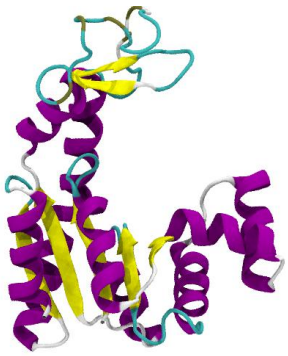
Practical 13: AdK Tutorial Documentation

Release 1.0

Oliver Beckstein

June 10, 2015



Objective: Perform aa MD simulation of the the enzyme adenylate kinase (AdK) in its open conformation without a ligand bound. Simulate it in a realistic environment (100 mM NaCl solution at $T = 300$ K and $P = 1$ bar) and analyze its structural properties.

This tutorial progresses through the individual steps needed to set up and run an equilibrium MD simulation of AdK using Gromacs.

Workflow summary

CHAPTER

ONE

DIRECTORY ORGANIZATION

The workflow for setting up, running, and analysing a simulation consists of multiple and rather different steps. It is useful to perform these different steps in separate directories in order to avoid overwriting files or using wrong files.

Directory layout

For this tutorial the suggested directory layout is the following:

```
coord/
top/
solvation/
emin/
posres/
MD/
analysis/
```

You will work through these directories in sequence.

Short description of the directories

`coord` original PDB (structural) files

`top` generating topology files (`.top`, `.itp`)

`solvation` adding solvent and ions to the system

`emin` performing energy minimization

`posres` short MD simulation with position restraints on the heavy protein atoms, to allow the solvent to equilibrate around the protein without disturbing the protein structure

`MD` MD simulation (typically, you will transfer the `md.tpr` file to a supercomputer, run the simulation there, then copy the the output back to this trajctory)

`analysis` post-processing a production trajectory to facilitate easy visualization (i.e., using VMD); analysis of the simulations can be placed in (sub)directories under analysis, e.g.

```
analysis/RMSD
analysis/RMSF
...
```

The subdirectories depend on the specific analysis tasks that you want to carry out. The above directory layout is only a suggestion but in practice some sort of ordered directory hierarchy has proven very useful.

Note: The command snippets in this tutorial assume the above directory layout. The workflow is such that each step is carried out *inside the appropriate directory* and *relative* paths are used to access files from previous steps. It should be clear from the context in which directory the commands are to be executed. If you get a `File input/output`

`error` from **`grompp`** (or any of the other commands) then check that you are able to see the file by just doing a `ls ../path/to/file` from where you are in the file system. If you can't see the file then check (1) that you are in the correct directory, (2) that you have created the file in a previous step.

CHAPTER

TWO

SETUP OF THE SOLVATED PROTEIN SYSTEM

2.1 Directory setup

- Create directories that reflects our workflow:

```
mkdir top solvation emin posres MD analysis
```

- Optional: get 4AKE and only keep chain A:

Note: The starting structure `coord/4ake_a.pdb` has been provided as part of the tutorial package so the instructions here are simply telling you what you would need to do if the file hadn't been provided.

  - Download from the protein databank through the web interface (pdb: 4AKE)
  - Modify the structure; in simple cases such as the one here, you can just open the PDB file in a text editor and remove all the lines that are not needed. In more difficult cases you might have to use molecular modeling software.
    - Remove all comment lines (but keep TITLE, HEADER)
    - Remove all crystal waters (HOH) [1]
    - Remove all chain B ATOM records.
    - Save as `coord/4ake_a.pdb`.

2.2 Generate a topology from a pdb

Generate a topology file for the CHARMM27 force field together with the TIP3P water model with pdb2gmx tool:

```
cd top
pdb2gmx -f ../coord/4ake_a.pdb -o protein.pdb -p 4ake.top -i protein_posre.itp -water tip3p -ff charm
```

Note: Total charge -4.000e (in the next step we will add ions to neutralize the system; we need a net-neutral system)

---

[1] Often you would actually want to retain crystallographic water molecules as they might have biological relevance. In our example this is likely not the case and by removing all of them we simplify the preparation step somewhat. If you keep them, **pdb2gmx** in the next step will actually create entries in the topology for them.

