THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

FESetup

OPEN ACCESS

FESetup: Automating Setup for Free Energy Simulations

SCHOLARONE™
Manuscripts

Ligand1

Ligand2

MCS

H
O
1
2
3
4
5
6
7
CH₃
8
9
10

H   H
N

CH₃

11
12
N   Cl
13
14
15
16
17
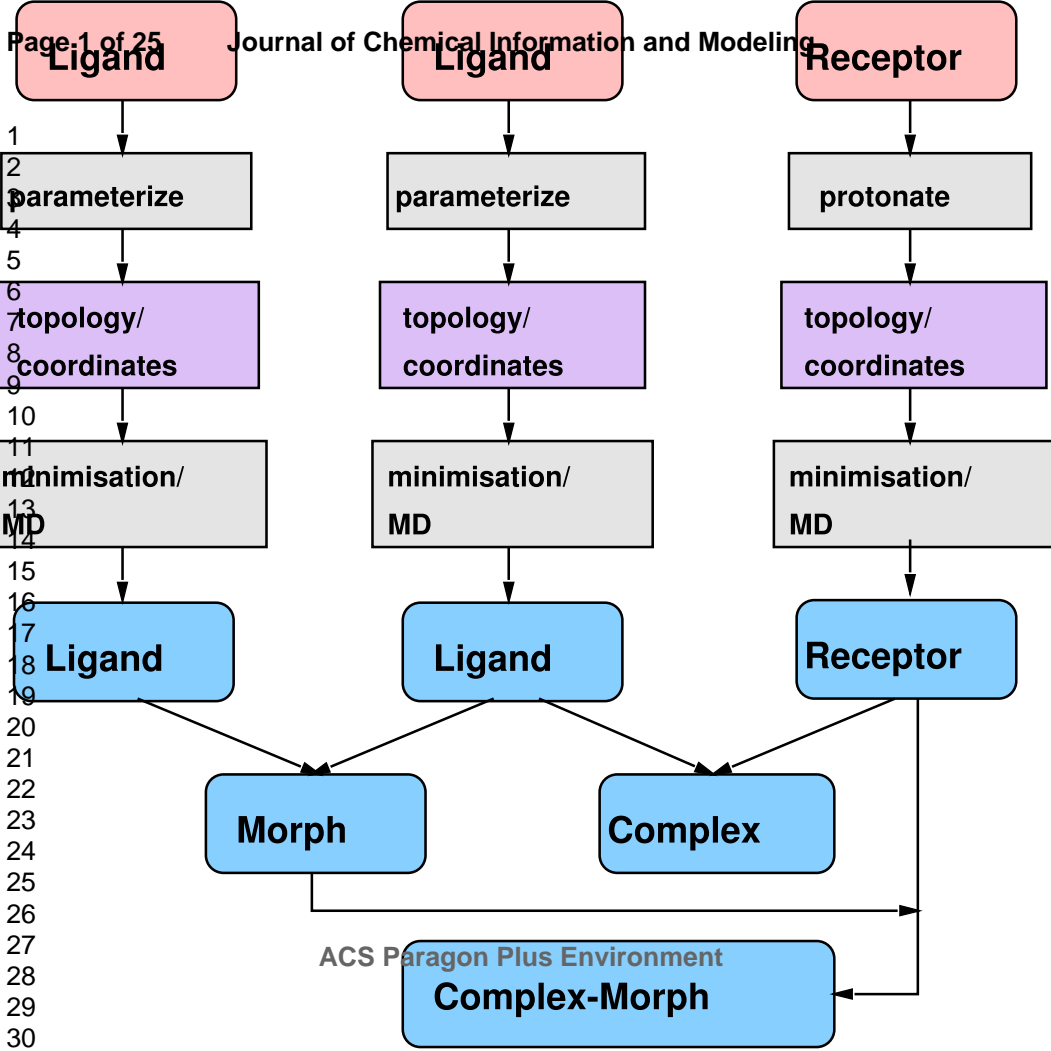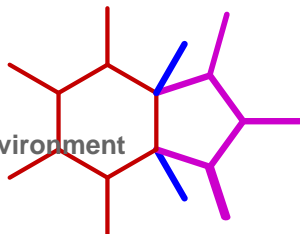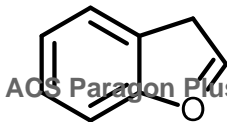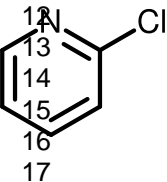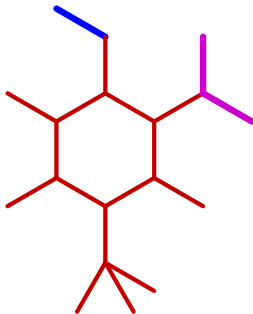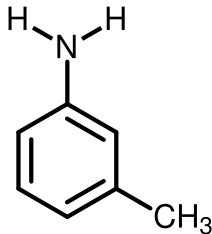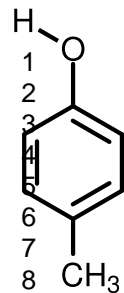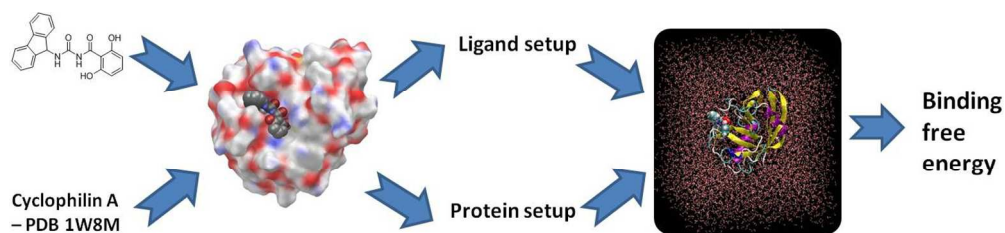
246x58mm (150 x 150 DPI)

