Biophysical Molecular Dynamics Simulations

Michael Schmid* and Henri Menke†

(October 20, 2014)

The present experiment is dedicated to an introduction to visualisation and molecular dynamics of bio molecules. At first the structure of the prion protein, the hemoglobin and the streptavidin-biotin complex are to be visualised using the VMD (Visual Molecular Dynamics) tool. Second a huge simulation for the folding of the trpcage protein is performed using GROMACS. The last task is to find a hint on the tremendous stability of the streptavidin-biotin complex using a GROMACS simulation.

BASICS

Proteins

Proteins are biological macromolecules made up of aminoacids, see figure 1 for a sketch of a general aminoacid. Out of the vast number of aminoacids only 23 are relevant for protein synthesis. Those are called proteinogenic. Albeit their very limited amount of basic modules proteins can fulfil diverse functions. Instead of the constellation of modules the crucial property defining the function of a protein is its form. The count of possible foldings of a protein outnumbers the amount of particles in the universe.

Proteins take part in the transport and conversion of substances as catalytic elements. They can be categorised in three major groups, namely membranes, globular and fibrous [1].

Structure

Bonds not only ensure that proteins stay in their shape, but also determine their structure and size depending on the strength of the relevant bonds. The form of a protein can be described roughly by the following levels of structure:

Primary structure: The carboxyl group (COOH) of one aminoacid can react with the amino group (H_2N) of another aminoacid and form a peptide bond as depicted in figure 2. Aminoacids are called bi-functional because they contain both groups and can hence form long linear polymer chains.

Aminoacids are the “residues” and peptide bonds the “backbone” of a polymer.

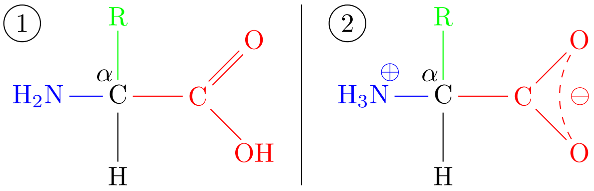

FIG. 1. General structure formula of an α-aminoacid in (1) its un-ionized state and in (2) its zwitterionic state. This form includes a rest R, a carboxyl group COOH and a hydrogenatom H bound to the central C_α.

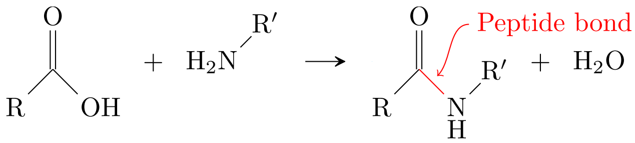

FIG. 2. Formation of a peptide bond by condensation of a carboxyl group and an amino group. The bond is covalent and hence has a very high binding energy. An aminoacid has both of these group which favours the formation of long-chain polymers.

Secondary structure: This describes the local order of the backbone. Two major structures can be observed: α-helix and β pleated sheet. Both are stabilised by hydrogen bonds between the NH and CO groups of the main chain.

Tertiary structure: Under influence of external factors such as water a protein reveals some functionality of its side chains. Some of them might be hydrophilic or hydrophobic. Thus a protein will choose another ordering when placed in an aqueous regime, namely that the hydrophilic parts stay close to water and the hydrophobic parts form some sort of excluded volume.

Quaternary structure: Proteins with more than one polypeptide chain (subunit) might arrange each of those subunits differently in order to fulfil different functionality.

Protein folding

We already got to know the different levels of structure of a protein. Protein folding is the process of obtaining a higher level structure by changing the shape of the overall polymer. In the first section we stated that there are more possible foldings than particles in the universe. Because the process takes place in nature in finite time it has to be non-ergodic and deterministic, i.e. not all possible foldings are tested and the outcome of one folding with fixed initial conditions is always the same.

Protein folding is a non linear process which can easily be perturbed. Little pertubation already causes the protein to fold in a completely different way. Thus it is really hard to find the correct numerical initial conditions to actually simulate a protein folding and we need to fall back on experimental methods.

Structural analysis

The primary structure of polymers (e.g. DNA or in our case proteins) can be determined by so called sequencing. That is, the protein is cut down to its basic aminoacids step by step.

Higher order structures are determined by spectral methods (X-ray diffraction or nuclear magentic resonance (NMR). The structures obtained are then uploaded to the protein database (PDB) which will be used later.

