Chapter 8

Analysis

In this chapter different ways of analyzing your trajectory are described. The names of the corresponding analysis programs are given. Specific information on the in- and output of these programs can be found in the online manual at www.gromacs.org. The output files are often produced as finished Grace/Xmgr graphs.

First, in sec. 8.1, the group concept in analysis is explained. 8.1.2 explains a newer concept of dynamic selections, which is currently supported by a few tools. Then, the different analysis tools are presented.

8.1 Using Groups

`gmx make_ndx, gmx mk_angndx, gmx select`

In chapter 3, it was explained how *groups of atoms* can be used in `mdrun` (see sec. 3.3). In most analysis programs, groups of atoms must also be chosen. Most programs can generate several default index groups, but groups can always be read from an index file. Let's consider the example of a simulation of a binary mixture of components A and B. When we want to calculate the radial distribution function (RDF) $g_{AB}(r)$ of A with respect to B, we have to calculate:

$$4\pi r^2 g_{AB}(r) \;=\; V \sum_{i \in A}^{N_A} \sum_{j \in B}^{N_B} P(r) \tag{8.1}$$

where V is the volume and $P(r)$ is the probability of finding a B atom at distance r from an A atom.

By having the user define the *atom numbers* for groups A and B in a simple file, we can calculate this g_{AB} in the most general way, without having to make any assumptions in the RDF program about the type of particles.

Groups can therefore consist of a series of *atom numbers*, but in some cases also of *molecule numbers*. It is also possible to specify a series of angles by *triples* of *atom numbers*, dihedrals by *quadruples* of *atom numbers* and bonds or vectors (in a molecule) by *pairs* of *atom numbers*.

When appropriate the type of index file will be specified for the following analysis programs. To help creating such index files (`index.ndx`), there are a couple of programs to generate them, using either your input configuration or the topology. To generate an index file consisting of a series of *atom numbers* (as in the example of g_{AB}), use `gmx make_ndx` or `gmx select`. To generate an index file with angles or dihedrals, use `gmx mk_angndx`. Of course you can also make them by hand. The general format is presented here:

```
[ Oxygen ]
   1        4        7

[ Hydrogen ]
   2        3        5        6
   8        9
```

First, the group name is written between square brackets. The following atom numbers may be spread out over as many lines as you like. The atom numbering starts at 1.

Each tool that can use groups will offer the available alternatives for the user to choose. That choice can be made with the number of the group, or its name. In fact, the first few letters of the group name will suffice if that will distinguish the group from all others. There are ways to use Unix shell features to choose group names on the command line, rather than interactively. Consult www.gromacs.org for suggestions.

8.1.1 Default Groups

When no index file is supplied to analysis tools or `grompp`, a number of default groups are generated to choose from:

`System`
: all atoms in the system

`Protein`
: all protein atoms

`Protein-H`
: protein atoms excluding hydrogens

`C-alpha`
: C_α atoms

`Backbone`
: protein backbone atoms; N, C_α and C

`MainChain`
: protein main chain atoms: N, C_α, C and O, including oxygens in C-terminus

`MainChain+Cb`
: protein main chain atoms including C_β

`MainChain+H`
: protein main chain atoms including backbone amide hydrogens and hydrogens on the N-terminus

`SideChain`
: protein side chain atoms; that is all atoms except N, C_α, C, O, backbone amide hydrogens, oxygens in C-terminus and hydrogens on the N-terminus

`SideChain-H`
: protein side chain atoms excluding all hydrogens

`Prot-Masses`
: protein atoms excluding dummy masses (as used in virtual site constructions of NH_3 groups and tryptophan side-chains), see also sec. 5.2.2; this group is only included when it differs from the "`Protein`" group

`Non-Protein`
: all non-protein atoms

`DNA`
: all DNA atoms

`RNA`
: all RNA atoms

`Water`
: water molecules (names like `SOL`, `WAT`, `HOH`, etc.) See `residuetypes.dat` for a full listing

`non-Water`
: anything not covered by the `Water` group

`Ion`
: any name matching an Ion entry in `residuetypes.dat`

`Water_and_Ions`
: combination of the `Water` and `Ions` groups

`molecule_name`
: for all residues/molecules which are not recognized as protein, DNA, or RNA; one group per residue/molecule name is generated

`Other`
: all atoms which are neither protein, DNA, nor RNA.

