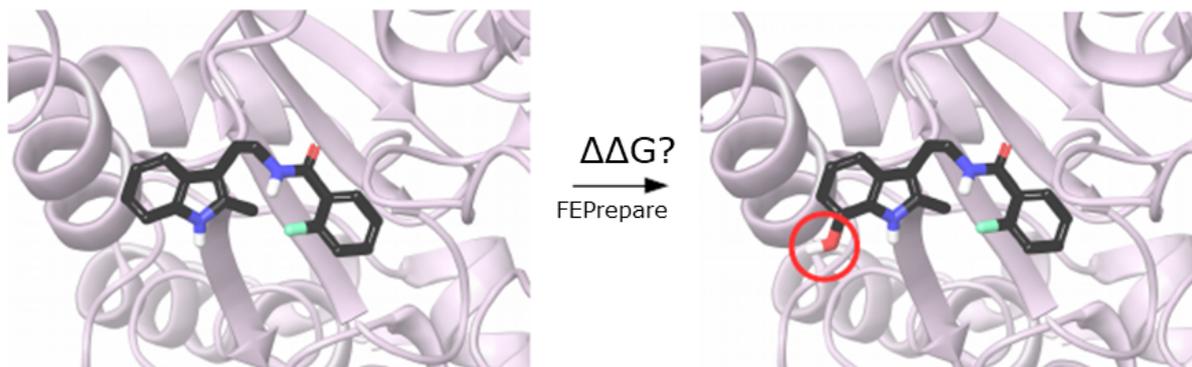


FEPprepare: A set-up tool for NAMD/FEP



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

<http://fepprepare.vi-seem.eu/>

Contents

1. INTRODUCTION	3
2. METHODOLOGY	4
3. DESCRIPTION OF THE PROGRAM.....	6
3.1 Input – C-Gen-FF.....	7
3.2 Output – Lig-Par-Gen	7
3.3 Output – C-Gen-FF.....	9
BIBLIOGRAPHY	10

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

To run a NAMD/FEP simulation, several input files need to be created. First of all, the user has to prepare the protein and align the ligands. Next the aligned ligands need to be uploaded to LigParGen [3], which will provide the topology and the parameter files of the two ligands (PDB, RTF, PRM files). New atom names need to be given to all the atoms of both reference and mutant ligands for consistency reasons. This is a very time consuming process, and the algorithm accounts for it. Alternately the user can use C-Gen-FF [4]; in this case the STR file should be acquired. The STR is actually a file that contains information about the topology and the parameters of the ligand in one file.

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 amount 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 tricky part 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).

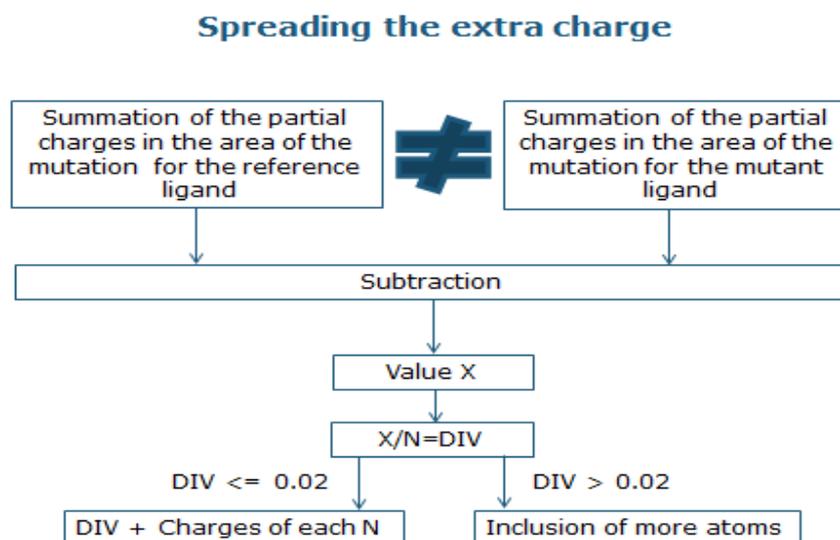


Figure 1: Workflow of the algorithm that distributes the partial charges of the atoms that take part in the mutation.