Examples

In the remainder of this experiment we might be confronted with several proteins. Here a few examples given by Sivaraman [1] will be presented.

Prion: This protein comes in two forms. One with α-helical structure which is found in the human body and one with numerous β sheets which causes several neurodegenerative diseases. The human prion protein consists of 209 residues.

Hemoglobin: The red blood colourant acts as an oxygen carrier in the human circulatory system. The oxygen carriage is performed by a helper group called heme with a central iron atom.

Streptavidin-biotin complex: This is, as the name already suggests, a complex of the streptavidin protein and the biotin molecule. Both components are bound by a hydrogen bond, the strongest non-covalent bond.

Trpcage: This protein is relatively small with only 20 residues and fold comparatively fast and is thus used in this lab to simulate protein folding.

Molecular Dynamics Simulation

To describe the motion of a particle in a completely correct way the Schrödinger equation needs to be solved. This is not possible as the Hilbert space for a particle is infinitely dimensional and can only be diagonalised if constrained by special boundary conditions [2].

Thus we apply the Born-Oppenheimer approximation and fall back to classical mechanics. There we can get the trajectory of a particle by integrating Newton's equations of motion

$$m \frac{\mathrm{d}^2}{\mathrm{d}\tau^2} x^\mu = F^\mu \tag{1}$$

in its covariant form with the proper time τ.

To suppress finite size effects periodic boundary conditions (PBC) are applied.

Force fields

Force fields are used to model several components of interaction potential between the particles. Because not much is known about the actual mechanism of those potential they are approximated. A covalent bond for example is replicated by a bead-spring model where the atoms are connected by "springs". Some quantities need to be fixed for this approximation, like the (mean) bond length, the bond angle and the torsion angle. Additionally the bond is not the only potential between two particles. Others are van-der-Waals and coulomb interaction.

The actual combination of these various terms depends on the force field used for the simulation. Every force field differs in some way. In our experiment an AMBER force field was used. This is an all-atom force field, i.e., it defines interaction parameters for single atoms, whereas the GROMOS force field is coarse grained in a way that it defines interaction parameters only for beads containing several atoms, such as CH_4.

A polymer will never start off in the minimum of such a potential because, as we already stated, the force fields are only an approximation. To avoid diverging forces between the particles an equilibration procedure is necessary. This is usually done using a steepest descent method.

Integrators

Equation (1) can only be solved analytically for a very limited number of cases. For all other cases a numerical approach is needed. Because statistical physics always involves fluctuation to generate a non-zero temperature the real trajectory of the particle is not relevant and an integrator with long-term stability can be used [3].

The maybe most used algorithm to fulfil the previously mentioned conditions is the velocity verlet integrator. It is a symplectic, energy conserving, fourth-order integrator. The verlet scheme is illustrated in figure 3. One possible implementation of a verlet like integrator is the leap frog

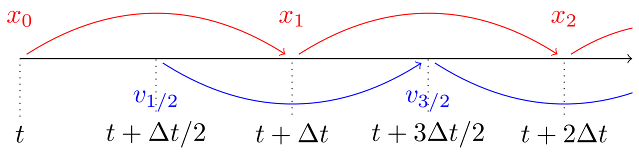

FIG. 3. The verlet scheme. The positions are updated every step, the velocities are updated every half step. It can be shown that using this order one can ensure energy conservation.

integrator [3]

$$\begin{aligned} \boldsymbol{v}(t+\Delta t/2) &= \boldsymbol{v}(t-\Delta t/2) + \frac{\Delta t}{m}\boldsymbol{F}(t), \\ \boldsymbol{r}(t+\Delta t) &= \boldsymbol{r}(t) + \Delta t \cdot \boldsymbol{v}(t+\Delta t/2). \end{aligned} \tag{2}$$

Thermostat and Barostat

On a microscopic scale the temperature of a local process is normally prescribed by its surroundings. Hence the temperature in our simulation the temperature needs to be kept fixed. In the human body there are (hopefully) no major pressure fluctuations, hence the pressure will also be kept fixed. Additionally we fix the number of particles and obtain the canonical or NPT ensemble [3].

To fix the temperature in our simulation we apply the Nosé-Hoover-Thermostat. Here an additional friction is added on the right-hand side of the equations of motion depending on the friction parameter ξ. The transformed equations read

$$\begin{aligned} \frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} &= \frac{\boldsymbol{F}_i}{m_i} - \frac{p_\xi}{m}\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}, \\ \frac{\mathrm{d}p_\xi}{\mathrm{d}t} &= T - T_0, \end{aligned} \tag{3}$$

where p_ξ is the momentum of the friction parameter, Q is the coupling constant of the heat bath to the system, T_0 is the target temperature of the system and T is the current temperature.

