1 Tutorial for setting up alchemical free energy simulations using AMBER20 with AMBER_DD_BOOST

1.1 Download AMBER20 with AMBER_DD_BOOST and alchemical_fe folder from GitLab

If you have an ssh-key set up (instructions) in GitLab (recommended):

git clone git@gitlab.com:RutgersLBSR/amber-drug-discovery-boost.git
git clone git@gitlab.com:RutgersLBSR/alchemical_fe.git

or, if you do NOT have an ssh-key set up in GitLab:

git clone https://gitlab.com/RutgersLBSR/alchemical_fe.git
git clone https://gitlab.com/RutgersLBSR/amber-drug-discovery-boost.git

At the prompt you will need to provide your GitLab *username* and *password*.

Note: If you have created a new GitLab account through a social media account such as Google or Facebook, you will need to manually set up your GitLab password in order for git clone to work.

The *alchemical_fe* folder should contain the following subfolders

1. *Documentation* – containing documentation specific to AMBER20_DD_BOOST
2. *Tutorials* – containing tutorials for setting up alchemical free energy simulations using AMBER20_DD_BOOST
3. *Examples* – containing test cases for relative binding free energy (rbfe) and relative solvation free energy (rsfe) calculations
4. *Tools* – containing scripts to help set up alchemical free energy simulations using AMBER20_DD_BOOST

1.2 Purpose of *afe_setup_clean.sh*

The *afe_setup_clean.sh* script, located in the *Tools* directory, is designed to help setup the directory structure and input files that are necessary for running alchemical free energy simulations with AMBER20_DD_BOOST. The script accepts a simplified input file named '*input*' that is described in detail later in the tutorial. Briefly, for a given *system*, such as a specific protein target, or a collection of small molecules, a) a list of desired *transformations* can be provided, b) key simulation settings can be specified, and c) initial configuration files (MD equilibrated parameter (parm) and coordinate (rst) files associated with the *system* and specified *transformations*) must be provided. *afe_setup_clean.sh* can then be used to generate a hierarchy of directories containing relevant parameter, coordinate, and AMBER input files and job submission scripts. *afe_setup_clean.sh* can subsequently be used to run and check all equilibration and production free energy simulations of that particular system.

1.3 Initial Requirements for using *afe_setup_clean.sh*

1. *AMBERHOME* needs to be set
2. *cpptraj* (AmberTools) needs to be installed and available in $PATH
3. *parmed* (AmberTools) needs to be installed and available in $PATH
4. *python3* needs to be installed and available in $PATH
5. the python scripts *parmutils-timutate.py* and *parmutils-tigen.py* located in *Tools* directory either present in working directory or available in $PATH
6. *RDKit* needs to be installed and available in $PATH
    a. RDKit can be installed with a package manager
        i. Fedora - *sudo dnf install rdkit.x86_64 python3-rdkit.x86_64*
        ii. Ubuntu - *sudo apt install python-rdkit*
    b. or, RDKit can be installed using *conda*
        i. See http://www.rdkit.org/docs/Install.html#installation for details.
7. a folder that contains relevant parameter (parm) and structure (rst7) files associated with the specified *system* and list of *transformations*.

1.4 Preparing the input file for *afe_setup_clean.sh*

afe_setup_clean.sh requires an input file named '*input*' that contains key settings of the alchemical free energy simulations that are going to be set up. A template input file can be generated by running the script with a flag -h or -help. A typical input file looks like the following -

```
system=Tyk2
translist=(ejm42~ejm54 ejm42~ejm55 ejm55~ejm54)
path_to_input=../Examples/rbfe/initial
nlambda=4
protocol=unified
mapping=MCSS
ntrials=3

cutoff=8
mincyc=2000
nstlimnvt=298000
nstlimnpt=500000
repex=true
nstlimti=5000
numexchgti=1000
hmr=false
scalpha=0.5
scbeta=1.0
gti_add_sc=5
gti_scale_beta=1
gti_cut=1
gti_cut_sc_on=6
gti_cut_sc_off=8
gti_lam_sch=1
gti_ele_sc=1
gti_vdw_sc=1
```

```
gti_cut_sc=2
gti_ele_exp=2
gti_vdw_exp=2

ticalc=rbfe
stage=setup

# job submission related
partition=v100
nnodes=1
ngpus=4
wallclock=24:00:00
```

Below is a description of keywords that are specific to *afe_setup_clean.sh*. For keywords/flags that are specific to AMBER20 and AMBER20_DD_BOOST refer to the AMBER20 reference manual and AMBER20_DD_BOOST_documentation.