2.3 Solvating our system:

Adding water

Create a simulation box with editconf and add solvent with genbox:

```
cd ../solvation
editconf -f ../top/protein.pdb -o boxed.pdb -d 0.5 -bt dodecahedron -c
genbox -cp boxed.pdb -cs spc216 -p ../top/4ake.top -o solvated.pdb
```

Note: In order to reduce the system size and make the simulations run faster we are choosing a very tight box (minimum protein-edge distance 0.5 nm, `-d 0.5`); for simulations you want to publish this number should be 1.2...1.5 nm so that the electrostatic interactions between copies of the protein across periodic boundaries are sufficiently screened.

---

genbox updates the number of solvent molecules ("SOL") in the topology file (check the `[ system ]` section in `top/system.top`) [2].

Adding ions

Ions can be added with the genion program in Gromacs.

First we need a basic TPR file (an empty file is sufficient, just ignore the warnings that **grompp** spits out by setting `-maxwarn 10`), then run **genion** (which has convenient options to neutralize the system and set the concentration (check the help!); **genion** also updates the topology's `[ system ]` section if the top file is provided [2]; it reduces the "SOL" molecules by the number of removed molecules and adds the ions, e.g. "NA" and "CL").

```
touch ions.mdp
grompp -f ions.mdp -p ../top/4ake.top -c solvated.pdb -o ions.tpr
echo "SOL" | genion -s ions.tpr -pname NA -pq 1 -nname CL  -nq -1 -conc 0.1 -neutral -p ../top/4
```

The final output is `solvation/ionized.pdb`. Check visually in VMD (but note that the dodecahedral box is not represented properly.)

[2] The automatic modification of the top file by **genbox** and **genion** can become a problem if you try to run these commands multiple times and you get error messages later (typically from **grompp**) that the number of molecules in structure file and the topology file do not agree. In this case you might have to manually delete or adjust the corresponding lines in :file"*system.top* file.

CHAPTER

THREE

ENERGY MINIMIZATION

In order to remove "clashes" (i.e. close overlaps of the LJ cores) we perform an *energy minimization*: Instead of a MD simulation we use an algorithm to change the coordinates in such a way as to reduce the total potential energy.

3.1 Telling Gromacs what it will do

First, we will copy a file from the templates folder (provided in this tutorial) that tells Gromacs MD program *how* to do energy minimization:

```
cp ../templates/em.mdp .
```

Note: The MDP file `em.mdp` is provided in `templates/em.mdp`: copy it to the `emin/` directory. You should have a look at it and modify it according to your needs. The individual parameters are explained under mdp options.

This `.mdp` file contains the settings that dictate the nature of the simulation. For energy minimization, we will use the simple *steepest descent* minimizer (`integrator = steep` in `em.mdp`, which runs in parallel). Use grompp (the GROMacs PreProcessor) to generate the run input file (TPR) from the run parameter file (MDP), coordinate file (the solvated system with ions; PDB), and the topology (TOP):

```
cd ../emin
grompp -f em.mdp -c ../solvation/ionized.pdb -p ../top/4ake.top -o em.tpr
```

3.2 Performing an energy minimization run

The energy minimization is performed with **`mdrun`** but by using the appropriate `integrator` option in the Run control options in the MDP file it has been instructed to do a energy minimization:

```
mdrun -v -s em.tpr -deffnm em -c em.pdb
```

Ideally, the maximum force *Fmax* (gradient of the potential) should be < 1e+03 kJ mol^{-1} nm^{-2} (but typically anything below 1e+05 kJ mol^{-1} nm^{-2} works). See the screen output or the `em.log` file for this information.

Note: If you want to minimize further, you can use the :file"*em.pdb* structure as an input for a second run with either the *conjugate gradients* (`integrator = cg`) or the Newton-like *Broyden-Fletcher-Goldfarb-Shanno* (`integrator = l-bfgs`) minimizer. For details see Run control options in the MDP file.

CHAPTER

FOUR

POSITION RESTRAINTS MD

We first perform a short MD simulation with harmonic position restraints on the heavy protein atoms. This allows the solvent to equilibrate around the protein without disturbing the protein structure. In addition, we use "weak coupling" temperature and pressure coupling algorithms to obtain the desired temperatue and pressure.

4.1 Telling Gromacs what it will do (again)

We must first tell Gromacs *how* to perform our equilibration run in the same way that we did for the energy minimization step. This step requires the `top/protein_posres.itp` file with the default value for the harmonic force constants of 1000 kJ mol^{-1} nm^{-2}. The position restraints are switched on by setting the `-DPOSRES` flag in the `posres.mdp` file (see mdp options).