For constant pressure the Parinello-Rahman approach is employed. In this scheme the volume is an intensive variable and depends on the pressure. This is achieved by allowing the box to rescale dynamically. The equation of motion for the box vector $\boldsymbol{b}$ reads

$$\frac{\mathrm{d}^2\boldsymbol{b}}{\mathrm{d}t^2} = VW^{-1}\boldsymbol{b}'^{-1}(\boldsymbol{P} - \boldsymbol{P}_0) \tag{4}$$

with the variable volume V and the coupling constant W. The matrices $\boldsymbol{P}$ and $\boldsymbol{P}_0$ represent the current and the target pressure, respectively.

Equilibration

As mentioned in the previous text, to avoid diverging forces we need to perform some kind of equilibration procedure. A finished equilibration can be detected if *all* the observables only fluctuate around their mean value. Only when the system is in equilibrium the observable may actually be measured because before thermodynamics don't apply.

From our simulation only the positions and veolcity of the particles are accessible. We need to calculate the observables from these given quantities. The temperature can be obtained by means of the equipartition theorem

$$\frac{f}{2}Nk_\mathrm{B}T = \frac{1}{2}\sum_i m_i \left\langle v_i^2 \right\rangle, \tag{5}$$

where the number of degrees of freedom f is usually 3 (translational).

An expression for the pressure can be derived from the virial theorem $\langle E_\mathrm{tot}\rangle = \langle E_\mathrm{int}\rangle + \langle E_\mathrm{ext}\rangle$.

$$-3Nk_\mathrm{B}T = \left\langle \sum_i m_i \boldsymbol{f}_i \cdot \boldsymbol{r}_i \right\rangle - 3PV. \tag{6}$$

Software

Visual Molecular Dynamics (VMD): VMD is a computer programme dedicated to the visualisation and animation of even large molecular structures. In later parts of this paper we used VMD to create some fancy pictures of different polymers [4].

GROMACS: The name of the software package GROMACS stands for GROningen MAchine for Chemical Simulations. It is a very user friendly package of various algorithms for mainly all-atom simulations. It is optimised for parallel computing [5].

ANALYSIS

All the structures relevant for the subsequent tasks were obtained from the protein database (PDB) at

`http://www.pdb.org`

The structures downloaded are

- the prion protein `1DX0`,
- the hemoglobin `3S66`,
- the trpcage `1L2Y`,
- the streptavidin-biotin complex `1STP`.

Listing 1. Shortened excerpt of the 1DX0 PDB file. One can see the three α helices and the two β sheets.

```
HELIX 1   H1 ASP A   144   GLU A   152   1 ...
HELIX 2   H2 ASN A   173   THR A   192   1 ...
HELIX 3   H3 GLU A   200   TYR A   225   1 ...
SHEET 1    A 2 TYR A 128   GLY A 131   0 ...
SHEET 2    A 2 VAL A 161   ARG A 164  -1 ...
```

The Protein Database

For the prion protein there are different entries in the protein database with different structure. This is due to the fact that the prion protein is found in various organisms with an optimised structure for each case. Also, as mentioned before, there are several ways of determining the shape of a polymer (X-ray diffration, NMR) which produce different results. Furthermore there a two forms of the prion protein, one normally folded and one misfolded which differ tremendously.

From the PDB we can see that the prion protein consists of three α helices and two antiparallel β sheets. Listing 1 shows an excerpt of the prion protein PDB file. The lines starting with HELIX denote the α helices, obviously and the lines with SHEET denote the β sheets. These keywords are followed by a sequence of three letter words which are the aminoacids forming the corresponding structure.

Scanning further through the file revealed that the disuflid-bond is placed between the aminoacids CYS 179 and CYS 214 of the second and third α helix.

Visualisation

Prion: Starting off with the prion, the task was to visualise the structure emphasising the disulfide bond. Figure 4 shows the structure with the three α-helices (coils) and the two β sheets (arrows). The two residues CYS 179 and CYS 214 involved in the disulfide bond are emphasised using the "CPK" representation clearly showing the bond between the two sulfur atoms (yellow). As the bond connects two of the α helices it defines the tertiary structure, because it will constrain the two helices in their movement under impact of an external force.