FESetup: Automating Setup for Alchemical Free Energy Simulations

Hannes H Loeffler[1],[*],[‡], Julien Michel[2],[‡], Christopher Woods[3],[‡]

[1]*Scientific Computing Department, STFC Daresbury, Keckwick Lane, Warrington WA4 4AD, UK; E–mail: Hannes.Loeffler@stfc.ac.uk*

[2]EaStCHEM School of Chemistry, University of Edinburgh, West Mains Road, Edinburgh EH9 3JJ, UK

[3]BrisSynBio, University of Bristol, 8–10 Berkeley Square, Bristol BS8 1HH, UK

ABSTRACT: FESetup is a new pipeline tool which can be used flexibly within larger workflows. The tool aims to support fast and easy setup of alchemical free energy simulations for molecular simulation packages such as AMBER, GROMACS, Sire or NAMD. Post–processing methods like MM–PBSA and LIE can be set up as well. Ligands are automatically parametrized with AM1–BCC and atom mappings for a single topology description are computed with a maximum common substructure search (MCSS) algorithm. An abstract

molecular dynamics (MD) engine can be used for equilibration prior to free energy setup or standalone. Currently, all modern AMBER force fields are supported. Ease of use, robustness of the code and automation where it is feasible are the main development goals. The project follows an open development model and we welcome contributions.

Introduction

The setup and input preparation for simulation has become an increasingly demanding task for individual users. While many traditional work–flows are still being managed manually, today's hardware capabilities allow the computation of vast amounts of data. For instance, a system of ca. tens of thousands of atoms can easily be simulated by classical molecular dynamics for at least tens of nanoseconds per day. Such small MD simulations would typically run most efficiently on just a handful of CPU cores while modern hardware may have many thousands of cores available. GPU and other accelerators have increased in popularity in recent times and have tremendously enhanced computing power too especially for small scale installations. Thus, there is a trend for carrying out a multitude of simulations in parallel allowing for large scale comparative studies. It is also important to improve on reproducibility of simulation protocols as manual setups are often poorly documented.

This necessarily means that numerous input files and control data needs to be created to allow the running of large numbers of simulations. However, creating the necessary input files for a simulation can be a laborious and time-consuming process. This was not a problem when simulation run times vastly exceeded manual setup times. However, we are now at a stage where

manual setup is becoming a bottleneck, particularly when running large numbers of simulations. It follows there is a growing need for automated setup tools that can minimize user time. The key here is to automate work–flows for simulation setup to the maximum extent reasonable. Of course, not every step will be easily amenable to automation for various reasons including "hard" problems in science such as computing missing structural data from insufficient information and limited development of present day algorithms. However, the goal must still be to automate what is possible yet at the same time accept that automatic procedures may not always be successful. Robust protocols are needed to minimize failure and, importantly, detect and handle or report failures.

Relative alchemical free energy simulation[1] is one example where popular simulation packages still offer limited support for setup. This process is very tedious, time–consuming and can easily lead to errors as the user may have to edit dozens of files or more and reorder hundreds of lines of input by hand. Alchemical free energy simulations certainly require considerably more computing resources than e.g. docking methods so this may be one reason for limited uptake within the simulation community. However, also the aforementioned obstacles met during setup may deter potential users. Therefore, alchemical simulation setup is an interesting target for automated simulation setup, especially considering its potential role in drug design and lead optimization[2a].

Several attempts at automating the setup of free energy calculations have been reported recently. The Free Energy Workflow (FEW) tool[2] is available for AMBER[3] for the setup of relative free energy simulations and the end–point methods MM–PB/GBSA[4] and LIE[5]. Another approach which supports AMBER 11 has been reported recently[6]. PMX is a program[7] which automates the setup of relative free energy simulations of side–chain mutations for GROMACS[8],

while StaGE[9] supports absolute hydration or binding free energy calculations with GROMACS.

LOMAP is another software project[10] that does not directly provide simulation input data.