Create the run input (TPR) file, using the *energy minimized system* as the starting structure:

```
cd ../posres
cp ../templates/posres.mdp .
grompp -f posres.mdp -o posres.tpr -p ../top/4ake.top -c ../emin/em.pdb -maxwarn 2
```

The mdp file contains cut-off settings that approximate the native CHARMM values (in the CHARMM program).

Weak (Berendsen) coupling is used for both temperature and pressure to quickly equilibrate. The protein and the solvent (water and ions) are coupled as separate groups. Gromacs provides a range of groups automatically (run `make_ndx -f` *TPR* to see them) and we use the groups `Protein` and `non-Protein` (these particularly groups work roughly since Gromacs 4.5.3). If the standard groups do not work then you will have to create the groups yourself using `make_ndx -f` *TPR* `-o md.ndx` (which would save them in a file `md.ndx`) and supply it to `grompp -n md.ndx`.

4.2 Performing the equilibration run

Run the position restraints equilibration simulation:

```
mdrun -v -stepout 10 -s posres.tpr -deffnm posres -c posres.pdb
```

(If this is too slow on your workstation, submit to saguaro using 8 cores.)

Note: Here the runtime of 10 ps is too short for real production use; typically 1 - 5 ns are used.

In order to visually check your system, create trajectory with all molecules in the primary unitcell (`-ur compact`; see also below the more extensive notes on *Trajectory visualization*):

```
echo "System" | trjconv -ur compact -s posres.tpr -f posres.xtc -pbc mol -o posres_ur.xtc
```

Check visually:

```
vmd ../emin/em.pdb posres_ur.xtc
```

(If you don't have a **vmd** command available on the command line then launch VMD, load the `emin/em.pdb` file (*File* → *New Molecule...*), highlight your molecule 1 ("em.pdb") and load the `posres/posres_ur.xtc` trajectory into your molecule 1", *File* → *Load Data Into Molecule*. You should see that the first frame (from the energy minimization) looks as if the water is in a distorted box shape whereas all further frames show a roughly spherical unit cell (the rhombic dodecahedron).)

CHAPTER

FIVE

EQUILIBRIUM MOLECULAR DYNAMICS

5.1 Setup the production run

As usual, we must tell Gromacs what it will be doing using grompp before we can perform our production simulation. Since we want to start our run where we left off (after doing equilibration), we prepare the TPR input file based on the last frame of the *Position restraints MD* with grompp:

```
cd ../MD
cp ../templates/md.mdp .
grompp -f md.mdp -p ../top/4ake.top -c ../posres/posres.pdb -o md.tpr -maxwarn 3
```

The `md.mdp` file uses different algorithms from the *Position restraints MD* for the temperature and pressure coupling, which are known to reproduce the exact *NPT* ensemble distribution.

5.2 Running the simulation

CPU run

If your workstation has a decent number of cores or if you simply don't mind waiting a bit longer you can also run the simulation as usual:

```
mdrun -v -stepout 10 -s md.tpr -deffnm md -cpi
```

This will automatically utilize all available cores. The `-cpi` flag indicates that you want Gromacs to continue from a previous run. You can kill the job with `CONTROL-C`, look at the output, then continue with exactly the same command line

```
mdrun -v -stepout 10 -s md.tpr -deffnm md -cpi
```

(Try it out!). The `-cpi` flag can be used on the first run without harm. For a continuation to occur, Gromacs needs to find the checkpoint file `md.cpt` and all output files (`md.xtc`, `md.edr`, `md.log`) in the current directory.

GPU run

We can also try utilizing the GPU(s) available on the workstation. We use a modified MDP file that contains settings compatible with GPU-based Gromacs simulations to generate a new TPR file, which is used to perform the simulation.

grompp -f mdgpu.mdp -p ../top/4ake.top -c ../posres/posres.pdb -o mdgpu.tpr -maxwarn 3
mdrun -v -stepout 10 -s mdgpu.tpr -deffnm mdgpu -cpi

CHAPTER

SIX

TRAJECTORY VISUALIZATION

Analysis are normally performed locally on a workstation, i.e. copy back all the files from the supercomputer to your local directory.