Hemoglobin: Next is hemoglobin for which we chose the PDB structure 3S66. The hemoglobin consists of four chains, each of them built up from several α helices. The secondary structure is determined by the disc-shaped heme groups connecting the chains. Because there are four heme groups (one for each chain) the protein can bind four oxygen molecules. The image is displayed in figure 5.

Streptavidin-Biotin Complex: Last in line is the streptavidin-biotin complex. It consists of a biotin molecule embedded in a streptavidin polymer. The polymer contains one α helix and several β sheets which are wrapped around the biotin molecule in its centre. The polymer is connected to the biotin via hydrogen bonds. A quaternary structure could be formed by moving the biotin back and forth inside the polymer "tube".

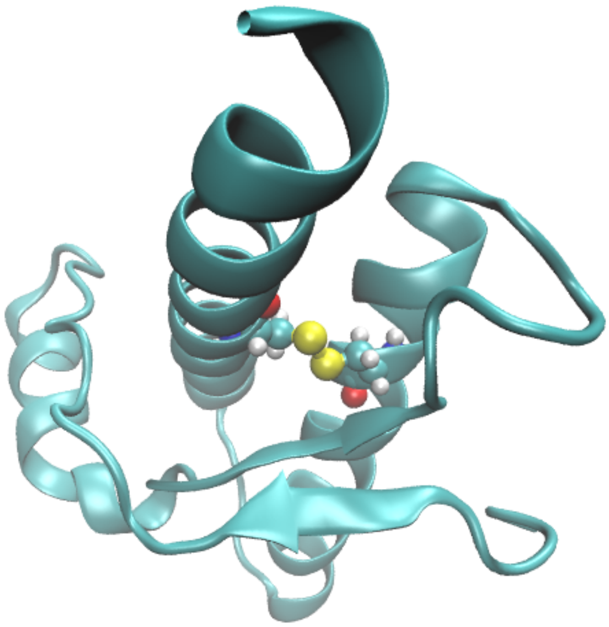

FIG. 4. Visualisation of the prion protein using VMD. The disulfide bond is represented using "CPK", the rest using "New Cartoon".

Protein Folding

Note bene: In the following we used the updated syntax of GROMACS 5.0, which differs from the old syntax in the way that every command is preceded by gmx and that genbox was renamed to solvate. Furthermore if any units are used, then lengths are always in nanometres and times in picoseconds (GROMACS default). Also the files ions.mdp, md.mdp and unfoldedTPR.gro were already given.

Using the package GROMACS the folding of the trpcage protein should be simulated. The structure 1L2Y was already downloaded before. First of all the topology for the simulation had to be created from the PDB file using a force field and by adding water to the system. This was done using the pdb2gmx command.

```
gmx pdb2gmx -p topol.top -f 1L2Y.pdb -o
   ↪ conf.gro -ignh -ff amber99sb-ildn
   ↪ -water tip3p
```

This reads the PDB file given by the flag -f 1L2Y.pdb and outputs a GROMACS readable configuration with -o conf.gro and a topology by means of -p topol.gro.

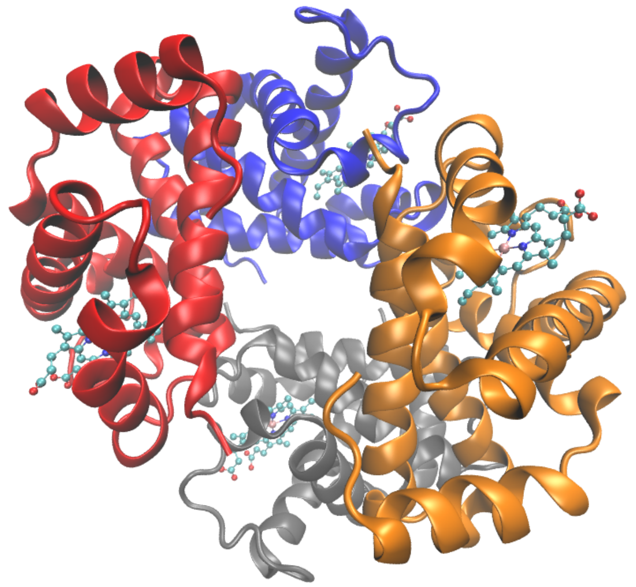

FIG. 5. Visualisation of hemoglobin using VMD. The HEM groups have been emphasised using the "CPK" representation. The several chains have been coloured differently and represented by "New Cartoon".