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 a hybrid.pdb file, and update the PRM file. 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 generate the PSF, solvate the system in a water box with limits that are in a 10 Å distance from the atom with the greatest coordinate in each direction, and insert counter-ions to electrically neutralize it. 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 performed are the ionized.psf file and the ionized.pdb file.

Now, we need to create a FEP file, where 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.

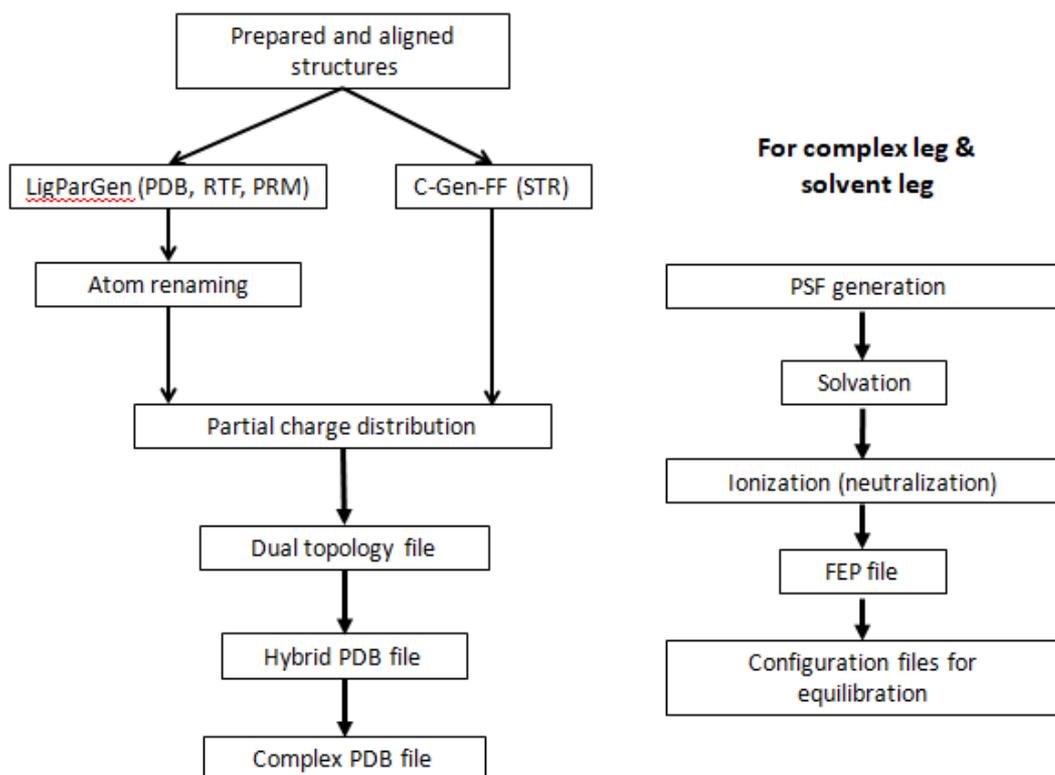


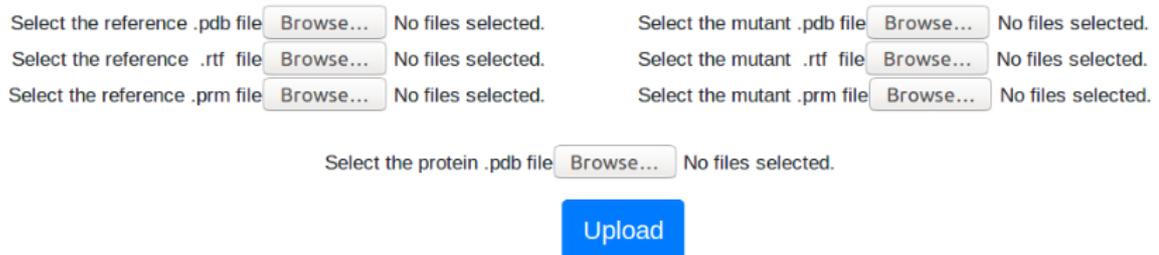
Figure 2: FEPrepare workflow.