Empty groups will not be generated. Most of the groups only contain protein atoms. An atom is considered a protein atom if its residue name is listed in the `residuetypes.dat` file and is listed as a "Protein" entry. The process for determinding DNA, RNA, etc. is analogous. If you need to modify these classifications, then you can copy the file from the library directory into your working directory and edit the local copy.

8.1.2 Selections

`gmx select`
Currently, a few analysis tools support an extended concept of *(dynamic) selections*. There are three main differences to traditional index groups:

- The selections are specified as text instead of reading fixed atom indices from a file, using a syntax similar to VMD. The text can be entered interactively, provided on the command line, or from a file.
- The selections are not restricted to atoms, but can also specify that the analysis is to be performed on, e.g., center-of-mass positions of a group of atoms. Some tools may not support selections that do not evaluate to single atoms, e.g., if they require information that is available only for single atoms, like atom names or types.
- The selections can be dynamic, i.e., evaluate to different atoms for different trajectory frames. This allows analyzing only a subset of the system that satisfies some geometric criteria.

As an example of a simple selection, `resname ABC and within 2 of resname DEF` selects all atoms in residues named ABC that are within 2 nm of any atom in a residue named DEF.

Tools that accept selections can also use traditional index files similarly to older tools: it is possible to give an `.ndx` file to the tool, and directly select a group from the index file as a selection, either by group number or by group name. The index groups can also be used as a part of a more complicated selection.

To get started, you can run `gmx select` with a single structure, and use the interactive prompt to try out different selections. The tool provides, among others, output options `-on` and `-ofpdb` to write out the selected atoms to an index file and to a `.pdb` file, respectively. This does not allow testing selections that evaluate to center-of-mass positions, but other selections can be tested and the result examined.

The detailed syntax and the individual keywords that can be used in selections can be accessed by typing `help` in the interactive prompt of any selection-enabled tool, as well as with `gmx help selections`. The help is divided into subtopics that can be accessed with, e.g., `help syntax` / `gmx help selections syntax`. Some individual selection keywords have extended help as well, which can be accessed with, e.g., `help keywords within`.

The interactive prompt does not currently provide much editing capabilities. If you need them, you can run the program under `rlwrap`.

For tools that do not yet support the selection syntax, you can use `gmx select -on` to generate static index groups to pass to the tool. However, this only allows for a small subset (only the first bullet from the above list) of the flexibility that fully selection-aware tools offer.

It is also possible to write your own analysis tools to take advantage of the flexibility of these selections: see the `template.cpp` file in the `share/gromacs/template` directory of your installation for an example.

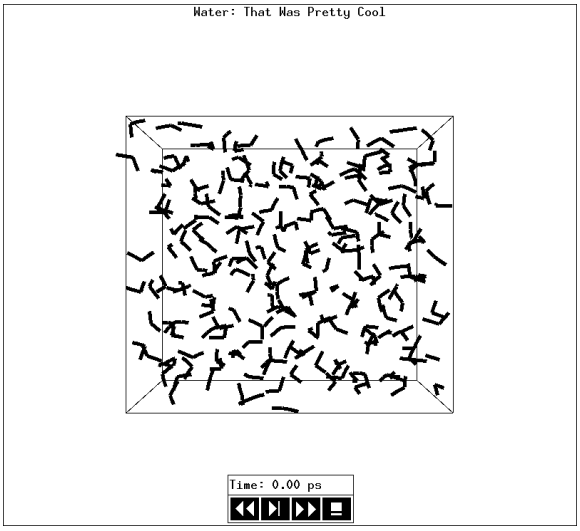

Figure 8.1: The window of `gmx view` showing a box of water.

8.2 Looking at your trajectory

`gmx view`
Before analyzing your trajectory it is often informative to look at your trajectory first. GROMACS comes with a simple trajectory viewer `gmx view`; the advantage with this one is that it does not require OpenGL, which usually isn't present on *e.g.* supercomputers. It is also possible to generate a hard-copy in Encapsulated Postscript format (see Fig. 8.1). If you want a faster and more fancy viewer there are several programs that can read the GROMACS trajectory formats – have a look at our homepage (www.gromacs.org) for updated links.