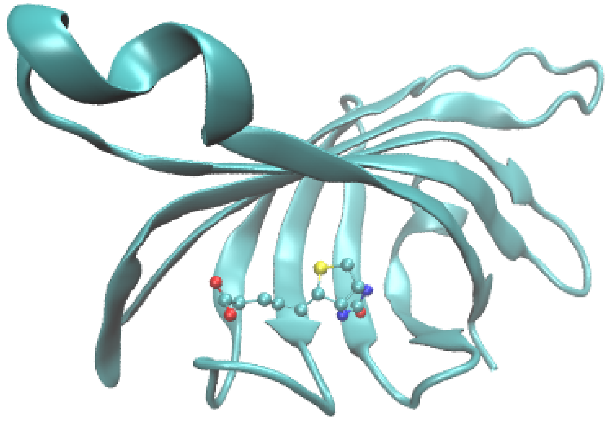

FIG. 6. Visualisation of the strepakldhaklhrel-biotin complex using VMD. The tiny biotin inside the huge streptavidin protein was drawn using "CPK", the protein using "New Cartoon".

The topology file ending with `.top` contains information on the systems topology, i.e. interaction parameters, involved molecules, solvent, et cetera. In the above command the AMBER99 force field is chosen using the `-ff amber99sb-ildn` flag and the TIP3P water is added by `-water tip3p`. The `-ignh` option instructs `pdb2gmx` to ignore the hydrogen atoms in the protein for the creation of the topology. When executing this command a warning is issued that the system has non-zero charge (in our case $+1e$.

Then the file `md.mdp` is adjusted according to the manual. The following, initially void entries, are filled with appropriate values

```
integrator  = sd
dt          = 0.002
nsteps      = 150000000
...
tau_t       = 2
ref_t       = 300
...
gen-vel     = yes
gen-temp    = 300
```

We use the stochastic leap-frog integrator `sd` with a time step `dt` of 2 fs at a total runtime of 300 ns which corresponds to 150 000 000 steps given as `nsteps`. The system will be thermalised to `ref_t` $= 300$ K with a coupling time of `tau_t` $= 2$ ps. The initial velocities are generated according to a Maxwell-Boltzmann distribution by setting `gen-vel = yes` with a temperature of 300 K by using `gen-temp = 300`.

Next the protein is placed in a cubic box of 4.5 nm per edge using the utility `editconf`.

```
gmx editconf -f conf.gro -o out.gro -c -bt
    ↪ cubic -box 4.5 4.5 4.5 -d 1.0
```

This command line is—like most the others—pretty self explanatory. The first parameter `-f conf.gro` selects the configuration, `-c` centres the protein in the box, `-bt cubic` demands the box to be cubic, `-box 4.5 4.5 4.5` sets all edges to a length of 4.5 nm and finally `-d 1.0` ensures that the protein is aligned in the box such that its surface has at least a distance of 1.0 nm to the box faces.

Before we can neutralise the system which has still $+1e$ charge we need to add water to the box because GROMACS can only replace particles. We use `solvate`:

```
gmx solvate -p topol.top -cp out.gro -cs
    ↪ spc216.gro -o pro_water.gro
```

The configuration and the topology are updated through `-p topol.top -cp out.gro`, water is loaded using `-cs spc216.gro` and the result is written out by `-o pro_water.gro`. Before we can continue the file needs to be processed by `grompp`.

```
gmx grompp -p topol.top -f ions.mdp -c
    ↪ pro_water.gro -o ions.tpr
```

We already input the file `ions.mdp` which is not relevant here but is demanded by GROMACS. The resulting topology is written to `ions.tpr`.

We neutralise the system by replacing one atom of the species `SOL` with an anion of type `CL` which is a chloride ion of $-1e$ charge—obviously.

```
gmx genion -p topol.top -s ions.tpr -o
    ↪ pro_ions.gro -nname CL -nn 1 <<< 13
```

This updates the topology and gives us a new system setup in `pro_ions.gro`. This file needs to be preprocessed again:

```
gmx grompp -p topol.top -f ions.mdp -c
    ↪ pro_ions.gro -o neutr.tpr
```

Still we can't start the energy minimisation because as we don't use a molecular dynamics integrator we can't simulate explicit water and need to remove that first.

