is to define a simulative protocol. In the current release, BiKi supports Gromacs version 4.6.1 and allows the minimization and equilibration protocol to be defined in full detail. Here too, a default equilibration protocol is provided.

  In the equilibration section, a stepwise protocol must be devised by setting key parameters, such as target temperature, thermodynamics ensemble, simulation time, and time step. After the equilibration steps have been set up, similar parameters must be assigned for the production section. Additionally, the production section has a "Run Count" field, where the number of replicas can be set for the production runs. This is useful when runs are carried out in parallel to accumulate statistics. A reset button allows the user to revert back to the default parameters. Once the protocol has been completely defined, the user can perform the final build: BiKi will create specific folders for minimization, for each equilibration step, and for the production. The user can choose to run the simulations locally from the GUI, to run them manually (in every step, a bash run.sh script file is generated), or to move the root folder to another machine. The time advancement or the status of the simulation can be queried directly from the GUI. Run.sh are bash scripts that, together with the simulation execution line, contain a line to remove PBC visual artifacts and allow an immediate visualization of the trajectory. By the scripting interface, it is also possible to wrap run.sh files into queue submission files for LSF, SLURM, PBS, and SGE queue systems. Finally, completed simulations can be rapidly restarted and extended via command line scripts.

The analysis aspect of BiKi Basic includes: clustering, RMSD computation, static pocket detection, and trajectory cleaning.

- **Clustering**. BiKi provides a multicore-CPU-optimized, RMSD-based, k-medoids clustering algorithm.[5] As input, the algorithm uses a set of trajectories, the corresponding atom selections, and the number of clusters (there is also a mode where the number of clusters can be obtained automatically). A clustering viewer that can display the clustering graph is associated with the clustering engine. The clustering engine saves medoids (centroids) as pdb files. These can readily be used in other tools. For example, we recently devised a pipeline comprising MD and clustering to enhance virtual screening results.[28] In this regard, BiKi output is completely compatible with the VS FLAP tool from Molecular Discovery.[29]
- **Static pocket detection.** Leveraging the interface to NanoShaper, BiKi can compute static pockets in protein structures. The static definition of pockets is based on the difference between the solvent excluded volumes, calculated via the Connolly−Richards definition, obtained with two different probe radii.[23] Useful information can be gathered from this process, including pocket volume, surface area, and a list of solvent-exposed atoms in each pocket.
- **RMSD calculator**. The suit includes a highly flexible RMSD calculator for static structures or along the MD trajectory. The tool is extremely flexible, in that the user can define up to four atom selections for a pair of structures: namely, one atom selection for the optimal

DOI: 10.1021/acs.jcim.7b00680
J. Chem. Inf. Model. 2018, 58, 219−224

alignment and one for RMSD computation per structure. In this way, even pdb files with different residue numbering can be seamlessly managed. To the best of our knowledge, this is the only available tool that provides the user with this flexibility.

Finally, BiKi implements scripting tools to customize all commands. Everything done in the GUI can replicated by scripting. BiKi ships several Python/Groovy scripts together with Javadoc documentation in case the user wants to write their own scripts. BiKi also allows an entire simulative campaign to be set up in one command line specifying: pdb preprocessing, box build-up, minimization/equilibration folders, and the production protocol.

Pocketron. BiKi offers a dynamic pocket detection algorithm, named Pocketron,[10] that can track all the pockets during an MD run. No a priori knowledge of the pocket locations is needed to run the tracker. Pocketron has an analysis counterpart that can represent the time-evolution variables, such as the volume/surface area over time, the most representative residues of each pocket, and an interaction network graph, providing semiquantitative information on the rate of dynamic interactions among the pockets. The information on pocket dynamics, interaction, and exchange of atoms over time can provide insights into the presence of cryptic and transient pockets and allosteric communications when analyzed as an ensemble.

Modules for Enhanced Sampling Simulative Protocols. BiKi implements three recently developed modules for enhanced sampling protocols, namely: BiKi Netics, an unbinding protocol to rank ligands based on their residence time;[8,9] BiKi MD-Binding, a tool to perform fully flexible protein ligand docking;[3] and BiKi Hydra, a tool to dynamically characterize water persistency and "happiness" in binding sites.[11]

BIKI NETICS

The BiKi Netics module aims to rank different ligands binding to the same target based on their residence time. It does this by simulating the different complexes in an MD setup where the potential energy is scaled by a factor λ. The relative stability of the simulated complexes is then observed and used as an indicator of the residence-time difference. This simple modification, coupled with proper restraining, makes it possible to study protein−ligand unbinding in a computationally effective albeit approximate way. Its use is facilitated by the GUI, where the user can seamlessly select the scaling value and the harmonic restraint. The latter is needed to avoid the breaking of structures that the scaling could cause. The GUI allows the user to run the simulations easily and to analyze them statistically, as performed in ref 8.

