1. **Setup**

The bash script *afe\_setup\_clean.sh* helps setting up AMBER20 TI simulations with the latest functionalities that are only available in the *AMBER\_DD\_BOOST* package and not available in the released version of AMBER20.

* 1. **Download AMBER20 with *AMBER\_DD\_BOOST***

*git clone* [*git@gitlab.com:RutgersLBSR/amber-drug-discovery-boost.git*](mailto:git@gitlab.com:RutgersLBSR/amber-drug-discovery-boost.git)

*git checkout lbsr\_dev*

* 1. **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. *RDKit* needs to be installed and available in $PATH
   1. RDKit can be installed with a package manager
      1. Fedora - *sudo dnf install rdkit.x86\_64 python3-rdkit.x86\_64*
      2. Ubuntu - *sudo apt install python-rdkit*
   2. or, RDKit can be installed using *conda*
      1. See <http://www.rdkit.org/docs/Install.html#installation> for details.
5. a folder called *initial* that contains initial configuration files.

The *initial* folder should contain a separate folder for every *system*, which in turn should contain two subfolders : For relative binding free energy (RBFE) calculations, the subfolders should be named *com* and *aq*, while for relative solvation free energy (RSFE) calculations, the subfolders should be named *aq* and *vac*. The folders *com*, *aq*, and *vac*, stands 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. The files should be named as follows :   
initial/*system*/aq/

*”ligand1”\_aq.parm7 , ”ligand1”\_aq.rst7,*

*”ligand2”\_aq.parm7, ”ligand2”\_aq.rst7,*

*“ligand3”\_aq.parm7, “ligand3”\_aq.rst7,…*

For rbfe -

initial/*system*/com/

*”ligand1”\_com.parm7, ”ligand1”\_com.rst7,*

*”ligand2”\_com.parm7, ”ligand2”\_com.rst7,*

*”ligand3”\_com.parm7,”ligand3”\_com.rst7, …*  
For rsfe -

Initial/*system*/vac/

*”ligand1”\_vac.parm7, ”ligand1”\_vac.rst7,*

*”ligand2”\_vac.parm7, ”ligand2”\_vac.rst7,*

*”ligand3”\_vac.parm7, ”ligand3”\_vac.rst7,…*

In the initial configuration files, the respective ligands should

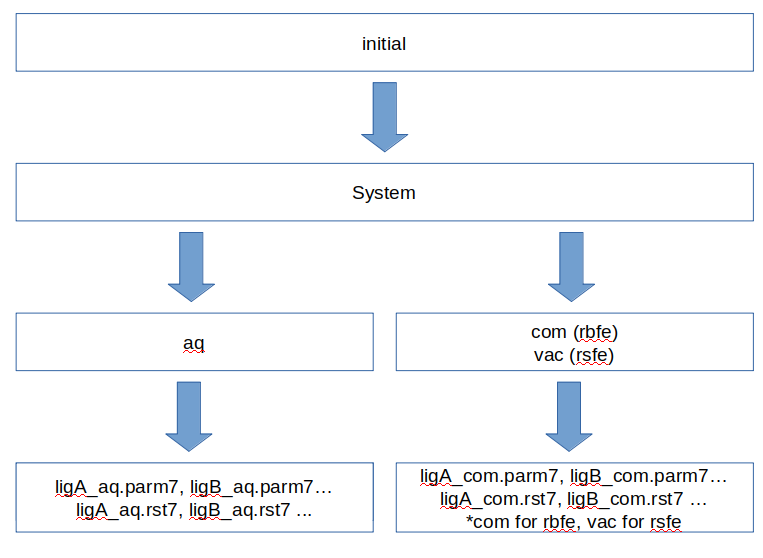
1. either have residue names (resnames) identical to that used in the filename, or “*L1”*, and
2. have identical resnames in *aq* and *com*/*vac* folders

Fig 1. The structure of the *initial* folder. For every system, there should be a separate *system* folder, 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. ***afe\_setup\_clean.sh* input file parameters**

Execute *afe\_setup\_clean.sh* with -h/-help to generate input.template. You can simply copy that as “input” and alter flags as needed. Below is a table for the description of variables.

