A brief intro into AMBER PACKAGE

In this assignment you will run few semi-realistic simulations that will help you master the very basics of the AMBER suite of programs. The goal is to become familar with the input/output and key parameters. Please provide TYPED solutions, no more than 2 pages long, including pictures. One report per group, please. Indicate who did what and % effort for each group member.

- Warm-up. Skim through sections 1 through 2 of ”TUTORIAL B1: Simulating a small fragment of DNA” that can be found here: `http://ambermd.org/tutorials/Introductory.php`. The set-up of these simulations is hardcore, but, fortunately, there are web-servers that do all of it for you these days. You will not be going through all of that rigmarole, all the input files are available. Very useful (but non-mandatory) tutorials are “Learning the Unix Command-line” and ”TUTORIAL B0: An Introduction to Molecular Dynamics Simulations with AMBER” which I suggest you at least look at before doing the mandatory one above.
- Carefully do all of the steps in section 3. This is where you will actually do some type of minimization followed by molecular dynamics at constant temperature. Thease are the very besic steps that will be useful for you in your project. Use `kuprin` machine; I have explained in class how to login. Each group creates one working directory named accordingly. Use the prmtop (polyAT_vac.prmtop) and inpcrd (polyAT_vac.inpcrd) from the tutorial website. DO NOT attempt to create the input files for these simulations, just take them from the tutorial. Note that this tutorial has been available for years, and successfully completed by thousands of people, which means it is guaranteed to work if you follow the steps carefully. All the necessary code has been pre-installed for you. Some directories may have been moved in the latest AMBER, *e.g.* you may want to look into $AMBERHOME/bin/
- Explore the power of your supercomputer. Go to `example` directory and familiarize yourself with its content. See README. First, run MD on a single CPU and notice time it takes to perform 2000 steps. Then run the same one in the supercomputer mode. Look at the bottom of `mdout` file for the total time, or use unix ”time” utility. What is the speed-up? That is the ratio of the two times? If you were to simulate thioredoxin for 1 ns on GPU, how much computer time would that take?

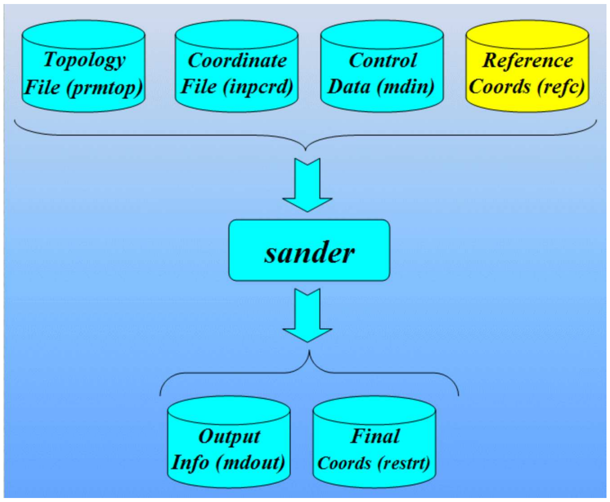

Figure 1: Data flow in Amber. prmtop - a file containing a description of the molecular topology and the necessary energy function (force-field) parameters. inpcrd (or a restrt from a previous run) - a file containing a description of the atom coordinates and optionally velocities. mdin - the sander input file consisting of a series of namelists and control variables that determine the options (e.g. how often to output snapshots) and type of simulation to be run, e.g. whether this is a minimization run or a run at constant temperature. mdout – run info, including energy of every snapshot.

• explore how the GPU speed-up depends on the structure size. Use `H++` server to prepare .crd and .top input files for a much smaller protein, 1VII. The size is similar to the one you want to fold. Just give the name at the input screen (”process PDB code”), and the server will do all of the steps for you. Use all defaults, and retrieve ”implicit solvent” coordinate and topology input files from ”VIEW RESULTS” page at the end. These are your input files to use in MD - you have just learned how to set initial conditions for your simulations. (rename the files to 1VII.crd and 1VII.top). Use those instead of the defaults in `example` directory (make sure you do this run in your own directory, not in `example`!). Report the run time and speed-up numbers in a table as a function of the number of atoms in each of the proteins you have tested. You can look up the number of atoms in ”mdout” output file. You can find the number of atoms in one of the AMBER output files, just look carefully. Or you can just count the atoms in the PDB file that you can make from the input amber files.

• Now that you have an idea of what is possible, create a VMD movie of the example protein (2trx) dynamics. Firts, download and install VMD `http://www.ks.uiuc.edu/Resear` on your laptops. Then, estimate number of steps you can afford ro run in about 1 hour on GPU (supercomputer mode). Then, figure out how to save, say, 100 equidistant snapshots (frames) from start to finish. Run the simulation and use

VMD to make a movie of the protein trajectory; include a few snaphots in your report.

• Look inside ”mdout” file, it gives you energy of your protein as a function of simulation time, that is at each step of the trajectory (for each frame). Use unix tools `grep` and `awk` to output the potential energy as a function of time (frame number). For example, in the following line: `Etot = -325.3714 EKtot = 1048.6677 EPtot = -1374.0391` you need the value of `EPtot`. Present a plot Eptot(t).