BIKI MD-BINDING

MD-Binding[3] is an algorithm aimed at accelerating the simulation of the protein−ligand binding process, in order to reach computational times that are compatible with an industrial pipeline. This protocol involves an adaptive electrostatics-driven steered MD simulation. MD-Binding uses an attractive electrostatic bias to induce the occurrence of concerted motions of the two binding partners and to accelerate the association process. In particular, it leverages a sophisticated adaptive steered MD protocol, which induces a gentle approaching of the ligand toward the site. In the GUI, the user can choose the attracting atoms on the ligand and protein side. Additionally, the user must select a "switch-off" set of atoms, typically one residue, to tell MD-Binding to turn off the bias when the ligand is in its proximity. This is done to focus the action of the bias on the climbing of the energy barrier related to bimolecular association, while gradually turning off the bias during the descent toward the destination energetic basin.

MD-Binding was validated against a diverse set of complexes of pharmaceutical interest, with promising results.[3]

Upon running of the simulations, an analysis tool allows some key observables to be checked, including the following: the distance from the switch-off residue; the values of the collective variable; the time varying kappa in the harmonic steered MD restraint; the RMSD (optionally) with respect to a given reference structure during the run; and the cumulative work.

MD-Binding setup can also be done via scripting. This requires the execution of just one Python line to specify the target protein, the ligand, the attracting residues, and the queue submission system. From a technical perspective, the MD-Binding setup via scripting is thus similar to that of a classical docking campaign.

BIKI HYDRA

BiKi Hydra is designed to give a quick estimate of the persistency of water molecules in a given region of a system, usually in and near the binding site. BiKi Hydra works by repelling water molecules in the desired region by applying an external multicentered repulsion with the form of a screened electrostatic interaction. The form of the bias is the same as that used in the MD-Binding approach, but with opposite forces and without time adaptivity. The ranking of water happiness is ruled by the time persistency of the water molecules upon repulsion. This technique was successfully applied to the A2A GPCR.[11] In rational drug design, identifying the relative propensities of water molecules to reside in a given region of a site can be instrumental in deciding how best to decorate a scaffold or optimize the thermodynamic or kinetic properties of a compound.

In setting up the protocol for this functionality, the user must define the classical plain MD parameters and the repulsion group, i.e., the site to be desolvated. This definition can be applied with the usual VMD selection syntax. If a ligand is present in the analyzed structure, the ligand is also included in the desolvation group. The analysis part involves a dedicated tool to compute the probability of finding water in a specific location in space. This is achieved by building a volumetric grid around the desolvated site, and counting the density along the trajectory of water molecules. A file in OpenDX format is saved and then visualized via VMD. Together with this file, the local maxima of water probability are also found and can easily be visualized in VMD.

CONCLUSIONS

The BiKi Life Sciences software suite was designed to achieve several goals: (i) simplify MD setup, (ii) promote simplified protocols with default values for industrial users using freely available MD codes, (iii) push MD as the practical everyday tool of choice for medicinal/computational chemists, and (iv) make automated dynamic docking/undocking[1] available for the first time at industry-compliant computational speeds. The tool

DOI: 10.1021/acs.jcim.7b00680
J. Chem. Inf. Model. 2018, 58, 219−224

was written in Java to confer maximum flexibility and to allow scripting by the native shipped interpreter in Groovy and in Python. BiKi can prepare topologies in Amber and Gromacs formats. As for Gromacs, it can prepare the full MD protocol in three steps (prepare the pdb file, prepare the box, create the simulation). Additionally, the installer allows BiKi to be used immediately without any manual intervention/compilation. In subsequent releases, we will improve the interface (particularly in terms of remote HPC integration), add new functionalities, improve interoperability among third-party tools, and include wizards, and an in-the-GUI scripts editor and runner to provide a completely guided software experience for the inexperienced user.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.7b00680.

Membrane module validation (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: sergio.decherchi@iit.it (S.D.).
*E-mail: walter.rocchia@iit.it (W.R.).
*E-mail: andrea.cavalli@iit.it (A.C.).