8.3 General properties

`gmx energy, gmx traj`
To analyze some or all *energies* and other properties, such as *total pressure*, *pressure tensor*, *density*, *box-volume* and *box-sizes*, use the program `gmx energy`. A choice can be made from a list a set of energies, like potential, kinetic or total energy, or individual contributions, like Lennard-Jones or dihedral energies.

The *center-of-mass velocity*, defined as

$$\mathbf{v}_{com} = \frac{1}{M} \sum_{i=1}^{N} m_i \mathbf{v}_i \tag{8.2}$$

with $M = \sum_{i=1}^{N} m_i$ the total mass of the system, can be monitored in time by the program `gmx traj -com -ov`. It is however recommended to remove the center-of-mass velocity every step (see chapter 3)!

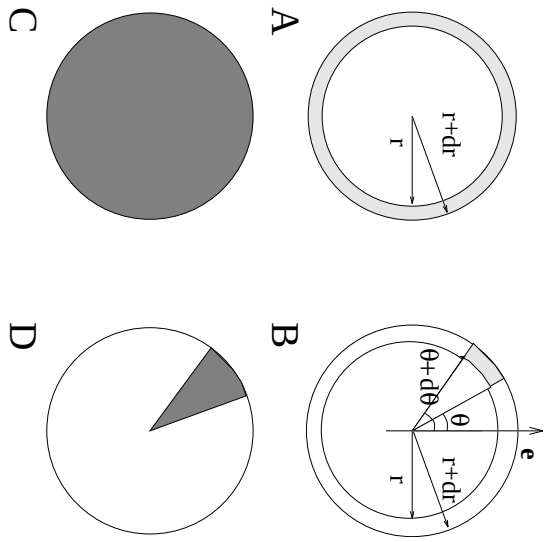

Figure 8.2: Definition of slices in `gmx rdf`: A. $g_{AB}(r)$. B. $g_{AB}(r,\theta)$. The slices are colored gray. C. Normalization $\langle\rho_B\rangle_{local}$. D. Normalization $\langle\rho_B\rangle_{local,\,\theta}$. Normalization volumes are colored gray.

8.4 Radial distribution functions

`gmx rdf`

The *radial distribution function* (RDF) or pair correlation function $g_{AB}(r)$ between particles of type A and B is defined in the following way:

$$\begin{aligned} g_{AB}(r) &= \frac{\langle\rho_B(r)\rangle}{\langle\rho_B\rangle_{local}} \\ &= \frac{1}{\langle\rho_B\rangle_{local}} \frac{1}{N_A} \sum_{i\in A}^{N_A} \sum_{j\in B}^{N_B} \frac{\delta(r_{ij}-r)}{4\pi r^2} \end{aligned} \tag{8.3}$$

with $\langle\rho_B(r)\rangle$ the particle density of type B at a distance r around particles A, and $\langle\rho_B\rangle_{local}$ the particle density of type B averaged over all spheres around particles A with radius r_{max} (see Fig. 8.2C).

Usually the value of r_{max} is half of the box length. The averaging is also performed in time. In practice the analysis program `gmx rdf` divides the system into spherical slices (from r to $r+dr$, see Fig. 8.2A) and makes a histogram in stead of the δ-function. An example of the RDF of oxygen-oxygen in SPC water [81] is given in Fig. 8.3.

With `gmx rdf` it is also possible to calculate an angle dependent rdf $g_{AB}(r,\theta)$, where the angle θ is defined with respect to a certain laboratory axis $\mathbf{e}$, see Fig. 8.2B.

$$g_{AB}(r,\theta) = \frac{1}{\langle\rho_B\rangle_{local,\,\theta}} \frac{1}{N_A} \sum_{i\in A}^{N_A} \sum_{j\in B}^{N_B} \frac{\delta(r_{ij}-r)\delta(\theta_{ij}-\theta)}{2\pi r^2 sin(\theta)} \tag{8.4}$$

$$cos(\theta_{ij}) = \frac{\mathbf{r}_{ij}\cdot\mathbf{e}}{\|r_{ij}\|\;\|e\|} \tag{8.5}$$