We simply remove the water by calling

```
gmx trjconv -f pro_water.gro -s neutr.tpr -
    ↪ o pro_nowater.gro <<< 14
```

which removes all water species. Unfortunately the `trjconv` programme doesn't update the topology so we needed to remove the solute from the `topol.top` semiautomatically using UNIX-`sed`:

```
sed -i '/^SOL/d' topol.top
```

Now a last preprocessor run is required to update all files before starting the energy minimisation.

```
gmx grompp -p topol.top -f ions.mdp -c
    ↪ pro_nowater.gro -o em.tpr
```

Here it is actually important to input `ions.mdp` because this is the run parameters file for `mdrun`.

It's time to carry out the actual energy minimisation for the neutralised system.

```
gmx mdrun -v -deffnm em -ntomp 8
```

This step is necessary because we need to equilibrate the simulation before starting the actual folding. Else all measured observables have no meaning because the basic principles of thermodynamics only apply for equilibrium systems.

After that we input the parameter file `unfoldedTRP.gro` which was already given to setup the folding process. Afterwards we start the folding process.

```
gmx grompp -p topol.top -f md.mdp -po mdout
    ↪ .mdp -c unfoldedTRP.gro -o sim.tpr
gmx mdrun -v -deffnm sim -ntomp 8
```

This will take a while...

After about 9 hours of runtime the simulation was finished. Using the GROMACS helper tool `g_energy` the potential energy was extracted from the energies file produced by the simulation.

```
g_energy -f sim.edr -s sim.tpr -o energy.
    ↪ xvg <<< 10
```

The tool `xmgrace` is used to compute the running average of the potential energy. The results are displayed in figure 7. In the subfigure (a) the potential energy is plotted together with its running average. A fit to the running average reveals a slope of $-4.138 \cdot 10^{-5}$ which is tiny but still negative and hence indicates a minimisation of the potential energy which is expected. In (b) the RMSD is plotted. There are two kinks in the curve which could be possible folding events. They are marked with green circles.

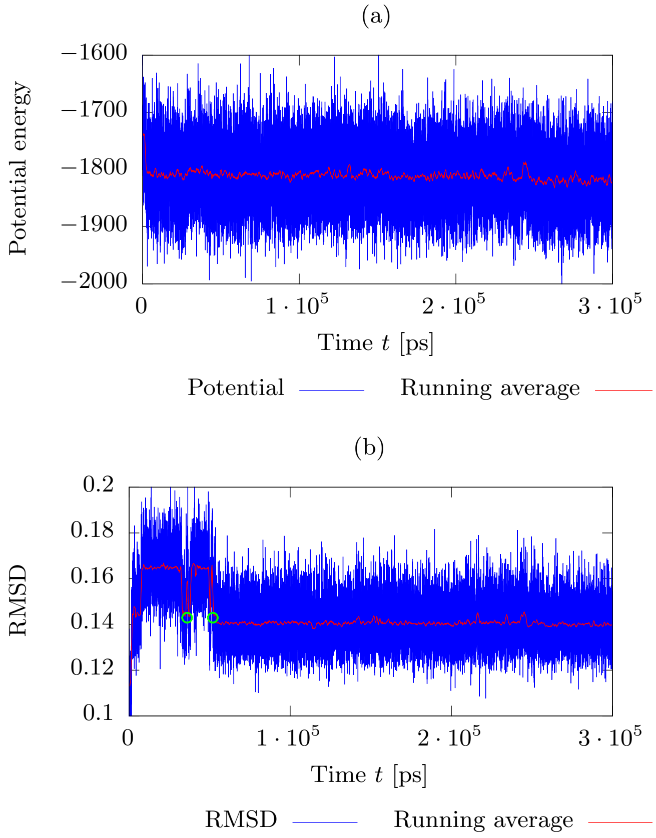

FIG. 7. Results of the simulation of the trpcage folding. The red lines represent the running averages of the respective quantities over 1000 samples. (a) The potential energy of the system. (b) The RMSD of the backbone of the protein.

During the simulation a `.gro` file was produced which contains information about the structure of the trpcage protein. Together with the `.xtc` file a movie of the folding can be produced. The movie won't be very interesting because one would mainly see fluctuations emergent from the thermalisation. Still, a picture of the final structure can be obtained from this data. The protein after 1 ns of folding simulation is displayed in figure 8

Streptavidin-Biotin-Complex

In this part the streptavidin-biotin-complex was simulated in water. The given simulation scripts were extended to use the AMBER99SB force field and the TIP3P water model. The given scripts were pretty complete up to these two things. Some minor adjustments such as setting the box size had to be done. Also some run parameters had to be filled in to ensure an NPT ensemble.