ORCID

Giovanni Bottegoni: 0000-0003-1251-583X
Andrea Spitaleri: 0000-0003-3012-3557
Walter Rocchia: 0000-0003-2480-7151

Author Contributions

S.D. conceived of and implemented the BiKi Life Sciences software suite. A.S., G.B., W.R., and A.C. tested the software and contributed to the design. A.C. conceived the BiKi project. The manuscript was written with contributions from all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare the following competing financial interest(s): The authors fully own BiKi Technologies s.r.l. the company that sells the BiKi Life Sciences software suite.

■ ACKNOWLEDGMENTS

We thank IIT for the financial support, Mattia Pini and Marco Perrando for help in the development, and Francesca Spyrakis, Luca Mollica, and Giuseppina LaSala for testing early versions of BiKi and for valuable suggestions.

■ REFERENCES

(1) De Vivo, M.; Masetti, M.; Bottegoni, G.; Cavalli, A. Role of Molecular Dynamics and Related Methods in Drug Discovery. *J. Med. Chem.* **2016**, *59*, 4035−4061.

(2) Shaw, D. E.; Deneroff, M. M.; Dror, R. O.; Kuskin, J. S.; Larson, R. H.; Salmon, J. K.; Young, C.; Batson, B.; Bowers, K. J.; Chao, J. C.; Eastwood, M. P.; Gagliardo, J.; Grossman, J. P.; Ho, C. R.; Lerardi, D. J.; Kolossváry, I.; Klepeis, J. L.; Layman, T.; McLeavey, C.; Moraes, M. A.; Mueller, R.; Priest, E. C.; Shan, Y.; Spengler, J.; Theobald, M.; Towles, B.; Wang, S. C. Anton, A Special-Purpose Machine for Molecular Dynamics Simulation. *Commun. ACM* **2008**, *51*, 91−97.

(3) Spitaleri, A.; Decherchi, S.; Cavalli, A.; Rocchia, W. Fast Dynamic Docking Guided by Electrostatic Bias: the MD-Binding Approach. *J. Chem. Theory Comput.* **2018**, DOI: 10.1021/acs.jctc.7b01088.

(4) Shan, Y.; Kim, E. T.; Eastwood, M. P.; Dror, R. O.; Seeliger, M. A.; Shaw, D. E. How Does a Drug Molecule Find its Target Binding Site? *J. Am. Chem. Soc.* **2011**, *133*, 9181−9183.

(5) Decherchi, S.; Berteotti, A.; Bottegoni, G.; Rocchia, W.; Cavalli, A. The Ligand Binding Mechanism to Purine Nucleoside Phosphorylase Elucidated via Molecular Dynamics and Machine Learning. *Nat. Commun.* **2015**, *6*, 6155.

(6) Buch, I.; Giorgino, T.; De Fabritiis, G. Complete Reconstruction of an Enzyme-Inhibitor Binding Process by Molecular Dynamics Simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 10184−101849.

(7) Tiwary, P.; Limongelli, V.; Salvalaglio, M.; Parrinello, M. Kinetics of Protein−Ligand Unbinding: Predicting Pathways, Rates, and Rate-Limiting Steps. *Proc. Natl. Acad. Sci. U. S. A.* **2015**, *112*, E386−E391.

(8) Mollica, L.; Decherchi, S.; Zia, S. R.; Gaspari, R.; Cavalli, A.; Rocchia, W. Kinetics of Protein-Ligand Unbinding via Smoothed Potential Molecular Dynamics Simulations. *Sci. Rep.* **2015**, *5*, 11539.

(9) Mollica, L.; Theret, I.; Antoine, M.; Perron-Sierra, F.; Charton, Y.; Fourquez, J. M.; Wierzbicki, M.; Boutin, J. A.; Ferry, G.; Decherchi, S.; Bottegoni, G.; Ducrot, P.; Cavalli, A. Molecular Dynamics Simulations and Kinetic Measurements to Estimate and Predict Protein-Ligand Residence Times. *J. Med. Chem.* **2016**, *59*, 7167−7176.

(10) La Sala, G.; Decherchi, S.; De Vivo, M.; Rocchia, W. Allosteric Communication Networks in Proteins Revealed through Pocket Crosstalk Analysis. *ACS Cent. Sci.* **2017**, *3*, 949−960.

(11) Zia, S. R.; Gaspari, R.; Decherchi, S.; Rocchia, W. Probing Hydration Patterns in Class-A GPCRs via Biased MD: The A2A Receptor. *J. Chem. Theory Comput.* **2016**, *12*, 6049−6061.

(12) Tribello, G. A.; Bonomi, M.; Branduardi, D.; Camilloni, C.; Bussi, G. PLUMED2: New feathers for an old bird. *Comput. Phys. Commun.* **2014**, *185*, 604−613.