Variables	Entry	Description
system	*name of system*	Name to designate your protein system (for rbfe calculations) or set of ligands (for rsfe calculations). Should be consistent with the name of folder found in *path_to_input* containing input parameter and coordinate files.
translist	*ligA~ligB*	List of desired transformations. Each transformation is specified as "*initial ligand~final ligand*". Multiple transformations can be specified separated by space.
path_to_input	*path*	Path to directory containing parameter and initial coordinate files associated with *System* and transformations list in *translist*
nlambda	*positive integer*	Number of lambda windows in TI simuations
protocol	*unified/decoupled*	Set to "unified" if you want one-step protocol; Set to "decoupled" if you want three-step protocol *The current script only supports the unified protocol
mapping	*MCSS/manual/checked*	"MCSS" enables atom mapping via the Maximum Common Sub Structure algorithm available within RDKit; "manual" does not allow the script to proceed after the atom maps are generated for each transformation and allows the user to manually inspect/change the respective atom mappings. "checked" assumes manual inspection has already been done and enables the script to proceed to

		generate the final input files for each transformation.
ntrials	*real number*	The number of independent runs
cutoff	*real number*	Non-bonded cutoff
mincyc	*real number*	Max minimization cycles in lambda window equilibration
nstlimnvt	*real number*	Length of simulation for 0K to 298K NVT heating during equilibration of lambda windows
nstlimnpt	*real number*	Length of simulation for NPT equilibration of lambda windows after heating step. Ignored if *ticalc* is set to 'rsfe'
repex	*true/false*	“true” enables the use of Hamiltonian replica exchange in TI simulations
nstlimti	*real number*	Length of TI simulations
numexchgti	*real number*	Number of exchanges in replica exchange TI simulations when setting *repex*=true. numexchgti is ignored if *repex*=false
hmr	*true/false*	“true” enables the use of hydrogen mass repartitioning. *dt* (timestep) automatically changed to 4 fs from the default 1 fs
ticalc	*rbfe/rsfe*	“rbfe” enables setup for relative binding free energy calculation “rsfe” enables setup for relative solvation free energy calculation
stage	*setup/run-equil/check-equil/run-TI/check-TI*	To control the overall action of the script. “setup”: for setting up input directories/files “run-equil”: for submitting equilibration jobs for all trials of all transformations. “check-equil”: for performing a basic check on whether all equilibration jobs completed successfully. “run-TI”: for submitting production TI jobs for all trials of all transformations. “check-TI”: for performing a basic check on whether all production jobs completed successfully.
partition	*The name of specific partition on HPC*	name of specific partition on HPC. Set to “null” if it is not relevant.
nnodes	*real number*	Number of nodes to be used for each transformation
ngpus	*real number*	Number of gpu(s)/node to be used for each transformation
wallclock	*time*	Wallclock for individual jobs.

1.5 Preparation of folder containing initial parameter and coordination files

The relevant parameter and structure files should be located at '*path_to_input/system'* and organized as follows. The *path_to_input/system* folder should contain two subfolders : for rbfe calculations, the subfolders should be named *com* and *aq*, while for rsfe calculations, the subfolders should be named *aq* and *vac*. The folders *com*, *aq*, and *vac*, stand for protein-ligand complex in solution, ligand in solution, and ligand in vacuum, respectively, and should contain equilibrated parameter (*parm*) and coordinate (*rst*) files of the various ligands in the respective states.

For example, for the system Tyk2, and the transformations ejm42~ejm54, ejm42~ejm55, ejm55~ejm54; a rbfe calculation (*ticalc*=rbfe) will need the *path_to_input/Tyk2* folder to be structured as follows –

ls path_to_input/Tyk2

com aq

ls path_to_input/Tyk2/com

ejm42_com.parm7 ejm42_com.rst7 ejm54_com.parm7 ejm54_com.rst7 ejm55_com.parm7 ejm55_com.rst7

ls path_to_input/Tyk2/aq

ejm42_aq.parm7 ejm42_aq.rst7 ejm54_aq.parm7 ejm54_aq.rst7 ejm55_aq.parm7 ejm55_aq.rst7

