8.7 Bonds/distances, angles and dihedrals

`gmx distance, gmx angle, gmx gangle`
To monitor specific *bonds* in your modules, or more generally distances between points, the program `gmx distance` can calculate distances as a function of time, as well as the distribution of the distance. With a traditional index file, the groups should consist of pairs of atom numbers, for example:

```
[ bonds_1 ]
 1     2
 3     4
 9    10
```

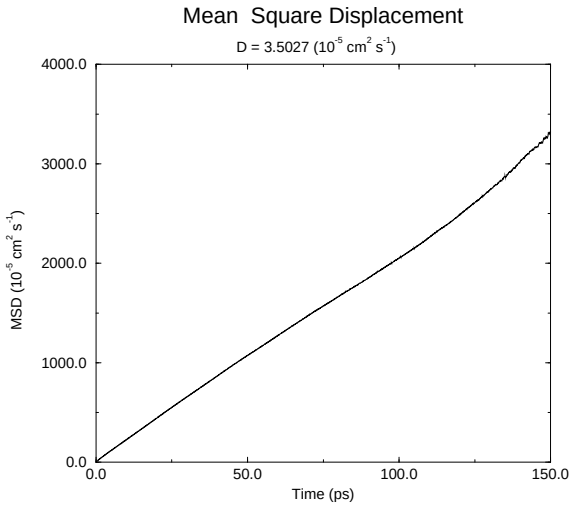

Figure 8.4: Mean Square Displacement of SPC-water.

```
[ bonds_2 ]
12    13
```

Selections are also supported, with first two positions defining the first distance, second pair of positions defining the second distance and so on. You can calculate the distances between CA and CB atoms in all your residues (assuming that every residue either has both atoms, or neither) using a selection such as:

```
name CA CB
```

The selections also allow more generic distances to be computed. For example, to compute the distances between centers of mass of two residues, you can use:

```
com of resname AAA plus com of resname BBB
```

The program `gmx angle` calculates the distribution of *angles* and *dihedrals* in time. It also gives the average angle or dihedral. The index file consists of triplets or quadruples of atom numbers:

```
[ angles ]
 1     2     3
 2     3     4
 3     4     5

[ dihedrals ]
 1     2     3     4
 2     3     5     5
```

For the dihedral angles you can use either the "biochemical convention" ($\phi = 0 \equiv cis$) or "polymer convention" ($\phi = 0 \equiv trans$), see Fig. 8.5.

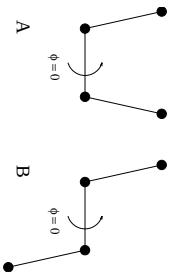

Figure 8.5: Dihedral conventions: A. “Biochemical convention”. B. “Polymer convention”.

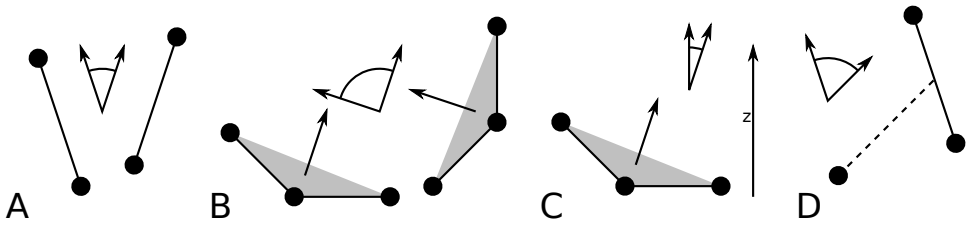

Figure 8.6: Angle options of `gmx gangle`: A. Angle between two vectors. B. Angle between two planes. C. Angle between a vector and the z axis. D. Angle between a vector and the normal of a sphere. Also other combinations are supported: planes and vectors can be used interchangeably.