LOMAP reduces the graph of all possible ligand pairs (relative free energies) to a minimum set

based on a definition of similarity used to weight the graph's edges to solve the shortest path tree

problem. The binding affinity calculator[11] (BAC) is an automation tool for rapid computation

and analysis of ligand–receptor free energies using MM–PB/GBSA.

Here, we will discuss a new tool called FESetup. FESetup's advance over previous tools is

that it is designed to support alchemical free energy simulations in a range of different MD and

Monte Carlo (MC) packages. FESetup presents an abstraction of the setup process that is

independent of a given MD or MC code, and that is flexible enough to work within larger

workflows e.g. using docking software to provide receptor–drug structures. FESetup has been

built to be open source and flexible, providing a strong foundation to build setup workflows for

different free energy methods. Further goals are support for mutation of both non–covalently

and covalently bound moieties, maximal automation, robustness, the development of an API and

documentation of all outputs in a log file. In this Note, we will summarize current progress and

how FESetup can help the user to set up free energy calculations for codes like AMBER,

GROMACS, Sire[12] and also NAMD[13]. FESetup is free software (GPL) and is installed locally.

The project is developed in an open fashion where interested parties can contribute at all levels

including code contributions and interfacing to FESetup.

Methodology

Absolute vs Relative

There are principally two ways to compute alchemical free energies along a coupling

parameter λ[14]: the *absolute* and the *relative* approach. *Absolute* free energy changes are obtained

by completely annihilating e.g. a ligand in solution[15] and in the bound state (which yields an

absolute binding free energy), or by annihilating a ligand in different solvents (which yields an

absolute solvation or transfer free energy). The setup for this type of alchemical calculation in

modern packages like AMBER or GROMACS is very easy. Both programs allow the user to tag

a set of atoms for decoupling through specific keywords in the input file. Both also allow some

flexibility for separating the van–der–Waals free energy path from the electrostatic free energy

path. The user is *not* required to modify the topology directly as this is transparently done in–

code.

Single vs Dual–Topology

For *relative* free energy simulations there is a choice to be made between the single–topology

and the dual–topology approach[16,17]. Codes like NAMD only allow the latter at the moment and

when this approach is applied to non–covalently bound molecules it suffers from the same

problem as the absolute approach. The final end states describe a "non–existing" molecule that

can freely drift through the receptor and this needs to be compensated for with adequate

restraints[18].

Codes like AMBER or GROMACS implement hybrid approaches since they allow the

assignment of a single and a dual–topology region at the same time. The single–topology region

can serve as a convenient "anchor" to keep the ligands in place as the coordinates are shared and

only direct conversion of one atom to another is allowed to occur. The atoms within this region

are thus always present.

MCSS mapping

The single–topology approach requires a one–to–one mapping of equivalent atoms. While this

is generally a rather simple task for a human operator it requires some care when done

algorithmically. In cases where there is a well–defined anchor region as in protein backbone

atoms, the remaining atoms can be mapped through a simple distance criterion[7]. A general

solution is needed, however, in the case of arbitrary molecular structures.

Maximum common substructure search (MCSS) algorithms have been used in connection with

the definition of a similarity criterion to decide when two atoms or a bond match each other[6,10].

FESetup uses fmcs[19] which is a connected MCSS algorithm[20]. Our similarity criterion is that any

atom or any bond can match each other but rings must always match rings and rings can't be

broken[21]. Hydrogen atoms are explicitly included in the MCS. Our scheme essentially aims for

maximum overlap[7], the idea being that a maximal single–topology description is the most

efficient protocol. This also implies that the number of vanishing or appearing atoms ("dummy"

atoms) is minimal (see Chart 1). In the definition of ref. 38 this implies an implicit intermediate.

The internal representation of molecules is package−independent and thus allows generation of

any arbitrary output format.
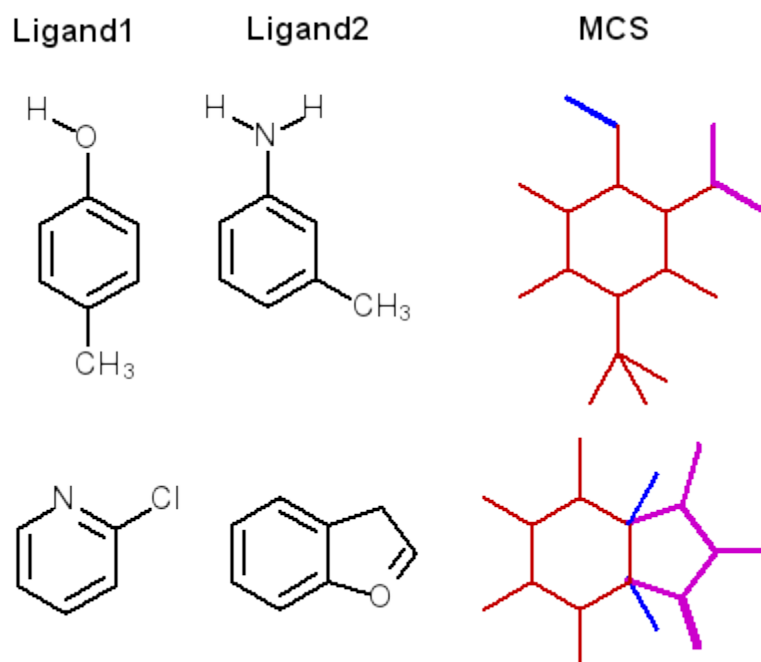
Chart 1. Example for two ligand morph pairs. The MCS (red) is shown with explicit hydrogens on the right together with vanishing (blue) and appearing atoms (magenta).

