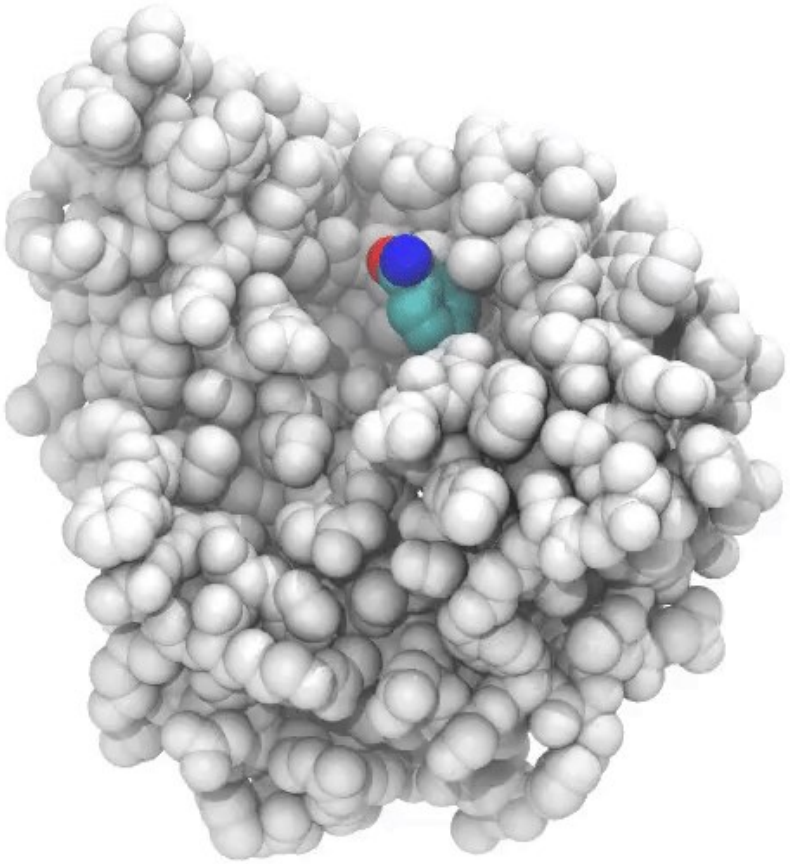
Structural bioinformatics

Practice 7: Molecular dynamics set up

Course 2023-2024

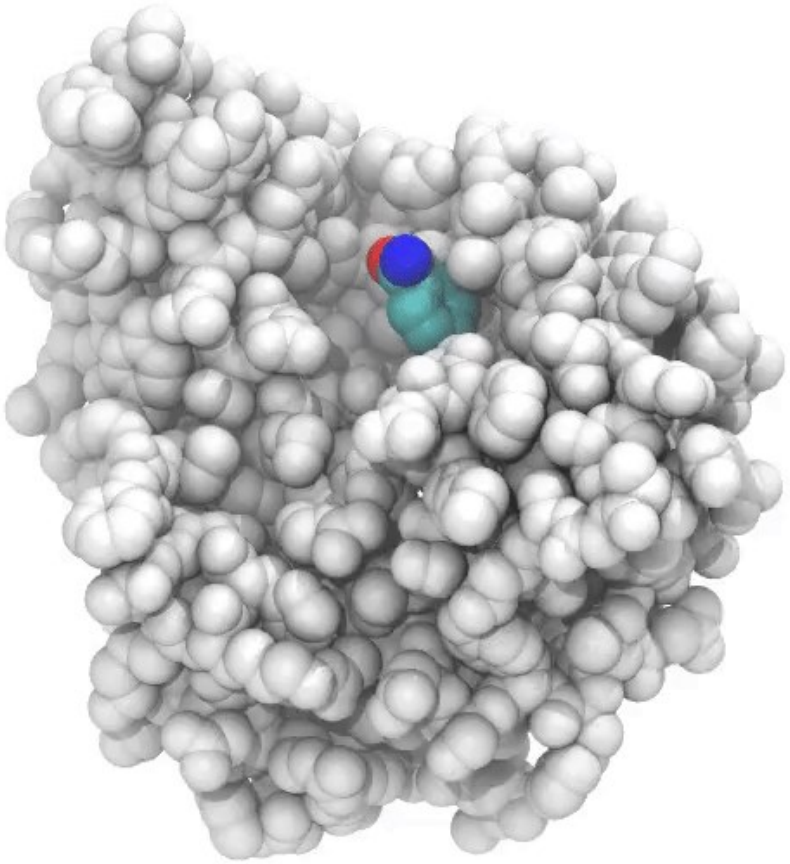
Molecular dynamics

- Molecular dynamics allows us to simulate a chemical system at an atomic level.
- The system is propagated forward in time using Newton's equations of motion.
