CHL Lab

Rosalind J. Xu 18’ June 2018

**SolEFP Protocol**

The SolEFP script is completely included in the job submission files (ir-md-solefp.sh).

To get started (and understand what solefp does), read Bartek's documentation in his Github repository: <https://github.com/globulion/slv> If he has not given you permission, ask him or Casey. In addition, ask Casey for the solefp papers.

When you are ready, go through the following steps:

**1**. Submit ir-md-solefp.sh (after ir-md-run.sh and ir-md-trim.sh, of course). You will need either solefp-files-amber or solefp-files-charmm for the solefp run, depending on which force field you are using. Put either folder in a location that is properly scanned by ir-md-solefp.sh.

-----------------------------------------------------------------------------------------------------------

🡪 ir-md-solefp.sh uses a program called launcher, which submits multiple serial jobs onto the same node, so that 48 jobs can run for 1 h on 1 node, but only 1 SU is charged. The launcher script is written in ir-md-solefp.sh: see the long echo line. Each line is run by one processor on one node.

🡪 In addition, ir-md-solefp.sh (and all other serial jobs) has to be run on skylake nodes (skx-normal) on stampede2: the KNL (normal) nodes are 7-8 times slower!

-----------------------------------------------------------------------------------------------------------

**2.** Before you scale up, do a test run to debug. Debugging is necessary for starting solefp on a new protein system. The problem with solefp is that, it will keep running even if it encounters an error, such as missing residue names, etc. This results in output frequencies that are inaccurate, and can misguide subsequent analysis and structural interpretation.

-----------------------------------------------------------------------------------------------------------

🡪 To debug, do:

**a.** Copy all contents of solefp-files into a folder. Copy the .gro file generated by the latest md run, and a single frame of trimmed trajectory (\*nojumpwholeatomwhole\*), into the folder.

**>>>** If your protein has multiple chains and your .gro file has repeated residue indexing (say, 1-40 for Chain A, then 1-20 for Chain B), you will need to use rewrite\_residue\_index.sh to renumber the indexes. If this is needed, make sure that the re-indexed .gro file is used for production run, too!

**b**. Open run\_biomol.py. Set lprint, write\_solefp\_input, and write\_debug\_file all to True. This allows the generation of debug files. These options should be turned off during production run to maximize speed.

**c.** Type in command line:

python run\_biomol.py [name of .gro file] [name of single frame trajectory] [name of output .dat file] remove\_by\_name chonhme > [name of log file]

**d.** Check the log file. Read all warning messages. If you see a warning message that says a residue is not found -- then you need to add that residue to gmx.tc. If that residue requires a new fragment, then you need to make a new fragment. See solefp-new-fragment.docx

**e**. Check the DEBUG\_FILE. Turn that file into .xyz format. In addition, make a .pdb file from that single frame (using trjconv; see gromacs\_general\_protocol.txt). Open both files in Chimera; show all atoms/bonds, including water.

**>>>** DEBUG\_FILE is the largest solefp cutoff region. Is this region spherical, or is it cut off by the periodic boundary (or boundary of the waters)? If former, good; if later, your protein is probably small and need to increase your box size in md simulation; see gromacs\_general\_protocol.txt Step 1 in Gromacs Preparation. Always increase the simulation box size; never decrease the solefp cutoffs in run\_biomol.py! The cutoffs listed in run\_biomol.py are the minima required for obtaining converged frequencies.

**>>>** How is the overlap between your protein and solefp fragments? Are there any protein side chains not represented by fragments (this happens if gmx.tc does not recognize your residue names)? Are there any poor overlaps / clashes?

**>>>** One thing about arginine: NH1,HN11,HN12 have to be cis to CD and NH2, HN21,HN22 have to be trans. If the order is reversed, the guanidinium fragment might clash with nearby fragments due to poor overlap. This problem should be addressed prior to md simulation (see cyanylation-system-manual.pdf), but if it already happened, use gmx-charmm-arg\_realigned.tc instead of gmx-charmm.tc. See the entry for ARG for difference between these two files.

-----------------------------------------------------------------------------------------------------------

3. Download the output .dat file. Open this file in TextWrangler. This file should have 23 columns. Columns 1-14 are breakdown of solefp frequencies. Column 15-19 is the electric contribution from the two nearest CONH. Column 20 is the total solefp frequency shift from the gas phase MeSCN frequency (2171 cm^-1), but an additional 3.1 cm^-1 needs to be subtracted to account for the through-bond effect on MeSCN as it is coupled to the protein backbone. So the total calculated frequency is:

column 20 (tot+env) + (-3.1 cm^-1) + 2171 cm^-1

4. Use rewrite\_time.awk to renumber the time points from 0, and spaced at the correct interval.

5. Use solefp-analysis.nb (either short or long, depending on which plots you want to make) to analyze the processed .dat file (by rewrite\_time.awk). Change the directory and filenames in the heading, and go to Evaluation/Evaluate Notebook. Interpret the outputs.

6. Take note of the average total frequency in the table (last column, Freq). This number is needed as input for lineshape conversion (see ir-md-ftir.sh).