Two structures can principally have more than one common substructure[20]. In our scheme, where we ignore atom and bond type identities, even more solutions are possible, e.g. the methanol to ethanol transformation has four unique solutions in terms of our definition of similarity. Different MCSs merely define a different pathway and the free energies obtained from a closed thermodynamic cycle should be independent of the pathway chosen to connect different molecules. However, some pathways may be more computationally efficient than others. This is especially so for ligands binding to a receptor, often involving a high degree of symmetry, where the binding mode may need to be preserved. Although a ligand can, in principal at least, adjust to λ dependent changes, sufficient sampling may be very hard to achieve in practice. The preservation of the binding mode is the responsibility of the user in FESetup.

Another issue is chirality because molecular graphs are, by definition, only two–dimensional. This means that chiral centers may be inverted because the MCSS algorithm simply follows the

longest path through the graph. A solution has been proposed earlier[6] but this removes potential

mappings in the R–groups and also the asymmetric carbon.

To handle all these complexities, FESetup allows the user to arbitrarily assign desired

mappings through a file mechanism. The user creates a file for each mutation containing pairs of

mapping indices for each corresponding atom pair. This is a very general solution that allows

overwriting of any choice FESetup would otherwise take.

Workflow

Scheme 1 depicts the principal work flow in FESetup to set up a relative free energy

calculation that morphs from ligand 1 to ligand 2 while bound to a receptor. Various third party

software and toolkits are recruited for setup. Openbabel[22] is used to convert file formats, carry

out preliminary minimization and possibly create alternative conformations. The latter can be

useful to provide multiple starting structures and could potentially be used to create charges from

multiple conformations. RDKit[23] is used to compute the MCS between a ligand pair (a "morph")

with the fmcs algorithm[19]. Sire is used to read AMBER topology files and provide data

structures for force field parameters and allow for their manipulation. Furthermore, Sire can

detect and select internal degrees of freedom for Monte Carlo (MC) simulations. The

AmberTools antechamber and leap are used to create AMBER topology files including force

field parameterization.
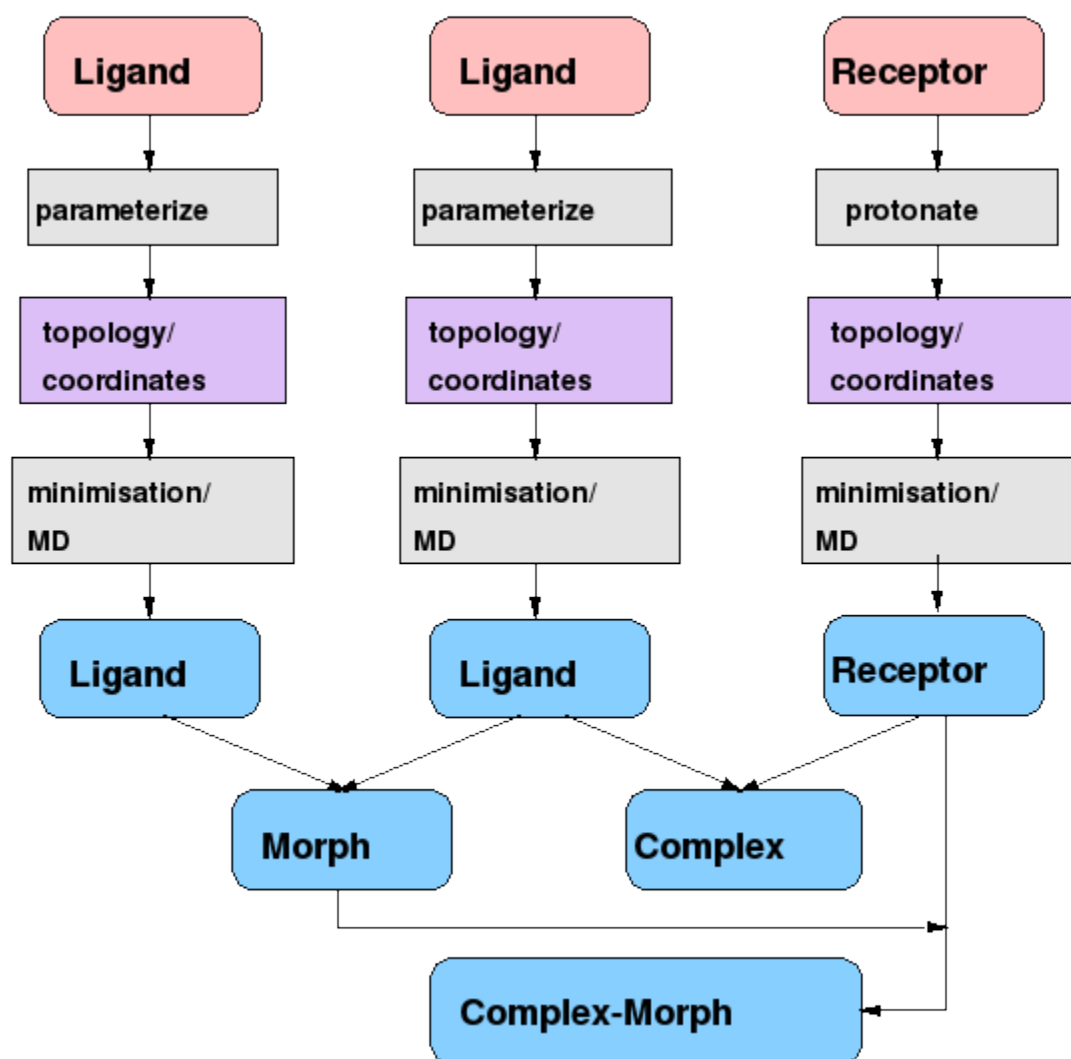
Scheme 1. The principal work flow in FESetup. Optional steps are in gray boxes. The pink boxes signify input structures and blue boxes ready simulation systems.