The program `gmx gangle` provides a selection-enabled version to compute angles. This tool can also compute angles and dihedrals, but does not support all the options of `gmx angle`, such as autocorrelation or other time series analyses. In addition, it supports angles between two vectors, a vector and a plane, two planes (defined by 2 or 3 points, respectively), a vector/plane and the z axis, or a vector/plane and the normal of a sphere (determined by a single position). Also the angle between a vector/plane compared to its position in the first frame is supported. For planes, `gmx gangle` uses the normal vector perpendicular to the plane. See Fig. 8.6A, B, C) for the definitions.

8.8 Radius of gyration and distances

`gmx gyrate, gmx distance, gmx mindist, gmx mdmat, gmx xpm2ps`
To have a rough measure for the compactness of a structure, you can calculate the *radius of gyration* with the program `gmx gyrate` as follows:

$$R_g = \left(\frac{\sum_i \|\mathbf{r}_i\|^2 m_i}{\sum_i m_i} \right)^{\frac{1}{2}} \tag{8.19}$$

where m_i is the mass of atom i and $\mathbf{r}_i$ the position of atom i with respect to the center of mass of the molecule. It is especially useful to characterize polymer solutions and proteins.

Sometimes it is interesting to plot the *distance* between two atoms, or the *minimum* distance between two groups of atoms (*e.g.*: protein side-chains in a salt bridge). To calculate these distances between certain groups there are several possibilities:

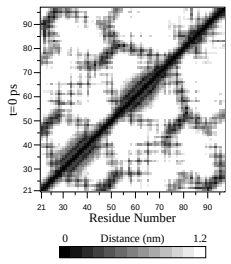

Figure 8.7: A minimum distance matrix for a peptide [157].

- The *distance between the geometrical centers* of two groups can be calculated with the program `gmx distance`, as explained in sec. 8.7.

- The *minimum distance* between two groups of atoms during time can be calculated with the program `gmx mindist`. It also calculates the *number of contacts* between these groups within a certain radius r_{max}.

- To monitor the *minimum distances between amino acid residues* within a (protein) molecule, you can use the program `gmx mdmat`. This minimum distance between two residues A_i and A_j is defined as the smallest distance between any pair of atoms (i $\in$ A_i, j $\in$ A_j). The output is a symmetrical matrix of smallest distances between all residues. To visualize this matrix, you can use a program such as `xv`. If you want to view the axes and legend or if you want to print the matrix, you can convert it with `xpm2ps` into a Postscript picture, see Fig. 8.7.

  Plotting these matrices for different time-frames, one can analyze changes in the structure, and *e.g.* forming of salt bridges.

8.9 Root mean square deviations in structure

`gmx rms, gmx rmsdist`
The *root mean square deviation* ($RMSD$) of certain atoms in a molecule with respect to a reference structure can be calculated with the program `gmx rms` by least-square fitting the structure

to the reference structure ($t_2 = 0$) and subsequently calculating the $RMSD$ (eqn. 8.20).

$$RMSD(t_1, t_2) = \left[\frac{1}{M} \sum_{i=1}^{N} m_i \|\mathbf{r}_i(t_1) - \mathbf{r}_i(t_2)\|^2 \right]^{\frac{1}{2}} \tag{8.20}$$

where $M = \sum_{i=1}^{N} m_i$ and $\mathbf{r}_i(t)$ is the position of atom i at time t. **Note** that fitting does not have to use the same atoms as the calculation of the $RMSD$; *e.g.* a protein is usually fitted on the backbone atoms (N,C$_\alpha$,C), but the $RMSD$ can be computed of the backbone or of the whole protein.

Instead of comparing the structures to the initial structure at time $t = 0$ (so for example a crystal structure), one can also calculate eqn. 8.20 with a structure at time $t_2 = t_1 - \tau$. This gives some insight in the mobility as a function of τ. A matrix can also be made with the $RMSD$ as a function of t_1 and t_2, which gives a nice graphical interpretation of a trajectory. If there are transitions in a trajectory, they will clearly show up in such a matrix.