|  |  |  |
| --- | --- | --- |
| **Variables** | **Entry** | **Description** |
| system | *your systems’ name* | Should be consistent with the folder name in “initial” folder. |
| translist | *ligA~ligB* | Put “~” between two ligands, can put more than one transformation. |
| nlambda | *real number* | The number of lambda windows |
| 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* | Set to “MCSS” if manual inspection is not needed (Maximum common core region);  Set to “manual” if atom mapping between state A and state B needs to be manually inspected;  Set to “checked” if manual inspection has already been done after running mapping=manual |
| 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 0K to 298K NVT heating of lambda windows |
| nstlimnpt | *real number* | Length of NPT equilibration of lambda windows. Ingored if ticalc is set to 'abfe' |
| repex | *true/false* | Set to “true” for the use of replica exchange in TI simulations; otherwise, set to “false” |
| 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* | Set to “true” if hydrogen mass repartitioning is applied and dt (timestep) would be set to 4 fs;  Set to “false” if hydrogen mass repartitioning is not applied and dt (timestep) would be set to 1 fs |
| scalpha | *real number* |  |
| scbeta | *real number* | If gtibeta=1, scbeta is defined as unit-less and scaled by σij |
| gti\_sc\_add | *int* | To control “gti\_add\_sc” flag:  1: AMBER18 default  2: AMBER20 default  5: AMBER20 with torsion term scaled |
| gti\_scale\_beta | *int* | To control “gti\_scale\_beta” flag:  0: default, original scbeta behavior  1: scbeta is defined as unit-less and scaled by σij |
| gti\_cut | *int* | To control “gti\_cut” flag:  0: AMBER18 default  1: No cutoff within the SC region |
| gti\_cut\_sc\_on | *real number* | To control “gti\_cut\_sc\_on” flag:  The value should be set to (cut-2.0) and it must be smaller than the value of gti\_cut\_sc\_off. |
| gti\_cut\_sc\_off | *real number* | To control “gti\_cut\_sc\_off” flag:  The ending distance of softcore smoothing and it must be smaller than the value of cutoff. |
| gti\_lam\_sch | *int* | To control “gti\_lam\_sch” flag:  0: default  1: turn on λ-scheduling |
| gti\_ele\_sc | *int* | To control “gti\_ele\_sc” flag:  0: default  1: turn on SSC(2) on lambda in electrostatic part |
| gti\_vdw\_sc | *int* | To control “gti\_vdw\_sc” flag:  0: default  1: turn on SSC(2) on lambda in vDW part |
| gti\_cut\_sc | *int* | To control “gti\_cut\_sc” flag:  0: default, no tail smoothing to SC  1: add smoothing to SC-vDW, beginning at (gti\_cut\_sc\_on) and ending at (gti\_cut\_sc\_off); using SSC(2).  2: add smoothing to SC-vDW and SC-elec, beginning at (gti\_cut\_sc\_on) and ending at (gti\_cut\_sc\_off). |
| gti\_ele\_exp | *int* | To control “gti\_ele\_exp” flag:  The exponent of relec, sc (m) in the softcore funtion; the default value is 2. |
| gti\_vdw\_exp | *int* | To control “gti\_vdw\_exp” flag:  The exponent of rvdw, sc (n) in the softcore funtion; the default value is 6. |
| ticalc | *rbfe/rsfe* | Set to “rbfe” for the relative binding free energy;  Set to “rsfe” for the relative hydration free energy |
| stage | *setup/run-equil/check-equil/run-TI/check-TI* | To control the overall action of the scipt.  “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* | Put the name of specific partition on HPC.  Set to “null” if it is not relevant. |
| nnodes | *real number* | Put the number of nodes to be used. |
| ngpus | *real number* | Put the number of gpu/node to be used. |
| wallclock | *time* | Put the wallclock for individual jobs. |

* 1. **Setting up input files using *afe\_setup\_clean.sh***

Once you have the *initial* folder and *input* file ready, run *afe\_setup\_clean.sh* with *stage* set to *setup*. The input files for TI simulations should be built automatically. 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.

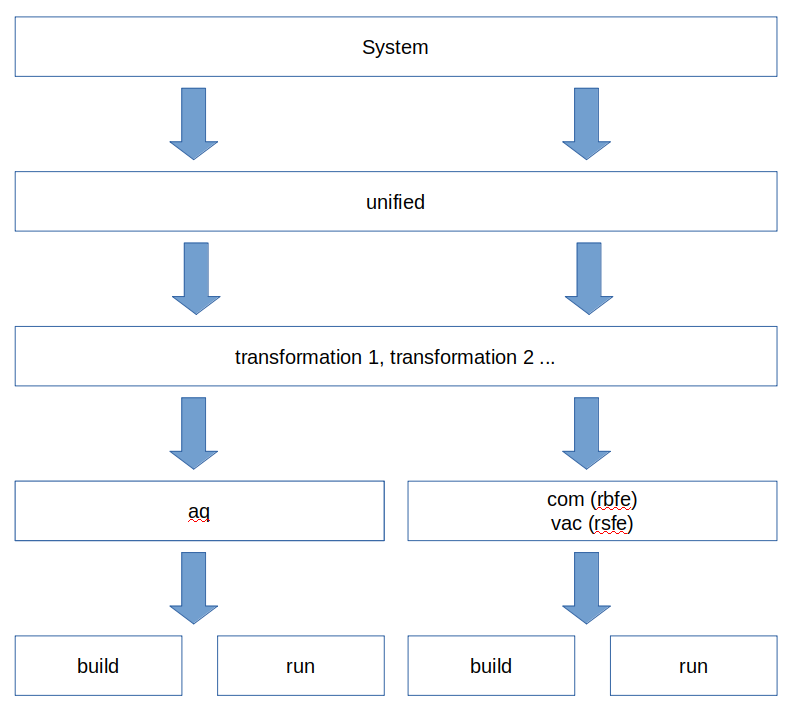


Fig 2. The structure for the system folder.

* 1. **Running TI simulations**

Before the production TI simulations, an equilibration protocol, consisting of a minimization, heating at constant NVT, and brief MD at constant NPT are performed. All of these steps can easily be performed by changing *stage* in *input*.