   The work flows for ligand and receptor are independent from each other and arbitrary numbers of either can be run. The ligand is automatically parametrized with antechamber but users can supply parameters of their own. FESetup supports currently all modern AMBER force fields for biomolecules, lipids and carbohydrates. For the ligand the default force field is GAFF[24] but this can be overwritten, e.g. for a previous free energy study on carbohydrates[25] we have used the GLYCAM force field[26]. By default atomic partial charges are derived with the AM1–BCC method[27].

The semi–empirical AM1–BCC charges are derived with sqm, the semi–empirical quantum mechanics code used to compute the atomic partial charges, which is supplied with AmberTools. The default convergence criteria are very tight, however. FESetup will reduce these criteria when SCF convergence fails. As part of this, the MMFF94 force field may be used to slightly modify the starting coordinates. This has been found to facilitate convergence in sqm in some cases. The final charges are written by antechamber with 4 digit precision only which can lead to rounding errors and a non–integer net charge. The code redistributes this charge to a higher precision evenly among all atoms to match the total molecular charge.

Protonation, and more specifically the tautomeric, state of the ligand is the responsibility of the user. A recent paper has pointed out the inherent problems of assigning these automatically with chemoinformatic tools[28]. Total charges are communicated through the input structure file. Openbabel will read these from e.g. the PDB and SDF formats but *not* the popular MOL2 format. The latter format is typically used with the antechamber tool chain.

After parametrization, leap is recruited to create topology and coordinate files for either vacuum or a solvated simulation box with counter ions. Preset minimization and MD protocols can be carried out to adjust the density and provide starting velocities via an abstraction interface that supports a number of MD engines: AMBER, GROMACS, NAMD or DL_POLY[29]. Topologies and coordinates will be transparently converted to GROMACS and DL_POLY formats (but alchemical free energy simulations are not yet supported for the latter). The direct conversion has the advantage that the newest AMBER force fields are always immediately available independent of a native port to a particular MD package.

The receptor can optionally be protonated with PROPKA3[30,31] but otherwise the work flow is

the same as for the ligand. The receptor can optionally be aligned along the principal axes. Any

biomolecule which is supported by the AMBER force fields and leap may be used.

Two ligands can be combined into a morph pair. FESetup will determine the maximum

common substructure and use this to set up the mapped region with a set of common coordinates.

Any atoms not mapped in this way are described as softcore (dummy) atoms. The code will

create all necessary topology and coordinate files for the perturbed simulation for the MD

packages AMBER, GROMACS and Sire. NAMD is principally supported too because NAMD

can read AMBER files but it is dual–topology only (see discussion above).

The receptor can be combined with a complex to form a complex–morph. The solvated box is

created from the coordinates of the unperturbed simulation system. The only additional atoms

and their coordinates are those for the dummy atoms. These will be computed from the internal

coordinates of the other state with the existing atoms. This is not necessary for AMBER[32]

because AMBER topologies can be created without explicit dummy atoms. The assumption is

made that all bonded terms involving softcore atoms will cancel in a thermodynamic cycle and

thus need not be explicitly computed. FESetup provides, however, setup with explicit dummy

atoms for AMBER too.