Alternatively the $RMSD$ can be computed using a fit-free method with the program `gmx rmsdist`:

$$RMSD(t) = \left[\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \|\mathbf{r}_{ij}(t) - \mathbf{r}_{ij}(0)\|^2 \right]^{\frac{1}{2}} \tag{8.21}$$

where the *distance* $\mathbf{r}_{ij}$ between atoms at time t is compared with the distance between the same atoms at time 0.

8.10 Covariance analysis

Covariance analysis, also called principal component analysis or essential dynamics [158], can find correlated motions. It uses the covariance matrix C of the atomic coordinates:

$$C_{ij} = \left\langle M_{ii}^{\frac{1}{2}} (x_i - \langle x_i \rangle) M_{jj}^{\frac{1}{2}} (x_j - \langle x_j \rangle) \right\rangle \tag{8.22}$$

where M is a diagonal matrix containing the masses of the atoms (mass-weighted analysis) or the unit matrix (non-mass weighted analysis). C is a symmetric $3N \times 3N$ matrix, which can be diagonalized with an orthonormal transformation matrix R:

$$R^T C R = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{3N}) \quad \text{where} \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{3N} \tag{8.23}$$

The columns of R are the eigenvectors, also called principal or essential modes. R defines a transformation to a new coordinate system. The trajectory can be projected on the principal modes to give the principal components $p_i(t)$:

$$\mathbf{p}(t) = R^T M^{\frac{1}{2}} (\mathbf{x}(t) - \langle \mathbf{x} \rangle) \tag{8.24}$$

The eigenvalue λ_i is the mean square fluctuation of principal component i. The first few principal modes often describe collective, global motions in the system. The trajectory can be filtered along one (or more) principal modes. For one principal mode i this goes as follows:

$$\mathbf{x}^f(t) = \langle \mathbf{x} \rangle + M^{-\frac{1}{2}} R_{*i}\, p_i(t) \tag{8.25}$$

When the analysis is performed on a macromolecule, one often wants to remove the overall rotation and translation to look at the internal motion only. This can be achieved by least square fitting to a reference structure. Care has to be taken that the reference structure is representative for the ensemble, since the choice of reference structure influences the covariance matrix.

One should always check if the principal modes are well defined. If the first principal component resembles a half cosine and the second resembles a full cosine, you might be filtering noise (see below). A good way to check the relevance of the first few principal modes is to calculate the overlap of the sampling between the first and second half of the simulation. **Note** that this can only be done when the same reference structure is used for the two halves.

A good measure for the overlap has been defined in [159]. The elements of the covariance matrix are proportional to the square of the displacement, so we need to take the square root of the matrix to examine the extent of sampling. The square root can be calculated from the eigenvalues λ_i and the eigenvectors, which are the columns of the rotation matrix R. For a symmetric and diagonally-dominant matrix A of size $3N \times 3N$ the square root can be calculated as:

$$A^{\frac{1}{2}} = R\,\mathrm{diag}(\lambda_1^{\frac{1}{2}}, \lambda_2^{\frac{1}{2}}, \ldots, \lambda_{3N}^{\frac{1}{2}})\,R^T \tag{8.26}$$

It can be verified easily that the product of this matrix with itself gives A. Now we can define a difference d between covariance matrices A and B as follows:

$$\begin{aligned} d(A,B) &= \sqrt{\mathrm{tr}\left(\left(A^{\frac{1}{2}} - B^{\frac{1}{2}}\right)^2\right)} && (8.27)\\ &= \sqrt{\mathrm{tr}\left(A + B - 2A^{\frac{1}{2}}B^{\frac{1}{2}}\right)} && (8.28)\\ &= \left(\sum_{i=1}^{N}\left(\lambda_i^A + \lambda_i^B\right) - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\sqrt{\lambda_i^A\lambda_j^B}\left(R_i^A \cdot R_j^B\right)^2\right)^{\frac{1}{2}} && (8.29) \end{aligned}$$

where tr is the trace of a matrix. We can now define the overlap s as:

$$s(A,B) = 1 - \frac{d(A,B)}{\sqrt{\mathrm{tr}A + \mathrm{tr}B}} \tag{8.30}$$

The overlap is 1 if and only if matrices A and B are identical. It is 0 when the sampled subspaces are completely orthogonal.

