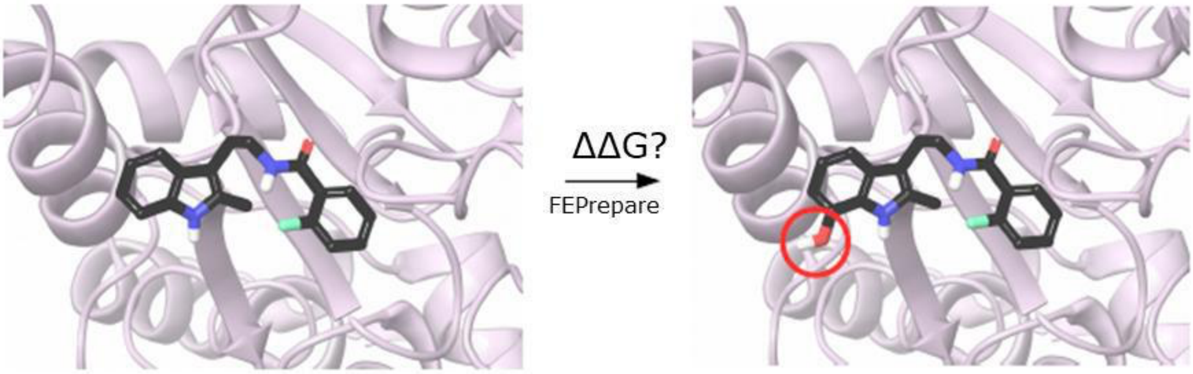
FEPrepare: A set-up tool for NAMD/FEP



Stamatia Zavitsanou, Alexandros Tsegenes & Zoe
Cournia Biomedical Research Foundation Academy
of Athens

http://feprepare.vi-seem.eu/

Table of Contents

1. INTRODUCTION

One of the most important tasks in drug design is to predict, among a series of lead candidates, which ones will bind more strongly to the therapeutic target. In this direction, relative binding free energy methodologies have been developed, which rely on physics-based molecular simulations and rigorous statistical mechanics to calculate the differences in the free energy of binding between a parent candidate drug and analogues. For example, Free Energy Perturbation (FEP) calculations calculate the free energy difference between an initial (reference) and a final (target) molecule to an average of a function of their energy difference evaluated by sampling for the initial state [1].

Automating free energy perturbation calculations is a step forward to delivering high throughput calculations for accurate predictions of relative binding affinities before a compound is synthesized, and consequently save enormous time and cost.

NAMD [2] is a free parallel molecular dynamics code, designed for high-performance simulations of large biomolecular systems. Although FEP calculations are possible with NAMD, no automated tool has been developed to streamline the process, making the calculations tedious and unfeasible for a large number of molecules. That gave us the motivation to provide an easily accessible web-based preparation tool which can produce all the files needed to run a NAMD simulation.

2. METHODOLOGY

In order to run a NAMD/FEP simulation, several inputs need to be created and no algorithm that prepares them exists. In order to create these files, one has to prepare the protein and align the ligands. Then upload these structures to LigParGen [3] in order to download the topology and the parameter files of the two ligands. Because of an inconsistency in the files that LigParGen provides, PDB, RTF, PRM, new atom names need to be given to all the atoms of both reference and mutant ligand. This is a very time-consuming process, this is why the algorithm takes care of it, with a script.

The most tedious file to create, but at the same time most important, is the Dual-Topology file. In the dual-topology approach, both reference and target state atoms exist at the same time, reference state atoms disappear, and target state atoms appear. In order to reduce the number of perturbations during the transformation, we do not just merge the two ligands into one, but rather merge the two ligands into one, keeping the reference ligand the same and adding only the atoms that are being mutated from the mutant ligand. As a result, the common part of the two ligands stays the same. The difficult thing is to decide which atoms are being mutated and therefore need to be merged with the atoms of the reference ligand. The algorithm takes into account the difference in the names of the atoms, as well as the difference in their partial charges, in order to figure out which of the atoms should be included in the calculation. Because of the modifications that take place around the area; the summation of the area's charge changes. In relative binding free energy calculations, we cannot afford to have different charges before and after the transformation. To avoid this from happening we distribute the difference of the charges before and after the transformation equally, to all the atoms that take part in our calculation (Figure 1).