The force field parameters for dummy atoms have to be created except for AMBER which

handles this entirely internally. For GROMACS and Sire we follow a scheme of copying the

bond and angle force field terms from the respective other state i.e. the end state where the atoms

do exist. The end state of a bond or angle that involves dummy atoms has thus the same, non–

zero parameters as the original atom. This implies that these bonded parameters of softcore

atoms will be independent of the coupling parameter λ. Dihedrals or impropers involving dummy atoms can be set to zero.

Input/Output

FESetup has been designed to be easy to use with little input from the user and with sensible defaults but also at the same time to enable the user to overwrite decisions the software may make. Input is handled through a shell script, called FESetup, which sets up the environment and calls dGprep.py which is the actual Python code handling user input. The input file is in an INI style format very close to the configuration file format as typically used on Microsoft operating systems and is thus easy to understand and parse. The input file is divided into 4 sections: [global], [complex], [protein] (historical synonym for receptor) and [ligand] (historical synonym for unparameterized molecule) sections.

FESetup will create all topology and template control files required for simulation. The input files do not, however, prescribe a specific λ schedule. It is not clear *a priori* what λ path would guarantee a smooth gradient (TI) or sufficient energetic overlap (FEP/BAR). This will depend on the nature of the system and is still an open question.

The *Supporting Information* demonstrate typical input examples and also provide the results for the relative hydration free energies of two test system. Further validation is presented through single point energies of the end–states. Fully worked tutorials can be found on our web page (see below).

API

With FESetup we also define an application programming interface (API). In fact, dGprep.py is an elaborate example of how to use the API. See the *Supporting Information* for details.

Online material

FESetup can be downloaded from http://www.hecbiosim.ac.uk/fesetup. The source code repository including a Wiki, tutorials, a manual page, a discussion board, bug tracker and feature tracker are hosted through CCP–Forge on http://ccpforge.cse.rl.ac.uk/gf/project/ccpbiosim/. FESetup is developed in Python 2.7 and thus principally highly portable. Restrictions may apply to the third party software. FESetup comes as a pre–compiled, self–installing shell script. Binaries are available for Intel (and compatible) Linux 32 and 64 bit. FESetup is licensed as free software under the GPL2 license.

Applications

FESetup is already cited in a number of published studies and is used in several ongoing studies.

In previous work[35] FESetup was used to prepare input files for multiple ligands bound to diverse proteins and in solution. Alchemical free energy calculations were not performed, but FESetup was very useful to automate the setup process and to ensure that a consistent setup protocol was applied throughout. The resulting input files were used to produce molecular dynamics simulation trajectories and these were processed by the software nautilus[33-34] to compute enthalpies and entropies of water molecules in diverse protein binding sites using the grid cell theory method.

In another paper[25] the usefulness of the specialized carbohydrate force field GLYCAM and the general force field GAFF have been investigated in their application to a lectin complexed with various mono–saccharide pyranoses. Alchemical free energy simulations have shown that GLYCAM mostly outperforms GAFF. FESetup has been used to set up all free energy simulations. The mapping feature proved to be very useful to preserve chirality and maximum mapping at the same time.

Conclusions and Outlook

FESetup is a pipeline tool to make the setup of alchemical free energy simulations easier, faster and less error–prone. The tool can flexibly be integrated into larger workflows taking in a wide variety of structures. It creates simulation input for the MD packages AMBER, GROMACS, Sire, and, to some extent, NAMD. A maximum common substructure algorithm is used to automatically determine one–to–one mappings between start and end states. Ligands are automatically and transparently parametrized to provide AM1–BCC charges.

An abstract MD engine can be used to equilibrate (minimize, heat, pressurize, restraint release) the unperturbed simulation systems. These engines are available for AMBER, GROMACS, NAMD and DL_POLY, and can be used independently from the alchemical setup code. In principle, this can be developed into a general MD driver and combined with job submission code to off–load compute–intensive tasks to a remote HPC system.

FESetup is free software released under the GPL2 license. The code is installed locally which means that no confidential data has to be transferred over a network, is always available and the source code can be inspected and modified. The third party software is freely available too. The project follows an open development model accepting contributions of all kinds.

Future development will focus on expanding functionality to mutation of covalently bound

moieties like side–chains, and integration of absolute free energy setup. The goal here is to

allow arbitrary transformations just as with the current ligand support. Other popular MD

packages (support for the PERT module in CHARMM[36] is currently developed) and other

popular force fields like CHARMM or OPLS will be supported too. Alternative parametrization

schemes like RESP[37] can be integrated too as the work–flow is mostly automatic.

ASSOCIATED CONTENT

Supporting Information. Detailed input examples, examples of free energy changes computed

with different codes using FESetup generated input files. Table of single point energies for

several transformations setup for different simulation packages.

AUTHOR INFORMATION

Corresponding Author

*Hannes H Loeffler, *E–mail: Hannes.Loeffler@stfc.ac.uk*

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval

to the final version of the manuscript. ‡These authors contributed equally.