A commonly-used measure is the subspace overlap of the first few eigenvectors of covariance matrices. The overlap of the subspace spanned by m orthonormal vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$ with a reference subspace spanned by n orthonormal vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ can be quantified as follows:

$$\mathrm{overlap}(\mathbf{v},\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{v}_i \cdot \mathbf{w}_j)^2 \tag{8.31}$$

The overlap will increase with increasing m and will be 1 when set $\mathbf{v}$ is a subspace of set $\mathbf{w}$. The disadvantage of this method is that it does not take the eigenvalues into account. All eigenvectors are weighted equally, and when degenerate subspaces are present (equal eigenvalues), the calculated overlap will be too low.

Another useful check is the cosine content. It has been proven that the the principal components of random diffusion are cosines with the number of periods equal to half the principal component index [160, 159]. The eigenvalues are proportional to the index to the power -2. The cosine content is defined as:

$$\frac{2}{T}\left(\int_0^T \cos\left(\frac{i\pi t}{T}\right) p_i(t)\mathrm{d}t\right)^2 \left(\int_0^T p_i^2(t)\mathrm{d}t\right)^{-1} \tag{8.32}$$

When the cosine content of the first few principal components is close to 1, the largest fluctuations are not connected with the potential, but with random diffusion.

The covariance matrix is built and diagonalized by `gmx covar`. The principal components and overlap (and many more things) can be plotted and analyzed with `gmx anaeig`. The cosine content can be calculated with `gmx analyze`.

8.11 Dihedral principal component analysis

`gmx angle, gmx covar, gmx anaeig`

Principal component analysis can be performed in dihedral space [161] using GROMACS. You start by defining the dihedral angles of interest in an index file, either using `gmx mk_angndx` or otherwise. Then you use the `gmx angle` program with the `-or` flag to produce a new `.trr` file containing the cosine and sine of each dihedral angle in two coordinates, respectively. That is, in the `.trr` file you will have a series of numbers corresponding to: cos(ϕ_1), sin(ϕ_1), cos(ϕ_2), sin(ϕ_2), ..., cos(ϕ_n), sin(ϕ_n), and the array is padded with zeros, if necessary. Then you can use this `.trr` file as input for the `gmx covar` program and perform principal component analysis as usual. For this to work you will need to generate a reference file (`.tpr`, `.gro`, `.pdb` etc.) containing the same number of "atoms" as the new `.trr` file, that is for n dihedrals you need $2n/3$ atoms (rounded up if not an integer number). You should use the `-nofit` option for `gmx covar` since the coordinates in the dummy reference file do not correspond in any way to the information in the `.trr` file. Analysis of the results is done using `gmx anaeig`.

8.12 Hydrogen bonds

`gmx hbond`

The program `gmx hbond` analyzes the *hydrogen bonds* (H-bonds) between all possible donors D and acceptors A. To determine if an H-bond exists, a geometrical criterion is used, see also Fig. 8.8:

$$\begin{array}{rcrcl} r & \leq & r_{HB} & = & 0.35 \text{ nm} \\ \alpha & \leq & \alpha_{HB} & = & 30^o \end{array} \tag{8.33}$$

The value of $r_{HB} = 0.35$ nm corresponds to the first minimum of the RDF of SPC water (see also Fig. 8.3).