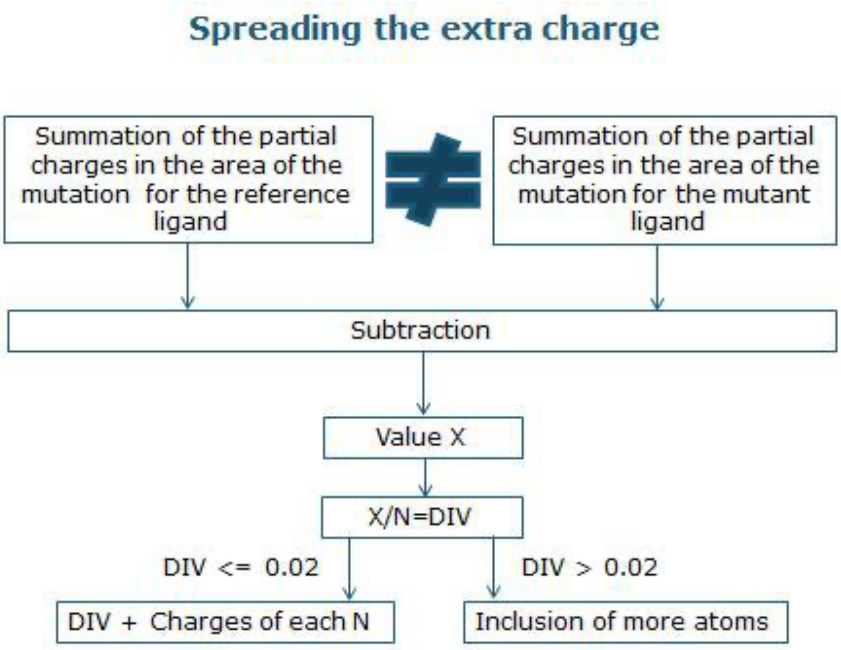

Spreading the extra charge
Summation of the partial charges in the area of the mutation for the reference ligand
≠
Summation of the partial charges in the area of the mutation for the mutant ligand
Subtraction
Value X
X/N=DIV
DIV <= 0.02
DIV > 0.02
DIV + Charges of each N
Inclusion of more atoms

Figure 1: Workflow of the algorithm that distributes the partial charges of the atoms that take part in the mutation.

After we have created the Dual-Topology file, or as some call hybrid.rtf file, we need to merge the atoms of the reference ligand with the atoms of the mutant ligand, that take part in the calculation, in the hybrid.pdb file, and update the PRM file as well. A very important file is the complex.pdb file. In order to create the complex.pdb file the algorithm merges the hybrid.pdb file with the PDB file of the protein. This file is used as an input to VMD. Since we have automated the whole procedure, there is no reason for the user to use the VMD GUI. The algorithm will do so, and generate the PSF, solvate the system in a water box with limits that are in a 15 Å distance from the atom with the greatest coordinate in each direction, and insert counterions to electrically neutralize it, respectively. In addition, the script will measure the values of the minimum, the maximum and the centre of the box. The final PSF and PDB files from the preparation that VMD did are the ionized.psf file and the ionized.pdb file, respectively.

Now, we need to create a FEP file, in which we specify, which atoms disappear, which atoms stay the same, and which atoms appear, during the simulation. These are the atoms selected in the partial charge distribution step. We call these files "ionized_complex.pdb" and "ionized_complex.fep", for the complex leg of the calculation and "ionized_solvent.pdb" and "ionized_solvent.fep", for the solvent leg of the calculation. The FEP file is simply a copy of *ionized.pdb* with a slight modification. We need to do the same things for the solvent as well. Of course, the algorithm implements that too. In the end the user can download all these files, as well as all the input files needed to run the simulation in NAMD.