Notes

ACKNOWLEDGMENT

ABBREVIATIONS
MD, Molecular Dynamics; MCSS, maximum common substructure search; MCS maximum

common substructure; AM1, Austin Model 1 Hamiltonian; BCC, bond charge correction; SCF,

self–consistent field; API, application programming interface; MC, Monte Carlo;

References

(1) Michel, J. Current and emerging opportunities for molecular simulations in structure-based

drug design. *Phys. Chem. Chem. Phys.* **2014**, *16*, 4465–4477.

(2) Homeyer, N., Gohlke, H. FEW: A workflow tool for free energy calculations of ligand

binding. *J. Comput. Chem.* **2013**, *34*, 965–973.

(2a) Jorgensen, W. L. Efficient Drug Lead Discovery and Optimization. *Acc. Chem. Res.* **2009**, *42*, 724–733.

(3) Case, D. A.; Thomas E., III, C.; Darden, T.; Gohlke, H.; Luo, R.; Merz, Jr., K. M.; Onufriev, A.; Simmerling, C.; Wang, B., Woods, R. J. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(4) Srinivasan, J.; Cheatham, T. E.; Cieplak, P.; Kollman, P. A., Case, D. A. Continuum Solvent Studies of the Stability of DNA, RNA, and Phosphoramidate−DNA Helices. *J. Am. Chem. Soc.* **1998**, *120*, 9401–9409.

(5) Gutiérrez-de-Terán, H., Åqvist, J. Linear Interaction Energy: Method and Applications in Drug Design. In *Computational Drug Discovery and Design*; Baron, R., Ed.; Springer New York, 2012; pp 305–323.

(6) Christ, C. D., Fox, T. Accuracy Assessment and Automation of Free Energy Calculations for Drug Design. *J. Chem. Inf. Model.* **2014**, *54*, 108–120.

(7) Gapsys, V.; Michielssens, S.; Seeliger, D., de Groot, B. L. pmx: Automated protein structure and topology generation for alchemical perturbations. *J. Comput. Chem.* **2015**, *36*, 348–354.

(8) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E., Berendsen, H. J. C. GROMACS: Fast, flexible, and free. *J. Comput. Chem.* **2005**, *26*, 1701–1718.

(9) Lundborg, M., Lindahl, E. Automatic GROMACS Topology Generation and Comparisons of Force Fields for Solvation Free Energy Calculations. *J. Phys. Chem. B* **2015**, *119*, 810–823.

(10) Liu, S.; Wu, Y.; Lin, T.; Abel, R.; Redmann, J. P.; Summa, C. M.; Jaber, V. R.; Lim, N. M., Mobley, D. L. Lead optimization mapper: automating free energy calculations for lead optimization. *J. Comp.-Aided Mol. Des.* **2013**, *27*, 755–770.

(11) Sadiq, S. K.; Wright, D.; Watson, S. J.; Zasada, S. J.; Stoica, I., Coveney, P. V. Automated Molecular Simulation Based Binding Affinity Calculator for Ligand-Bound HIV-1 Proteases. *J. Chem. Theory Comput.* **2008**, *48*, 1909–1919.

(12) Woods, C. J., Michel, J. M. *Sire: An advanced, multiscale, molecular simulation framework*, http://siremol.org/ (accessed Sep 4, 2015).

(13) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L., Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(14) Hansen, N., van Gunsteren, W. F. Practical Aspects of Free-Energy Calculations: A Review. *J. Chem. Theory Comput.* **2014**, *10*, 2632–2647.

(15) Mobley, D., Guthrie, J. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *J. Comp.-Aided Mol. Des.* **2014**, *28*, 711–720.

(16) Boresch, S., Karplus, M. The Role of Bonded Terms in Free Energy Simulations: 1. Theoretical Analysis. *J. Phys. Chem. A* **1999**, *103*, 103–118.

(17) Michel, J., Essex, J. Prediction of protein--ligand binding affinity by free energy simulations: assumptions, pitfalls and expectations. *J. Comp.-Aided Mol. Des.* **2010**, *24*, 639–658.

(18) Gumbart, J. C.; Roux, B., Chipot, C. Standard Binding Free Energies from Computer Simulations: What Is the Best Strategy?. *J. Chem. Theory Comput.* **2013**, *9*, 794–802.

(19) Dalke, A., Hastings, J. FMCS: a novel algorithm for the multiple MCS problem. *J. Cheminform.* **2013**, *5*, O6.

(20) Raymond, J. W., Willett, P. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J. Comp.-Aided Mol. Des.* **2002**, *16*, 521–533.

(21) Liu, S.; Wang, L., Mobley, D. L. Is Ring Breaking Feasible in Relative Binding Free Energy Calculations?. *J. Chem. Inf. Mod.* **2015**, *55*, 727–735.

(22) O'Boyle, N.; Banck, M.; James, C.; Morley, C.; Vandermeersch, T., Hutchison, G. Open Babel: An open chemical toolbox. *J. Cheminform.* **2011**, *3*, 33–46.

(23) Landrum, G. *RDKit: Open-source cheminformatics*, ηττπ://ωωω.ρδκιτ.οργ (accessed Sep 4, 2015).