A typical analysis tasks reads the trajectory (XTC) or energy (EDR) file, computes quantities, and produces data files that can be plotted or processed further, e.g. using Python scripts. A strength of Gromacs is that it comes with a wide range of tools that each do one particular analysis task well (see the Gromacs manual and the Gromacs documentation).

6.1 Keeping our protein in one piece

If you just look at the output trajectory `md.xtc` in VMD then you will see that the protein can be split across the periodic boundaries and that the simulation cell just looks like a distorted prism. You should *recenter* the trajectory so that the protein is at the center, *remap* the water molecules (and ions) to be located in a more convenient unitcell representation.

We will use the trjconv tool in Gromacs to center and remap our system.

Note: **`trjconv`** asks the user a number of questions that depend on the chosen options. In the command line snippets below, this user input is directly fed to the standard input of **`trjconv`** with the `printf TEXT | trjconv` "pipe" construct. In order to better understand the command, run it interactively without the pipe construct and manually provide the required information.

Center (`-center`) on the *Protein* and remap all the molecules (`-pbc mol`) of the whole *System*:

```
printf "Protein\nSystem\n" | trjconv -s md.tpr -f md.xtc -center -ur compact -pbc mol -o md_center.xt
```

6.2 Pinning down a tumbling protein

It is often desirable to *RMS-fit* the protein on a reference structure (such as the first frame in the trajectory) to remove overall translation and rotation. In Gromacs, the trjconv tool can also do more "trajectory conversion tasks". After (1) centering and remapping the system, we want to (2) RMS-fit (due to technical limitations in **`trjconv`** you cannot do both at the same time).

RMS-fit (`-fit rot+trans`) to the protein *backbone* atoms in the initial frame (supplied in the TPR file) and write out the whole *System*:

```
printf "Backbone\nSystem\n" | trjconv -s md.tpr -f md_center.xtc -fit rot+trans -o md_fit.xtc
```

6.3 Check our modified trajectory

Visualize in VMD:

```
vmd ../posres/posres.pdb md_fit.xtc
```

Workflow details

For this tutorial we use Gromacs (version 4.6.6) to set up the system, run the simulation, and perform analysis. The overall work flow contains the following steps:

1. Download tutorial files and organize the work space
2. Setup
   - Obtain structure 4AKE from PDB, select chain A
   - Use default protonation states
   - Generate topology
   - Solvate in water in simulation cell (rhombic dodecahedron)
   - Add NaCl ions to neutralize and final physiological concentration of 100 mM
3. Energy minimization (EM)
4. Position restraint equilibration of solvent (MD); *NPT* (use weak coupling (Berendsen) schemes)
5. Equilibrium MD simulation (unrestrained, *NPT*, use Nose-Hoover and Parrinello-Rahman temperature and pressure coupling)
6. Trajectory visualization
   - Center the protein in the box (periodic boundary conditions)
   - RMS-fit the protein in each snapshot to the first snapshot

All input files are provided in the same directory as `AdKTutorial.html`. Start by uncompressing the package file:

```
tar -jxvf AdKTutorial.tar.bz2
cd AdKTutorial
```

A starting structure can be found in the `AdKTutorial/coord` directory and MDP files are in `AdKTutorial/templates`.

CHAPTER

SEVEN

REFERENCES

Indices and tables

- *References*
- search

[Kabsch1983] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. 1983 22 2577-2637. doi: 10.1002/bip.360221211

[Willis1975] Willis & Pryor, Thermal vibrations in crystallography, Cambridge Univ. Press, 1975

[Henzler-Wildman2007] K. A. Henzler-Wildman, V. Thai, M. Lei, M. Ott, M. Wolf-Watz, T. Fenn, E. Pozharski, M. A. Wilson, G. A. Petsko, M. Karplus, C. G. Hübner, and D. Kern. Intrinsic motions along an enzymatic reaction trajectory. *Nature*, 450:838–844, Dec 2007. doi: 10.1038/nature06410

[Beckstein2009] O. Beckstein, E. J. Denning, J. R. Perilla, and T. B. Woolf. Zipping and unzipping of adenylate kinase: Atomistic insights into the ensemble of open/closed transitions. *J. Mol. Biol.*, 394(1):160–176, 2009. doi: 10.1016/j.jmb.2009.09.009.