(13) Humphrey, W.; Dalke, A.; Schulten, K. VMD - Visual Molecular Dynamics. *J. Mol. Graphics* **1996**, *14*, 33−38.

(14) Case, D. A.; Cerutti, D. S.; Cheatham, T. E., III; Darden, T. A.; Duke, R. E.; Giese, T. J.; Gohlke, H.; Goetz, A. W.; Greene, D.; Homeyer, N.; Izadi, S.; Kovalenko, A.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Mermelstein, D.; Merz, K. M.; Monard, G.; Nguyen, H.; Omelyan, I.; Onufriev, A.; Pan, F.; Qi, R.; Roe, D. R.; Roitberg, A.; Sagui, C.; Simmerling, C. L.; Botello-Smith, W. M.; Swails, J.; Walker, R. C.; Wang, J.; Wolf, R. M.; Wu, X.; Xiao, L.; York, D. M.; Kollman, P. A. *AMBER 2017*; University of California, San Francisco. 2017.

(15) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved Side-Chain Torsion Potentials for the Amber ff99SB Protein Force Field. *Proteins: Struct., Funct., Genet.* **2010**, *78*, 1950−1958.

(16) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696−3713.

(17) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J. Comput. Chem.* **2004**, *25*, 1157−1174.

(18) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic Atom Type and Bond Type Perception in Molecular Mechanical Calculations. *J. Mol. Graphics Modell.* **2006**, *25*, 247−260.

(19) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L.; de Jong, W. NWChem: A Comprehensive and Scalable Open-Source Solution for Large Scale Molecular Simulations. *Comput. Phys. Commun.* **2010**, *181*, 1477−1489.

(20) Wolf, M. G.; Hoefling, M.; Aponte-Santamaría, C.; Grubmüller, H.; Groenhof, G. g_membed: Efficient Insertion of a Membrane Protein into an Equilibrated Lipid Bilayer with Minimal Perturbation. *J. Comput. Chem.* **2010**, *31*, 2169−2174.

(21) Yesylevskyy, S. O. ProtSqueeze: Simple and Effective Automated Tool for Setting up Membrane Protein Simulations. *J. Chem. Inf. Model.* **2007**, *47*, 1986−1994.

(22) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A Web-Based Graphical User Interface for CHARMM. *J. Comput. Chem.* **2008**, *29*, 1859−1865.
(23) Decherchi, S.; Rocchia, W. Building and Analyzing Molecular Surfaces: A Tutorial on NanoShaper. In *Computational Electrostatics for Biological Applications: Geometric and Numerical Approaches to the Description of Electrostatic Interaction Between Macromolecules*; Rocchia, W., Spagnuolo, M., Eds.; Springer International Publishing: Cham, 2015; pp 199−213.
(24) Decherchi, S.; Rocchia, W. A General and Robust Ray-Casting-Based Algorithm for Triangulating Surfaces at the Nanoscale. *PLoS One* **2013**, *8*, e59744.
(25) Lomize, M. A.; Pogozheva, I. D.; Joo, H.; Mosberg, H. I.; Lomize, A. L. OPM database and PPM web server: resources for positioning of proteins in membranes. *Nucleic Acids Res.* **2012**, *40* (Database issue), D370−D376.
(26) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781−1802.
(27) Páll, S.; Abraham, M. J.; Kutzner, C.; Hess, B.; Lindahl, E. Tackling Exascale Software Challenges in Molecular Dynamics Simulations with GROMACS. In *Solving Software Challenges for Exascale: International Conference on Exascale Applications and Software, EASC 2014*, Stockholm, Sweden, April 2−3, 2014; revised selected papers, Markidis, S., Laure, E., Eds.; Springer International Publishing: Cham, 2015; pp 3−27.
(28) Spyrakis, F.; Benedetti, P.; Decherchi, S.; Rocchia, W.; Cavalli, A.; Alcaro, S.; Ortuso, F.; Baroni, M.; Cruciani, G. A Pipeline To Enhance Ligand Virtual Screening: Integrating Molecular Dynamics and Fingerprints for Ligand and Proteins. *J. Chem. Inf. Model.* **2015**, *55*, 2256−2274.
(29) Baroni, M.; Cruciani, G.; Sciabola, S.; Perruccio, F.; Mason, J. S. A Common Reference Framework for Analyzing/Comparing Proteins and Ligands. Fingerprints for Ligands And Proteins (FLAP): Theory and Application. *J. Chem. Inf. Model.* **2007**, *47*, 279−294.

DOI: 10.1021/acs.jcim.7b00680
J. Chem. Inf. Model. 2018, 58, 219−224