3. DESCRIPTION OF THE PROGRAM

FEPprepare 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://fepprepare.vi-seem.eu/>

3.1 Input – Lig-Par-Gen

The inputs that FEPprepare 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.

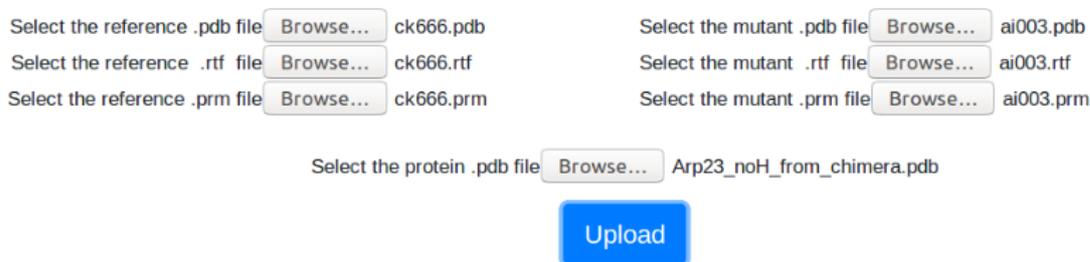


Select the reference .pdb file No files selected. Select the mutant .pdb file No files selected.
Select the reference .rtf file No files selected. Select the mutant .rtf file No files selected.
Select the reference .prm file No files selected. Select the mutant .prm file No files selected.

Select the protein .pdb file 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 ck666.pdb Select the mutant .pdb file ai003.pdb
Select the reference .rtf file ck666.rtf Select the mutant .rtf file ai003.rtf
Select the reference .prm file ck666.prm Select the mutant .prm file ai003.prm

Select the protein .pdb file Arp23_noH_from_chimera.pdb

Figure 4: Files selected.

Once you have chosen your input files, hit the Upload button.

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

3.2 Input – C-Gen-FF

The inputs that FEPrepare needs are the coordinate and the stream files for both ligands (reference and mutant) given as PDB and STR files, as well as the PDB file of the prepared protein.

Upload all requested files in order to proceed:

Select the reference ligand .pdb file	<input type="button" value="Choose Files"/>	no files selected	Select the mutant ligand .pdb file	<input type="button" value="Choose Files"/>	no files selected
Select the reference ligand .str file	<input type="button" value="Choose Files"/>	no files selected	Select the mutant ligand .str file	<input type="button" value="Choose Files"/>	no files selected
Select the protein .pdb file		<input type="button" value="Choose Files"/>	no files selected		

Figure 5: File Selection.

For example we have chosen CK666 as a reference ligand and AI007 as a mutant ligand. Our protein is Arp2/3.

Upload all requested files in order to proceed:

Select the reference ligand .pdb file	<input type="button" value="Choose Files"/>	Arp2Arp3_reb_...6_ligand.pdb	Select the mutant ligand .pdb file	<input type="button" value="Choose Files"/>	ai007_prep_align_maestro.pdb
Select the reference ligand .str file	<input type="button" value="Choose Files"/>	ck666.str	Select the mutant ligand .str file	<input type="button" value="Choose Files"/>	ai007.str
Select the protein .pdb file		<input type="button" value="Choose Files"/>	Arp23_noH_from_chimera.pdb		

Figure 6: Files selected.

Once you have chosen your input files, 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.