- Forces acting upon atoms are obtained using classical mechanics force-fields.



Molecular dynamics

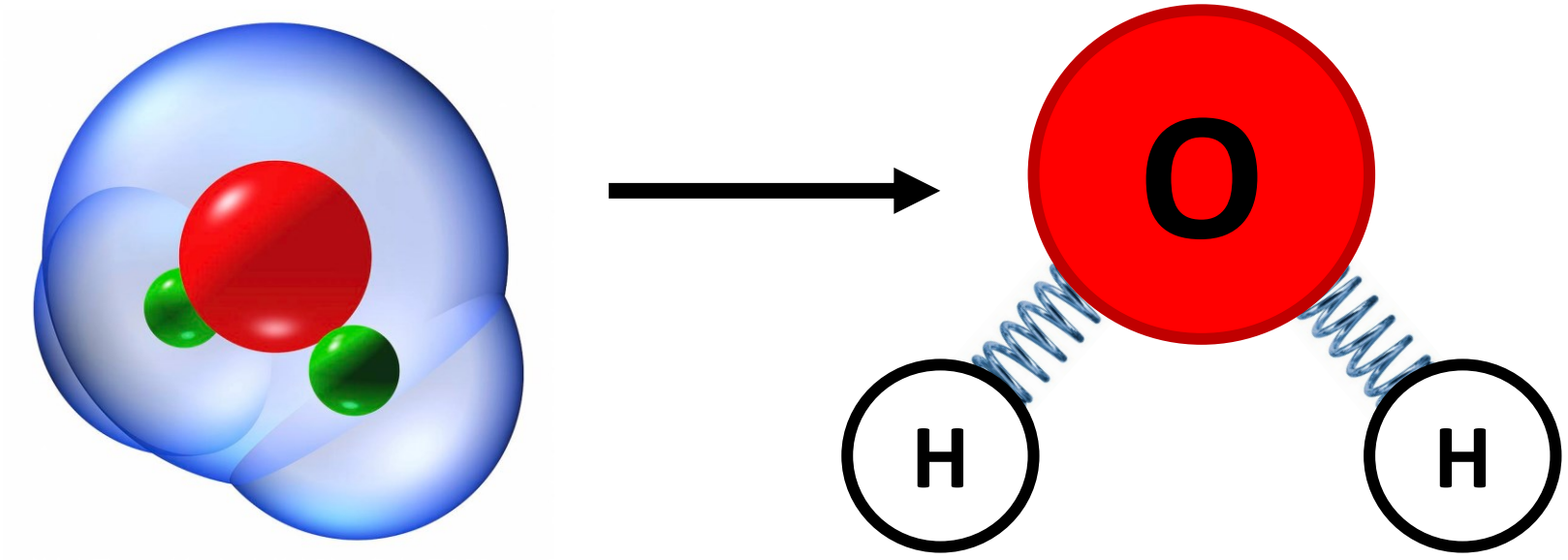
- Molecular dynamics allows us to simulate a chemical system at an atomic level.
- The system is propagated forward in time using **Newton's equations** of motion.
- Forces acting upon atoms are obtained using **classical mechanics** force-fields.