Similary, for the set of ligands designated as SmallMol, and the transformations methanol~methane, ethane~methane, and ethane~methanol; a rsfe calculation (*ticalc*=rsfe) will need the *path_to_input/SmallMol* folder to be structured as follows –

ls path_to_input/SmallMol

aq vac

ls path_to_input/*SmallMol/aq*

ethane_aq.parm7 ethane_aq.rst7 methane_aq.parm7 methane_aq.rst7 methanol_aq.parm7 methanol_aq.rst7

ls path_to_input/*SmallMol/vac*

ethane_vac.parm7 ethane_vac.rst7 methane_vac.parm7 methane_vac.rst7 methanol_vac.parm7 methanol_vac.rst7

In the input parameter files, the respective ligands should

(a) either have residue names (resnames) identical to that used in the filename, or "*L1"*, and

(b) have identical resnames in *aq* and *com*/*vac* folders

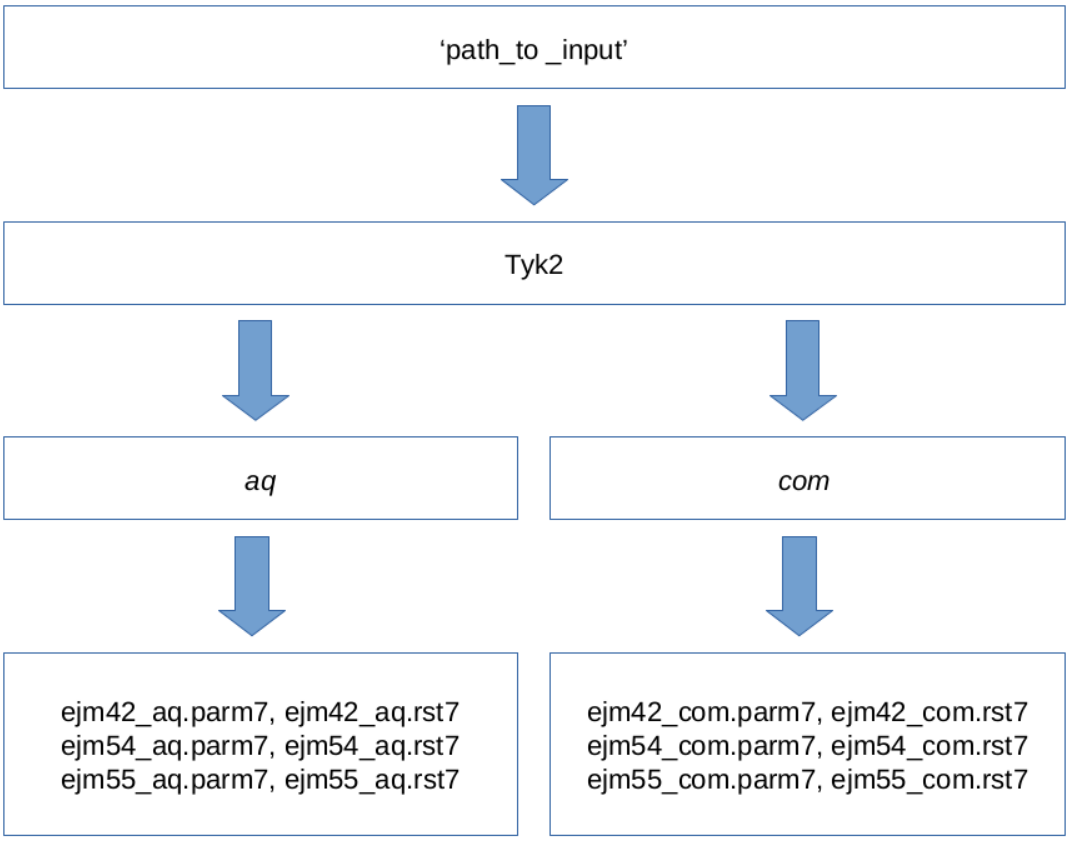

Fig 1. The structure of the *path_to_input/Tyk2* folder. For every system, there should be a separate *system* folder (Tyk2 as an example), which in turn should contain two subfolders: *com* and *aq*, for relative binding free energy (rbfe) calculations and *aq* and *vac* for relative hydration free energy (rsfe) calculations.

1.6 Setting up input files and directories using *afe_setup_clean.sh*

Once the *input* file and folder containing input parameter and coordinate files are ready, running *afe_setup_clean.sh* with *stage* set to *setup* will setup the directory structure and input files necessary for running the various free energy simulations. Intermediate files generated during the setup process and final files required to run the TI simulations will be placed in respective *build* and *run* folders.