In Figure 2, one can see a workflow of the methodology we have used as described above.

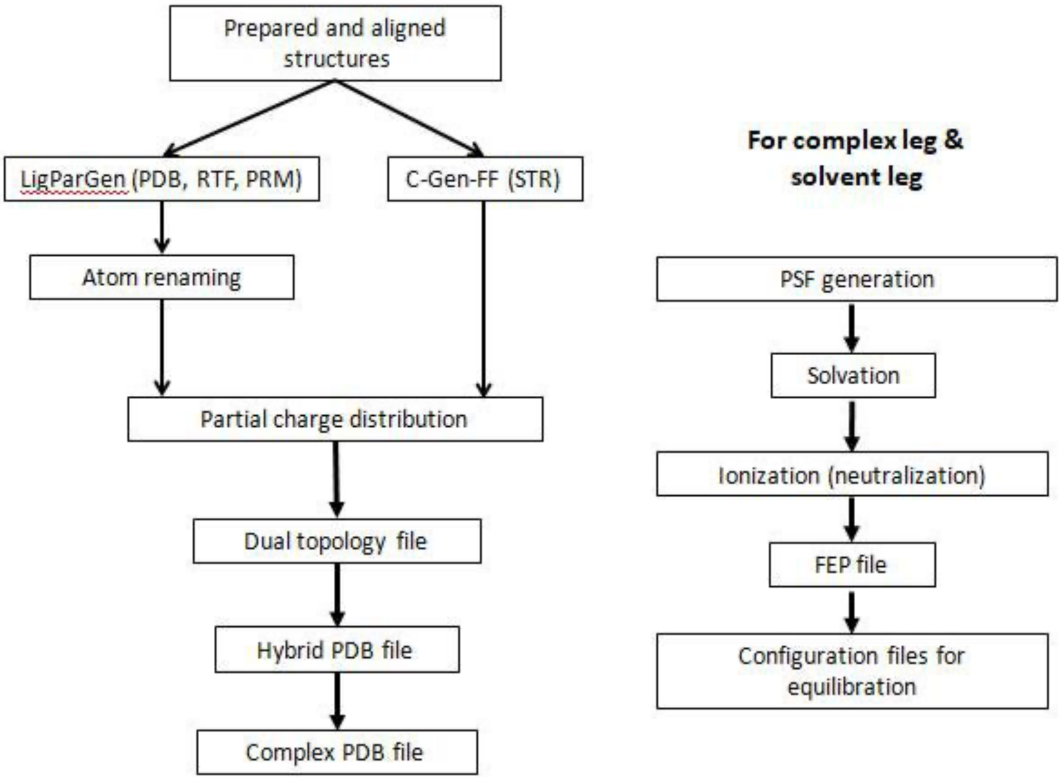

Prepared and aligned structures
LigParGen (PDB, RTF, PRM)
C-Gen-FF (STR)
Atom renaming
Partial charge distribution
Dual topology file
Hybrid PDB file
Complex PDB file
For complex leg & solvent leg
PSF generation
Solvation
Ionization (neutralization)
FEP file
Configuration files for equilibration

Figure 2: FEPrepare workflow.