Then `acpype` and `openbabel` took care of extracting the topology from the PDB file. The output of the script was the energy minimised structure. The potential en-

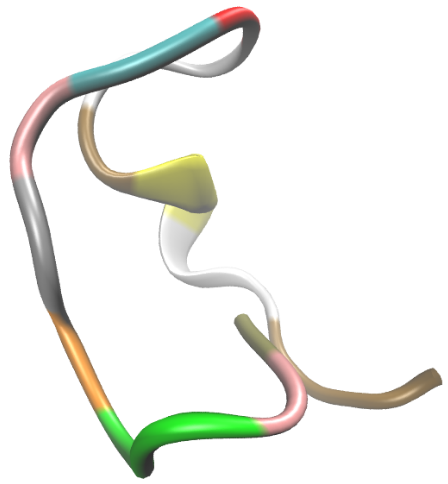

FIG. 8. The structure of the trpcage protein after 300 ns of simulation. The final state is not the preferred folded state.

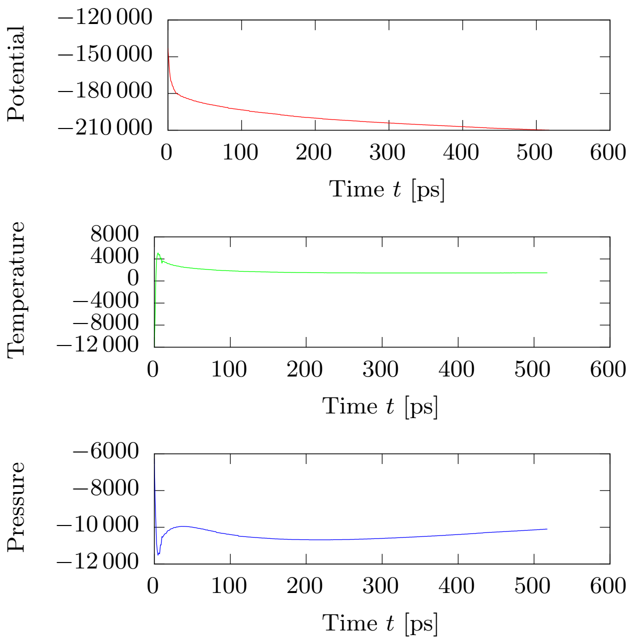

FIG. 9. Minimisation of the potential energy before starting the simulation of the streptavidin-biotin-complex. The potential energy decays and the other observables (temperature,pressure) converge to a nearly constant value.

ergy, the temperature and the pressure were analysed using g_energy to see the minimisation behaviour of those observables.

```
g_energy -f em.edr -s em.tpr -o em.xvg <<<
    ↪ "13 14 15 0"
```

This is not the actual equilibration run but rather a form of "force capping" to avoid starting too far away from the minimum and to circumvent diverging energies. A graph of the minimisation is shown in figure 10.

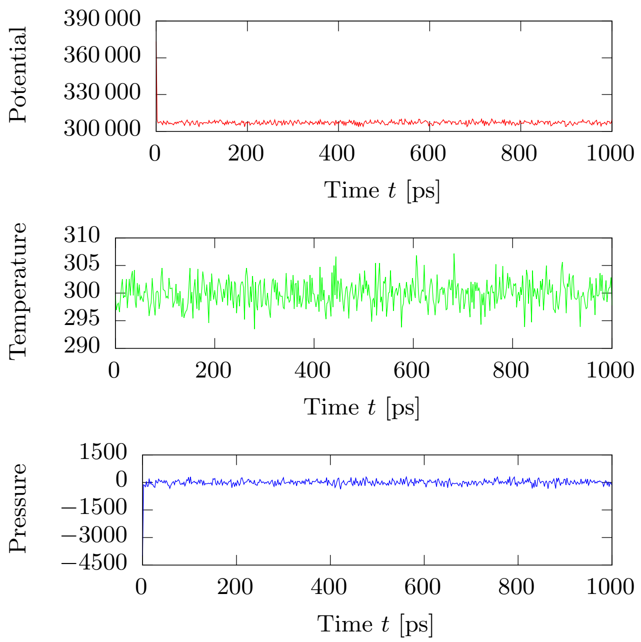

FIG. 10. Equilibration and simulation run of the behaviour of the streptavidin-biotin-complex in water. One can see that all observables equilibrate *really* quick, as in the pictures on can only see them fluctuating around their mean value.