3.3 Output – Lig-Par-Gen

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

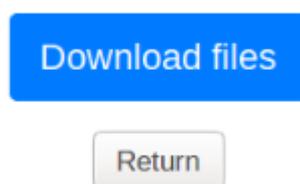


Figure 7: Files needed for NAMD/FEP simulation.

The file that the user can download as “files.zip” should contain:

- Two subfolders, “complex” and “solvent”.
- The six renamed files (PDB, RTF, PRM) for both ligands (in our case CK666 & AI003) reference.pdb, reference.rtf, reference.prm, mutant.pdb, mutant.rtf, mutant.prm).
- The two hybrid files named as “ligand.pdb” and “ligand.rtf”.
- The renamed PRM file after the creation of the two hybrid files, “updated.prm”. This file is necessary for the simulation to run.
- The “complex.pdb” file.
- The “fep.tcl” script that NAMD needs to run the simulation.
- A file with the OPLS-AA parameters of proteins (par_opls_aam.inp).
- A file with the OPLS-AA topology of proteins (top_opls_aam.inp).

In the “complex” file the user should be able to see the following:

- The PDB files for each of the protein’s chains and the ligand (in our case “chainA.pdb” and “chainB.pdb” for the protein and “chainX.pdb” for the ligand).
- Files named “ionized.pdb”, “ionized.fep”, and “ionized.psf”.
- Files named “ionized_complex.pdb” and “ionized_complex.fep” which contain the updated ionized files.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen”, and “VMD_prepare_complex_after_gui_autopsf”.
- The files that VMD gives as an output, “complex_wb.log”, “complex_wb.pdb”, “complex_wb.psf”, “psf-complex.psf”, “psf-complex.pdb”.

In the “solvent” file the user should be able to see the following:

- Files named “ionized.pdb”, “ionized.fep”, and “ionized.psf”.
- Files named “ionized_solvent.pdb” and “ionized_solvent.fep” which contain the updated ionized files.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen_solv”, and “VMD_prepare_ligand_after_gui_autopsf”.
- The files that VMD gives as an output, “ligand_wb.log”, “ligand_wb.pdb”, “ligand_wb.psf”, “psf-solvated.psf”, “psf-solvated.pdb”.

3.4 Output – C-Gen-FF

The file that the user can download as “files.zip” should contain:

- Two subfolders, “complex” and “solvent”.
- The two hybrid files named as “ligand.pdb” and “ligand.rtf”.
- The “complex.pdb” file.
- The “fep.tcl” script that NAMD needs to run the simulation.

Into the “complex” file the user should be able to see the following:

- The PDB files for each of the protein’s chains and the ligand (in our case “chainA.pdb” and “chainB.pdb” for the protein and “chainX.pdb” for the ligand).
- Files named “ionized.pdb”, “ionized.fep”, and “ionized.psf”.
- Files named “ionized_complex.pdb” and “ionized_complex.fep” which contain the updated ionized files.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen”, and “VMD_prepare_complex_after_gui_autopsf”.
- The files that VMD gives as an output, “complex_wb.log”, “complex_wb.pdb”, “complex_wb.psf”, “psf-complex.psf”, “psf-complex.pdb”.

Into the “solvent” file the user should be able to see the following:

- Files named “ionized.pdb”, “ionized.fep”, and “ionized.psf”.
- Files named “ionized_solvent.pdb” and “ionized_solvent.fep” which contain the updated ionized files.
- Two text files, “min-max_center” and “vmd_log”.
- The files used to run VMD, “psfgen_solv2”, and “VMD_prepare_ligand_after_gui_autopsf”.
- The files that VMD gives as an output, “ligand_wb.log”, “ligand_wb.pdb”, “ligand_wb.psf”, “psf-solvated.psf”, “psf-solvated.pdb”.

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.
- [4] Vanommeslaeghe, K.; MacKerell Jr, A. D., Automation of the CHARMM General Force Field (CGenFF) I: bond perception and atom typing. *Journal of chemical information and modeling* 2012, 52 (12), 3144-3154.