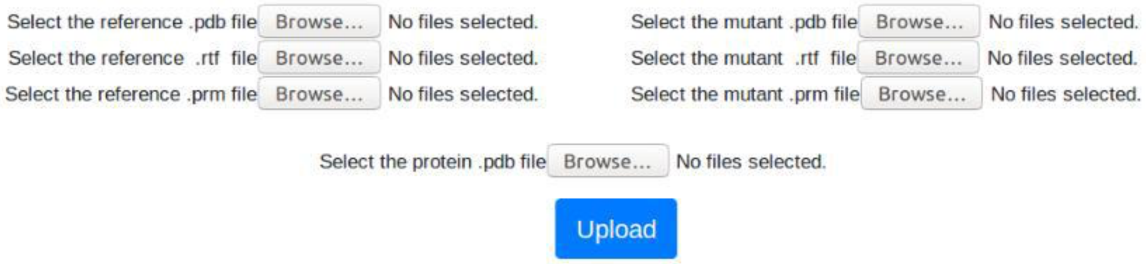
3. DESCRIPTION OF THE PROGRAM

This tool creates all the files needed to run a NAMD/FEP simulation. It has been implemented as a web-server using Python and PHP and can be accessed at: http://feprepare.vi-seem.eu/.

3.1 Input - LigParGen

The topology and parameter files

The inputs that FEPrepare needs are the coordinate, the topology and the parameter files for both ligands (reference and mutant) given as .PDB, .RTF and PRM files, and the .PDB file of the prepared protein.



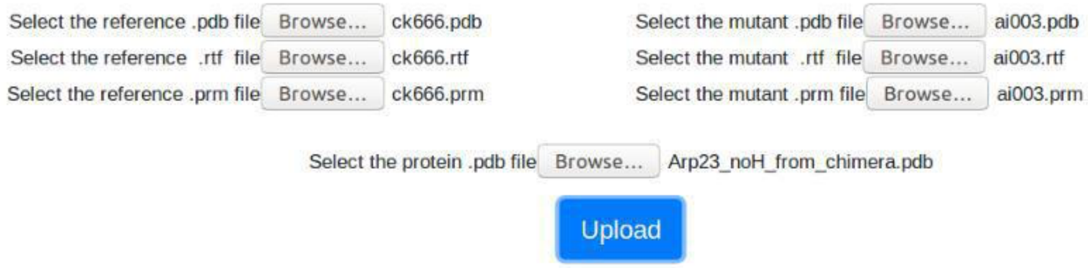
Select the reference .pdb file Browse… No files selected.
Select the reference .rtf file Browse… No files selected.
Select the reference .prm file Browse… No files selected.

Select the mutant .pdb file Browse… No files selected.
Select the mutant .rtf file Browse… No files selected.
Select the mutant .prm file Browse… No files selected.

Select the protein .pdb file Browse… No files selected.

Figure 3: File selection.

For example, we have chosen CK666 as a reference ligand and AI003 as a mutant ligand. Our protein is Arp2/3.



Select the reference .pdb file Browse… ck666.pdb
Select the reference .rtf file Browse… ck666.rtf
Select the reference .prm file Browse… ck666.prm

Select the mutant .pdb file Browse… ai003.pdb
Select the mutant .rtf file Browse… ai003.rtf
Select the mutant .prm file Browse… ai003.prm

Select the protein .pdb file Browse… Arp23_noH_from_chimera.pdb

Figure 4: Files selected.

After all these required inputs are filled hit the Upload button.

All these files can be downloaded from: http://feprepare.vi-seem.eu/example. In case one needs to see how the example works.

3.2 Output – LigParGen

As a result, you can download the files as a .zip file.

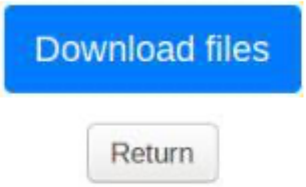


Figure 7: Files needed for NAMD/FEP simulation.

This folder should contain:

- Two subfolders, “complex” and “solvent”.
- Six renamed files (PDB, RTF, PRM) for the parent and the mutated ligand (reference.pdb, reference.rtf, reference.prm, mutant.pdb, mutant.rtf, mutant.prm).
- Two hybrid files for both parent and mutated ligand following the dual topology implementation of NAMD, named as “ligand.pdb” and “ligand.rtf”.
- The renamed PRM file after the creation of the two hybrid files, “updated.prm”.
- The “complex.pdb” file.
- The “fep.tcl” script that NAMD needs to run the simulation.
- A file with the OPLS-AA parameters of proteins as downloaded from http://zarbi.chem.yale.edu/oplsaam.html (par_opls_aam.inp).
- A file with the OPLS-AA topology of proteins as downloaded from http://zarbi.chem.yale.edu/oplsaam.html (top_opls_aam.inp).