Classical mechanics

Classical mechanics apply Newton's equations on the atoms that we are simulating

This actually means that we represent atoms like solid balls and bonds like elastic springs



Force fields

A force field is a program that describes the energy landscape for a chemical system

The force field allows you to estimate the potential energy of the system that you are analyzing.

In this tutorial we will use the AMBER parm99 force field, this force field accounts for:

```
Energies (kJ/mol)
        Bond            Angle    Proper Dih.  Improper Dih.          LJ-14
 8.13493e+03    1.83442e+03    5.07958e+03    2.57149e+02    2.42157e+03
  Coulomb-14        LJ (SR)   Coulomb (SR)   Coul. recip. Position Rest.
 1.75029e+04    1.81398e+05   -5.81275e+05    1.73394e+04    1.84602e-01
   Potential Pressure (bar)
-3.47306e+05   -2.24263e+04
```

Molecular dynamics engine

MD engines are programs that use the energies described by the force field to calculate forces and accelerations of atoms

It is the molecular dynamics engine the one who moves the atoms from frame to frame.

In this tutorial we will use the GROMACS molecular dynamics engine

GROMACS

fast, flexible & free

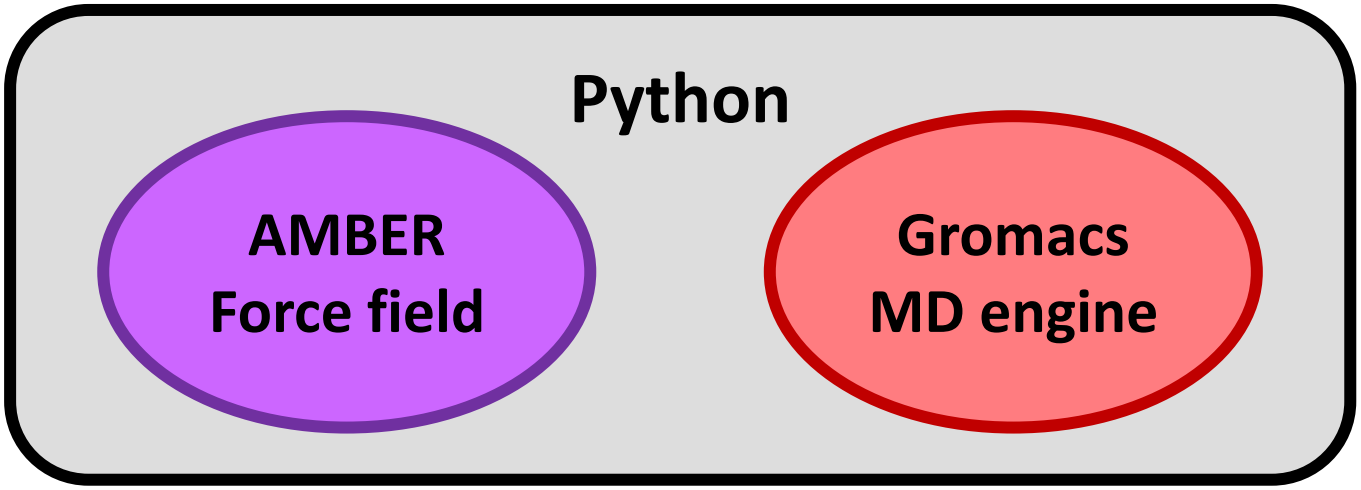
Molecular dynamics set up

In today's tutorial we will set up a system for making a simulation

We will follow this tutorial:

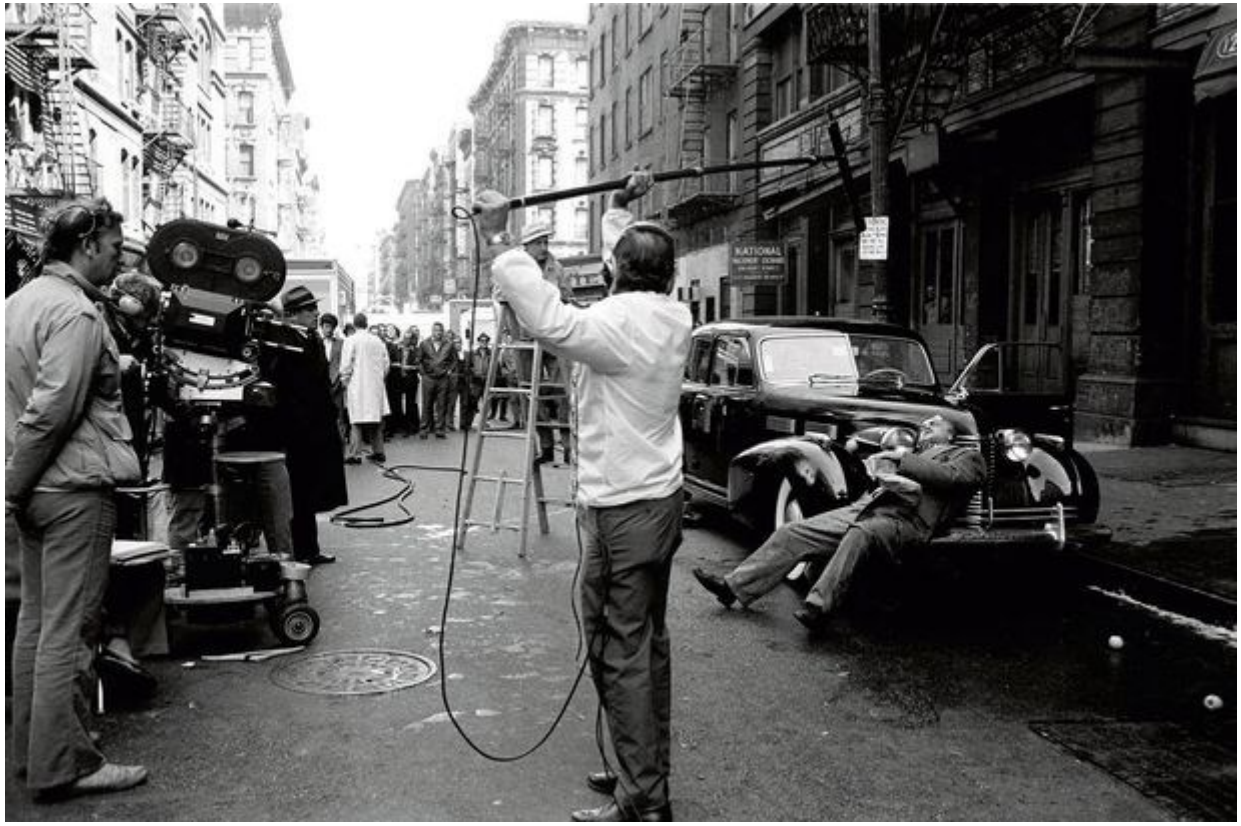
https://mmb.irbbarcelona.org/biobb/workflows/tutorials/md_setup