Next the equilibration and the dynamics of the system is simulated which will hopefully result in a change of the structure of complex. The parameter file `equi.mdp` was used for this run. Afterwards the same observables as before were analysed again. The results can be found in figure 10. The equilibration is done after a few timesteps which can only be derived from the graphs due to the fact that the plot range is much larger than the visible data. This indicates a decay in the first pico seconds. The kinetic energy behaves exactly like the temperature because they are directly connected by the equipartition theorem.

The change in structure was visualised using VMD. The structure before and after the simulation are displayed in figure 11. The hydrogen bonds in these pictures are marked by red and blue coils. One can see that during the simulation the position of the biotin inside the streptavidin changed slightly, especially in a rotational way. It is visible that the hydrogen bonds between the two molecules are present in both states. Hence these hydrogen bond are very strongly binding.

SUMMARY

Visualisation: The visualisation of the prion protein clearly revealed the disulfide bond. Its capability of connecting the two α helices is crucial to the tertiary structure of the overall molecule. The hemoglobin was found to have four α helices, each of them connected to a heme disc. The heme discs are the components of the hemoglobin responsible for the oxygen transport. The streptavidin-biotin-complex consists of a biotin molecule "embraced" by the β sheets of the streptomycin.

Protein Folding: The potential energy and the RMSD of the protein backbone were extracted from a folding simulation of 1 ns. A correlation between kinks in the RMSD and the potential energy were expected but not observed. The RMSD showed two major kinks which could be interpreted as the actual folding events but could not be verified due to absence of a sudden change of potential energy. The simulation didn't result in a complete folding. Possible reasons could be erroneous input structure files.

Stability of Streptavidin-Biotin: Long-life hydrogen bonds in the streptavidin-biotin-complex are responsible for its stability. The long-term presence of these could be verified in a simulation of the complex in explicit water. As a result the biotin embedded in the streptavidin moved hardly during the run.

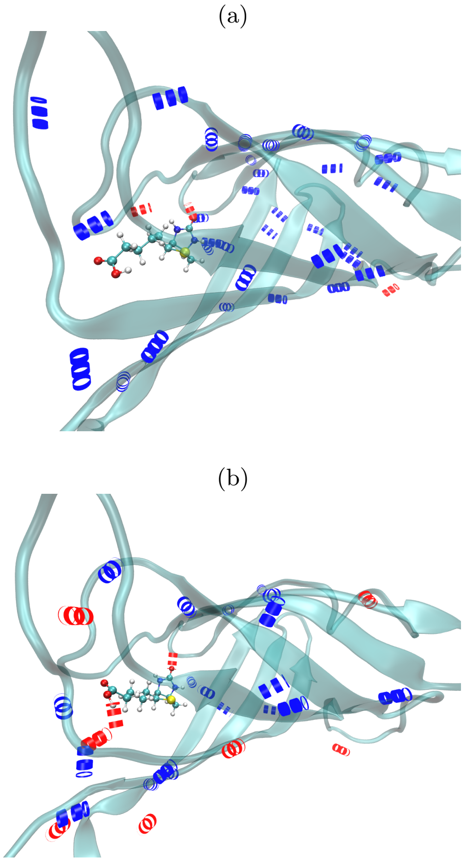

FIG. 11. The two structures of the streptavidin-biotin-complex at (a) the beginning of the simulation and (b) at the end.

* Michael_1233@gmx.de
† henrimenke@gmail.com

[1] G. Sivaraman, *Basics of Computational Biophysics* (2013).

[2] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Vol. 1 (Academic press, 2001).

[3] A. Arnold, *Physik auf dem Computer*, 2nd ed. (Universität Stuttgart, 2013).

[4] W. Humphrey, A. Dalke, and K. Schulten, Journal of Molecular Graphics **14**, 33 (1996).

[5] H. Bekker, H. J. C. Berendsen, E. J. Dijkstra, S. Achterop, R. van Drunen, D. van der Spoel, A. Sijbers, H. Keegstra, B. Reitsma, and M. K. R. Renardus, in *Physics Computing*, Vol. 92 (1993) pp. 252–256.

[6] F. Dommert, *FP Praktikum 2013 – Classical Molecular Dynamics* (2013).