The “complex” folder contains the following files:

- The PDB files for each of the protein chains and the ligand (in the case of the example “chainA.pdb” and “chainB.pdb” for the protein and “chainX.pdb” for the ligand).
- The files required to run VMD, “psfgen”, and “VMD_prepare_complex_after_gui_autopsf” or VMD_prepare_complex_after_gui_autopsf_ionized” in case the user requested 150mM NaCl salt addition.
- The files that VMD provides as output, i.e. “complex_wb.log”, “complex_wb.pdb”, “complex_wb.psf”, “psf-complex.psf”, “psf-complex.pdb”, which are required for VMD to generate the final files for the FEP calculation.
- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf” as generated by VMD.
- Files named “ionized_complex.pdb” and “ionized_complex.fep”, which contain the updated ionized files, after implementing the FEPrepare NAMD/FEP Workflow (see main text for details).
- Two text files: a) “min-max_center”, which provides the cell basis vectors and the cell origin to be included in the NVT equilibration input file, and b) “vmd_log”, which describes all the steps performed by VMD.
- A file named “complex-input-files.tar.gz” which provides you with the configuration files needed to run NAMD.

The “solvent” folder contains the following files:

- The files used to run VMD, “psfgen_solv”, and “VMD_prepare_ligand_after_gui_autopsf” or VMD_prepare_ligand_after_gui_autopsf_ionized” in case the user requested 150mM NaCl salt addition.
- The files that VMD gives as an output, “ligand_wb.log”, “ligand_wb.pdb”, “ligand_wb.psf”, “psf-solvated.psf”, “psf-solvated.pdb”, which are required for VMD to generate the final files for the FEP calculation.
- Files named “ionized.pdb”, “ionized.fep”, “ionized.psf” as generated by VMD.
- Files named “ionized_solvent.pdb” and “ionized_solvent.fep” which contain the updated ionized files, after implementing the FEPrepare
- NAMD/FEP Workflow (see main text for details).
- Two text files: a) “min-max_center”, which provides the cell basis vectors and the cell origin to be included in the NVT equilibration input file, and b) “vmd_log”, which describes all the steps performed by VMD.
- A file named “solvent-input-files.tar.gz” which provides you with the configuration files needed to run NAMD

These files are provided to the user so that they have full control of the FEPrepare workflow and VMD-generated files.

The actual files needed for the NAMD/FEP calculation are “ionized_complex.pdb”, “ionized_complex.fep”, “ionized_solvent.pdb”, “ionized_solvent.fep”, “ionized.psf”.

Your data will be deleted after two days. No other user has access to your data.

In case you do not see all the files mentioned above, there was probably an error during the calculation. Please check that you have provided the correct input or contact us at zavitsanoustamatia@gmail.com or zcournia@bioacademy.gr.

BIBLIOGRAPHY

[1] Athanasiou C, Vasilakaki S, Dellis D, Cournia Z. “Using Physics-based pose predictions and Free Energy Perturbation calculations to predict binding poses and relative binding affinities for FXR ligands in the D3R Grand Challenge 2” Journal of Computer-Aided Molecular Design, 2017, in press

[2] Phillips, J C, et al. “Scalable molecular dynamics with NAMD”. Journal of Computational Chemistry 2005, 26, 1781-1802.

[3] L. S. Dodda, I. Cabeza de Vaca, J. Tirado-Rives, and W. L. Jorgensen, "LigParGen web server: an automatic OPLS-AA parameter generator for organic ligands," Nucleic acids research, vol. 45, no. W1, pp. W331-W336, 2017.