The program `gmx hbond` analyzes all hydrogen bonds existing between two groups of atoms (which must be either identical or non-overlapping) or in specified donor-hydrogen-acceptor triplets, in the following ways:

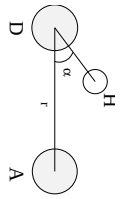

Figure 8.8: Geometrical Hydrogen bond criterion.

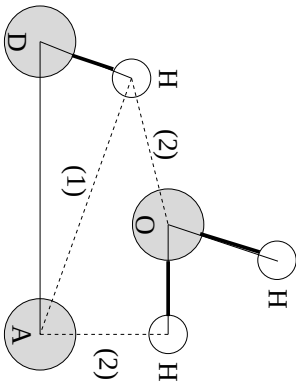

Figure 8.9: Insertion of water into an H-bond. (1) Normal H-bond between two residues. (2) H-bonding bridge via a water molecule.

- Donor-Acceptor distance (r) distribution of all H-bonds
- Hydrogen-Donor-Acceptor angle (α) distribution of all H-bonds
- The total number of H-bonds in each time frame
- The number of H-bonds in time between residues, divided into groups n-n+i where n and n+i stand for residue numbers and i goes from 0 to 6. The group for $i = 6$ also includes all H-bonds for $i > 6$. These groups include the n-n+3, n-n+4 and n-n+5 H-bonds, which provide a measure for the formation of α-helices or β-turns or strands.
- The lifetime of the H-bonds is calculated from the average over all autocorrelation functions of the existence functions (either 0 or 1) of all H-bonds:

$$C(\tau) \;=\; \langle s_i(t)\; s_i(t+\tau)\rangle \tag{8.34}$$

with $s_i(t) = \{0, 1\}$ for H-bond i at time t. The integral of $C(\tau)$ gives a rough estimate of the average H-bond lifetime τ_{HB}:

$$\tau_{HB} \;=\; \int_0^\infty C(\tau)d\tau \tag{8.35}$$

Both the integral and the complete autocorrelation function $C(\tau)$ will be output, so that more sophisticated analysis (*e.g.* using multi-exponential fits) can be used to get better estimates for τ_{HB}. A more complete analysis is given in ref. [162]; one of the more fancy option is the Luzar and Chandler analysis of hydrogen bond kinetics [163, 164].

- An H-bond existence map can be generated of dimensions *# H-bonds*×*# frames*. The ordering is identical to the index file (see below), but reversed, meaning that the last triplet in the index file corresponds to the first row of the existence map.
- Index groups are output containing the analyzed groups, all donor-hydrogen atom pairs and acceptor atoms in these groups, donor-hydrogen-acceptor triplets involved in hydrogen bonds between the analyzed groups and all solvent atoms involved in insertion.

8.13 Protein-related items

`gmx do_dssp, gmx rama, gmx wheel`
To analyze structural changes of a protein, you can calculate the radius of gyration or the minimum residue distances over time (see sec. 8.8), or calculate the RMSD (sec. 8.9).

You can also look at the changing of *secondary structure elements* during your run. For this, you can use the program `gmx do_dssp`, which is an interface for the commercial program `DSSP` [165]. For further information, see the `DSSP` manual. A typical output plot of `gmx do_dssp` is given in Fig. 8.10.

One other important analysis of proteins is the so-called *Ramachandran plot*. This is the projection of the structure on the two dihedral angles ϕ and ψ of the protein backbone, see Fig. 8.11.

To evaluate this Ramachandran plot you can use the program `gmx rama`. A typical output is given in Fig. 8.12.

When studying α-helices it is useful to have a *helical wheel* projection of your peptide, to see whether a peptide is amphipathic. This can be done using the `gmx wheel` program. Two examples are plotted in Fig. 8.13.