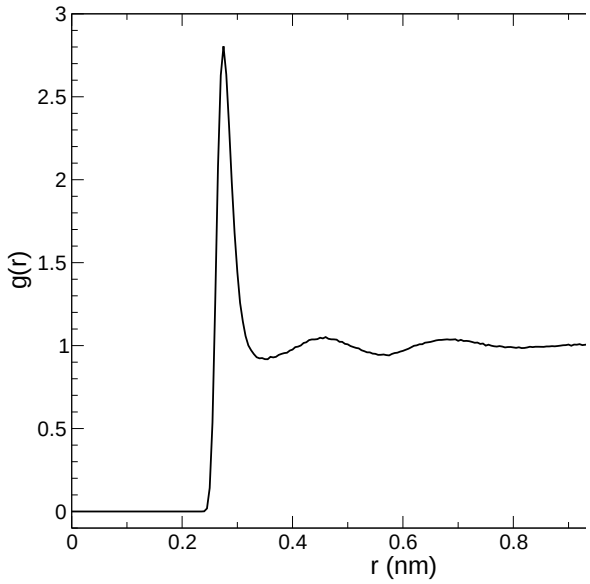

Figure 8.3: $g_{OO}(r)$ for Oxygen-Oxygen of SPC-water.

This $g_{AB}(r,\theta)$ is useful for analyzing anisotropic systems. **Note** that in this case the normalization $\langle\rho_B\rangle_{local,\,\theta}$ is the average density in all angle slices from θ to $\theta+d\theta$ up to r_{max}, so angle dependent, see Fig. 8.2D.

8.5 Correlation functions

8.5.1 Theory of correlation functions

The theory of correlation functions is well established [106]. We describe here the implementation of the various correlation function flavors in the GROMACS code. The definition of the (ACF) $C_f(t)$ for a property $f(t)$ is:

$$C_f(t) \;=\; \langle f(\xi)f(\xi+t)\rangle_\xi \tag{8.6}$$

where the notation on the right hand side indicates averaging over ξ, *i.e.* over time origins. It is also possible to compute cross-correlation function from two properties $f(t)$ and $g(t)$:

$$C_{fg}(t) \;=\; \langle f(\xi)g(\xi+t)\rangle_\xi \tag{8.7}$$

however, in GROMACS there is no standard mechanism to do this (**note:** you can use the `xmgr` program to compute cross correlations). The integral of the correlation function over time is the correlation time τ_f:

$$\tau_f \;=\; \int_0^\infty C_f(t)\mathrm{d}t \tag{8.8}$$

In practice, correlation functions are calculated based on data points with discrete time intervals Δt, so that the ACF from an MD simulation is:

$$C_f(j\Delta t) \;=\; \frac{1}{N-j}\sum_{i=0}^{N-1-j} f(i\Delta t)f((i+j)\Delta t) \tag{8.9}$$

where N is the number of available time frames for the calculation. The resulting ACF is obviously only available at time points with the same interval Δt. Since, for many applications, it is necessary to know the short time behavior of the ACF (*e.g.* the first 10 ps) this often means that we have to save the data with intervals much shorter than the time scale of interest. Another implication of eqn. 8.9 is that in principle we can not compute all points of the ACF with the same accuracy, since we have $N-1$ data points for $C_f(\Delta t)$ but only 1 for $C_f((N-1)\Delta t)$. However, if we decide to compute only an ACF of length $M\Delta t$, where $M \leq N/2$ we can compute all points with the same statistical accuracy:

$$C_f(j\Delta t) \;=\; \frac{1}{M}\sum_{i=0}^{N-1-M} f(i\Delta t)f((i+j)\Delta t) \tag{8.10}$$

Here of course $j < M$. M is sometimes referred to as the time lag of the correlation function. When we decide to do this, we intentionally do not use all the available points for very short time intervals ($j << M$), but it makes it easier to interpret the results. Another aspect that may not be neglected when computing ACFs from simulation is that usually the time origins ξ (eqn. 8.6) are not statistically independent, which may introduce a bias in the results. This can be tested using a

block-averaging procedure, where only time origins with a spacing at least the length of the time lag are included, *e.g.* using k time origins with spacing of $M\Delta t$ (where $kM \leq N$):

$$C_f(j\Delta t) \;=\; \frac{1}{k}\sum_{i=0}^{k-1} f(iM\Delta t)f((iM+j)\Delta t) \tag{8.11}$$

However, one needs very long simulations to get good accuracy this way, because there are many fewer points that contribute to the ACF.