(24) 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.

(25) Mishra, Sushil K; Calabro, Gaetano; Loeffler, Hannes H.; Michel, J.; Koča, J. Evaluation of selected classical force fields for alchemical binding free energy calculations of protein-carbohydrate complexes. *J. Chem. Theory Comput.* **2015**, *11*, 3333–3345.

(26) Kirschner, K. N.; Yongye, A. B.; Tschampel, S. M.; González-Outeiriño, J.; Daniels, C. R.; Foley, B. L., Woods, R. J. GLYCAM06: A generalizable biomolecular force field. Carbohydrates. *J. Comput. Chem.* **2008**, *29*, 622–655.

(27) Jakalian, A.; Bush, B. L.; Jack, D. B., Bayly, C. I. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. *J. Comput. Chem.* **2000**, *21*, 132–146.

(28) Sayle, R. So you think you understand tautomerism?. *J. Comp.-Aided Mol. Des.* **2010**, *24*, 485–496.

(29) Todorov, I. T.; Smith, W.; Trachenko, K., Dove, M. T. DL_POLY_3: new dimensions in molecular dynamics simulations via massive parallelism. *J. Mater. Chem.* **2006**, *16*, 1911–1918.

(30) Søndergaard, C. R.; Olsson, M. H. M.; Rostkowski, M., Jensen, J. H. Improved Treatment of Ligands and Coupling Effects in Empirical Calculation and Rationalization of pKa Values. *J. Chem. Theory Comput.* **2011**, *7*, 2284–2295.

(31) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M., Jensen, J. H. PROPKA3:

Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. *J. Chem.*

Theory Comput. **2011**, *7*, 525–537.

(32) Kaus, J. W.; Pierce, L. T.; Walker, R. C., McCammon, J. A. Improving the Efficiency of

Free Energy Calculations in the Amber Molecular Dynamics Package. *J. Chem. Theory Comput.*

2013, *9*, 4131–4139.

(33) Michel, J.; Henchman, R. H.; Gerogiokas, G.; Southey, M. W. Y.; Mazanetz, M. P., Law,

R. J. Evaluation of Host--Guest Binding Thermodynamics of Model Cavities with Grid Cell

Theory. *J. Chem. Theory Comput.* **2014**, *10*, 4055–4068.

(34) Gerogiokas, G.; Calabro, G.; Henchman, R. H.; Southey, M. W. Y.; Law, R. J., Michel, J.

Prediction of Small Molecule Hydration Thermodynamics with Grid Cell Theory. *J. Chem.*

Theory Comput. **2014**, *10*, 35–48.

(35) Gerogiokas, G.; Southey, M. W. Y.; Mazanetz, M. P.; Hefeitz, A.; Bodkin, M.; Law, R. J.,

Michel, J. Correction: Evaluation of water displacement energetics in protein binding sites with

grid cell theory. *Phys. Chem. Chem. Phys.* **2015**, *17*, 16213–16213.

(36) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won,

Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig,

M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov,

V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.;

Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M., Karplus, M. CHARMM: The biomolecular

simulation program. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(37) Bayly, C. I.; Cieplak, P.; Cornell, W., Kollman, P. A. A well-behaved electrostatic

potential based method using charge restraints for deriving atomic charges: the RESP model. *J.*

Phys. Chem. **1993**, *97*, 10269–10280.

(38) Mobley, D. L.; Klimovich, Pavel V. Perspective: Alchemical free energy calculations for

drug discovery. *J. Chem Phys.* **2012**, *137*, 230901.

Insert Table of Contents Graphic and Synopsis Here
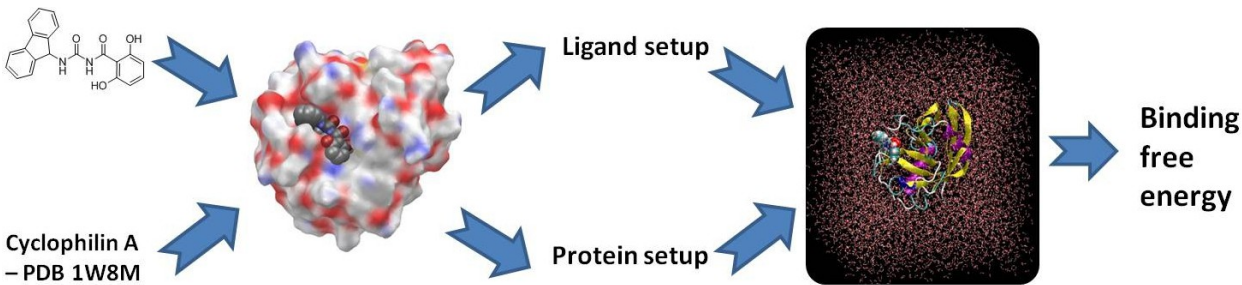
A graphical workflow of FESetup. A protein from the PDB and a ligand are combined and

solvated. Input files are created to carry out binding free energy simulations.