8.14 Interface-related items

`gmx order, gmx density, gmx potential, gmx traj`
When simulating molecules with long carbon tails, it can be interesting to calculate their average orientation. There are several flavors of order parameters, most of which are related. The program `gmx order` can calculate order parameters using the equation:

$$S_z = \frac{3}{2}\langle \cos^2 \theta_z \rangle - \frac{1}{2} \tag{8.36}$$

where θ_z is the angle between the z-axis of the simulation box and the molecular axis under consideration. The latter is defined as the vector from C_{n-1} to C_{n+1}. The parameters S_x and S_y are defined in the same way. The brackets imply averaging over time and molecules. Order parameters can vary between 1 (full order along the interface normal) and $-1/2$ (full order perpendicular to the normal), with a value of zero in the case of isotropic orientation.

The program can do two things for you. It can calculate the order parameter for each CH_2 segment separately, for any of three axes, or it can divide the box in slices and calculate the average value of the order parameter per segment in one slice. The first method gives an idea of the ordering of

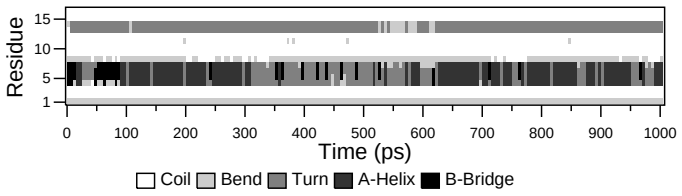

Figure 8.10: Analysis of the secondary structure elements of a peptide in time.

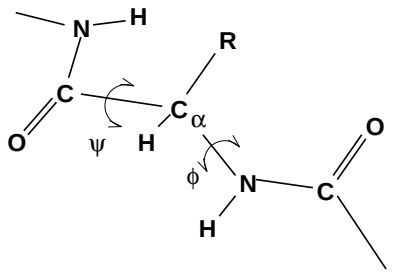

Figure 8.11: Definition of the dihedral angles ϕ and ψ of the protein backbone.

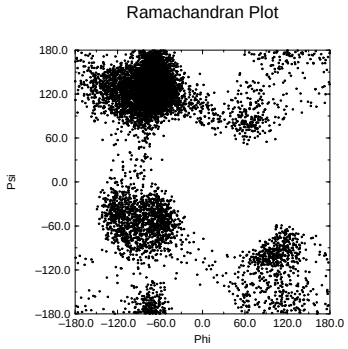

Figure 8.12: Ramachandran plot of a small protein.

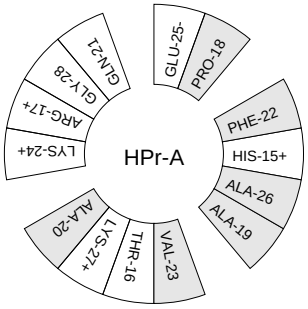

Figure 8.13: Helical wheel projection of the N-terminal helix of HPr.

a molecule from head to tail, the second method gives an idea of the ordering as function of the box length.

The electrostatic potential (ψ) across the interface can be computed from a trajectory by evaluating the double integral of the charge density ($\rho(z)$):

$$\psi(z) - \psi(-\infty) = -\int_{-\infty}^{z} dz' \int_{-\infty}^{z'} \rho(z'')dz''/\epsilon_0 \tag{8.37}$$

where the position $z = -\infty$ is far enough in the bulk phase such that the field is zero. With this method, it is possible to "split" the total potential into separate contributions from lipid and water molecules. The program `gmx potential` divides the box in slices and sums all charges of the atoms in each slice. It then integrates this charge density to give the electric field, which is in turn integrated to give the potential. Charge density, electric field, and potential are written to `xvgr` input files.

The program `gmx traj` is a very simple analysis program. All it does is print the coordinates, velocities, or forces of selected atoms. It can also calculate the center of mass of one or more molecules and print the coordinates of the center of mass to three files. By itself, this is probably not a very useful analysis, but having the coordinates of selected molecules or atoms can be very handy for further analysis, not only in interfacial systems.

The program `gmx density` calculates the mass density of groups and gives a plot of the density against a box axis. This is useful for looking at the distribution of groups or atoms across the interface.