8.5.2 Using FFT for computation of the ACF

The computational cost for calculating an ACF according to eqn. 8.9 is proportional to N^2, which is considerable. However, this can be improved by using fast Fourier transforms to do the convolution [106].

8.5.3 Special forms of the ACF

There are some important varieties on the ACF, *e.g.* the ACF of a vector $\boldsymbol{p}$:

$$C_{\boldsymbol{p}}(t) \;=\; \int_0^\infty P_n(\cos\angle\left(\boldsymbol{p}(\xi), \boldsymbol{p}(\xi+t)\right)\mathrm{d}\xi \tag{8.12}$$

where $P_n(x)$ is the n^{th} order Legendre polynomial [1]. Such correlation times can actually be obtained experimentally using *e.g.* NMR or other relaxation experiments. GROMACS can compute correlations using the 1^{st} and 2^{nd} order Legendre polynomial (eqn. 8.12). This can also be used for rotational autocorrelation (`gmx rotacf`) and dipole autocorrelation (`gmx dipoles`).

In order to study torsion angle dynamics, we define a dihedral autocorrelation function as [153]:

$$C(t) \;=\; \langle\cos(\theta(\tau)-\theta(\tau+t))\rangle_\tau \tag{8.13}$$

Note that this is not a product of two functions as is generally used for correlation functions, but it may be rewritten as the sum of two products:

$$C(t) \;=\; \langle\cos(\theta(\tau))\cos(\theta(\tau+t)) \;+\; \sin(\theta(\tau))\sin(\theta(\tau+t))\rangle_\tau \tag{8.14}$$

8.5.4 Some Applications

The program `gmx velacc` calculates the *velocity autocorrelation function*.

$$C_{\boldsymbol{v}}(\tau) \;=\; \langle\boldsymbol{v}_i(\tau)\cdot\boldsymbol{v}_i(0)\rangle_{i\in A} \tag{8.15}$$

The self diffusion coefficient can be calculated using the Green-Kubo relation [106]:

$$D_A \;=\; \frac{1}{3}\int_0^\infty \langle\mathbf{v}_i(t)\cdot\mathbf{v}_i(0)\rangle_{i\in A}\,dt \tag{8.16}$$

[1] $P_0(x)=1$, $P_1(x)=x$, $P_2(x)=(3x^2-1)/2$

which is just the integral of the velocity autocorrelation function. There is a widely-held belief that the velocity ACF converges faster than the mean square displacement (sec. 8.6), which can also be used for the computation of diffusion constants. However, Allen & Tildesley [106] warn us that the long-time contribution to the velocity ACF can not be ignored, so care must be taken.

Another important quantity is the dipole correlation time. The *dipole correlation function* for particles of type A is calculated as follows by `gmx dipoles`:

$$C_\mu(\tau) \;=\; \langle \mu_i(\tau) \cdot \mu_i(0) \rangle_{i \in A} \tag{8.17}$$

with $\mu_i = \sum_{j \in i} \mathbf{r}_j q_j$. The dipole correlation time can be computed using eqn. 8.8. For some applications see [154].

The viscosity of a liquid can be related to the correlation time of the Pressure tensor $\boldsymbol{P}$ [155, 156]. `gmx energy` can compute the viscosity, but this is not very accurate [137], and actually the values do not converge.

8.6 Mean Square Displacement

`gmx msd`
To determine the self D_A of particles of type A, one can use the Einstein relation [106]:

$$\lim_{t \to \infty} \langle \|\mathbf{r}_i(t) - \mathbf{r}_i(0)\|^2 \rangle_{i \in A} \;=\; 6 D_A t \tag{8.18}$$

This *mean square displacement* and D_A are calculated by the program `gmx msd`. Normally an index file containing atom numbers is used and the MSD is averaged over these atoms. For molecules consisting of more than one atom, $\mathbf{r}_i$ can be taken as the center of mass positions of the molecules. In that case, you should use an index file with molecule numbers. The results will be nearly identical to averaging over atoms, however. The `gmx msd` program can also be used for calculating diffusion in one or two dimensions. This is useful for studying lateral diffusion on interfaces.

An example of the mean square displacement of SPC water is given in Fig. 8.4.