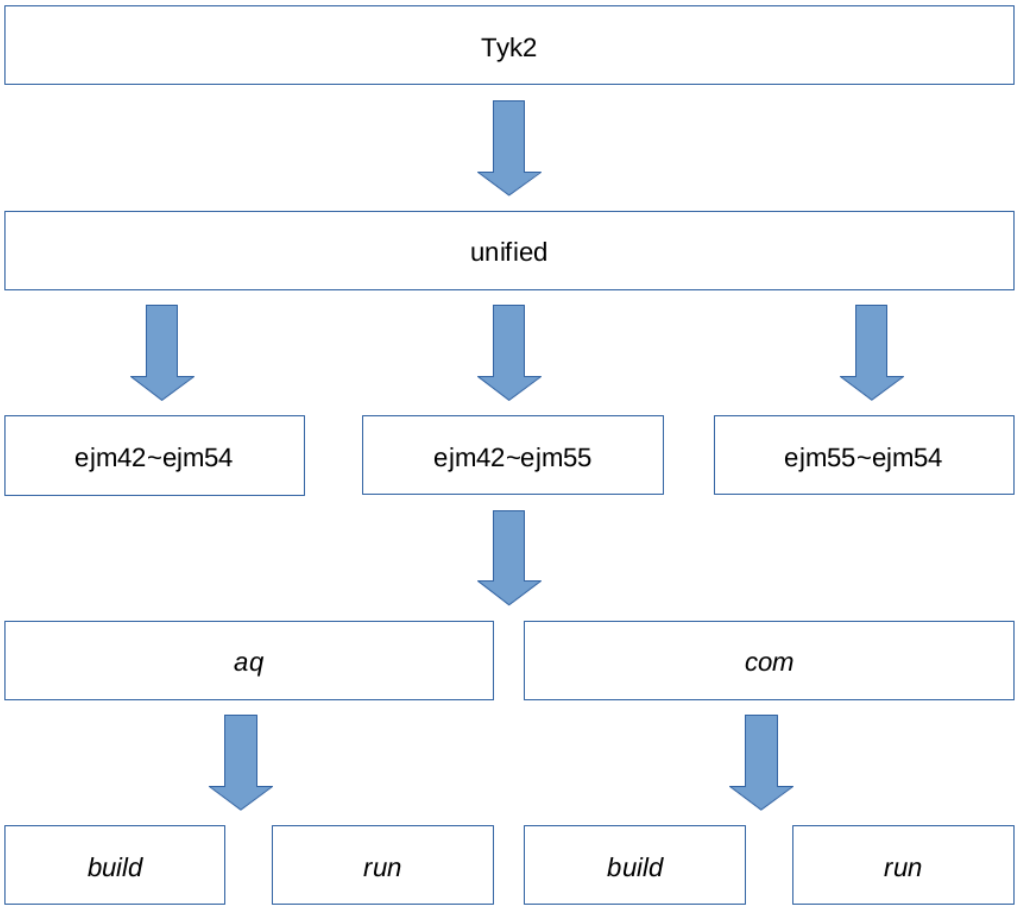

Fig 2. The structure for the Tyk2 folder prepared by *afe_clean_setup.sh* for carrying out rbfe calculations.

Note: Inside the respective *build* directories, the program *tleap*, available as part of *AmberTools*, is used within the script *ticopy.sh* to create a new parameter file that contains the solvent, macromolecule, and the two end-state ligands. The t*leap* script reads the residues and atomic coordinates from pdb files. By default, the *tleap* script loads the ff14SB protein force field, the OL3 RNA force field, TIP4P/Ew water and ion parameters, and the GAFF force field for nonstandard residues. If different parameters are required, then one needs to manually edit *ticopy.sh.* Similarly, the *tleap* script does not automatically include the *tleap* commands required to bond residues through disulfide bonds. Many proteins include disulfide bonds; therefore, failure to manually edit the *tleap* script to include bond commands which bond the disulfides will necessarily yield an incorrect parameter file. In order to manually edit the ticopy.sh and setup the input directories, afe_setup_clean.sh should be first run with *mapping*=manual. Once *ticopy.sh* has been edited, *afe_setup_clean.sh* and then be run with *mapping*=checked to complete the setup process.

1.7 Workflow and running of the TI simulations

Before the production TI simulations, an equilibration protocol is carried out for all lambda windows. Each window is first minimized, followed by a heating step at constant NVT in which

the temperature is slowly increased from 0 to 298K. For rbfe calculations, one more step is carried out in which each lambda window is subjected to a brief equilibration at constant NPT. The end structures from the equilibration stage are used as starting configurations for production TI simulations. All of these steps can easily be performed by changing *stage* in *input*.