biobb library



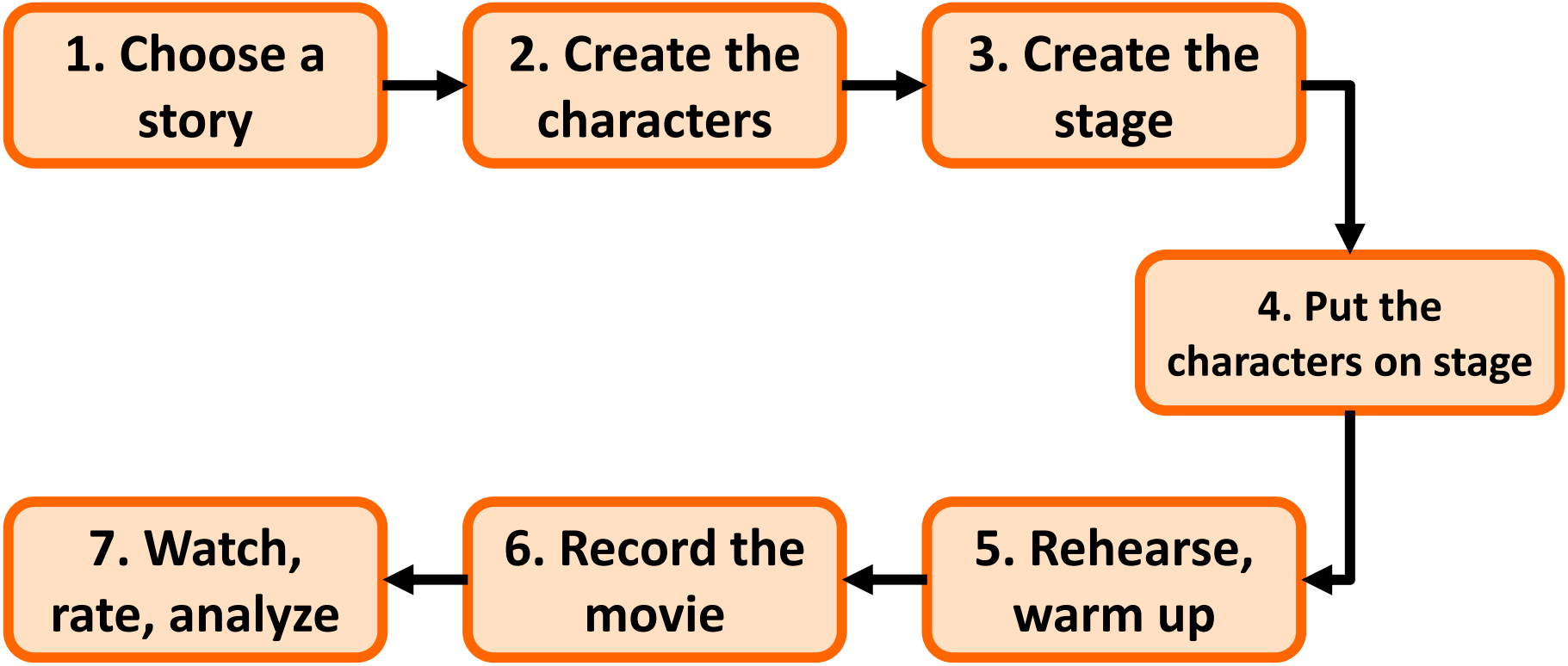
Molecular dynamics set up

Making a molecular dynamics simulation is similar to making a movie



Molecular dynamics set up

Making a molecular dynamics simulation is similar to making a movie



0. Install and load required packages

Conda Installation and Launch

```
git clone https://github.com/bioexcel/biobb_wf_md_setup.git
cd biobb_wf_md_setup
conda env create -f conda_env/environment.yml
conda activate biobb_MDsetup_tutorial
jupyter-nbextension enable --py --user widgetsnbextension
jupyter-nbextension enable --py --user nglview
jupyter-notebook biobb_wf_md_setup/notebooks/biobb_MDsetup_tutorial.ipynb
```

1. Fetch a PDB

1. Choose a story

1. Fetch a PDB

Fetching PDB structure

Downloading **PDB structure** with the **protein molecule** from the RCSB PDB database.
Alternatively, a **PDB file** can be used as starting structure.

Building Blocks used:

- Pdb from **biobb_io.api.pdb**

```
# Downloading desired PDB file
# Import module
from biobb_io.api.pdb import pdb

# Create properties dict and inputs/outputs
downloaded_pdb = pdbCode+'.pdb'
prop = {
    'pdb_code': pdbCode
}

#Create and launch bb
pdb(output_pdb_path=downloaded_pdb,
    properties=prop)
```

1. Fetch a PDB

1. Choose a story

1. Fetch a PDB

In this section you will learn to:

- Fetch and store in your working directory a PDB using the biobb package
- Fix a protein structure (include atoms that are missing) using the biobb package

1. Fetch a PDB

1. Choose a story

1. Fetch a PDB

Don’t move forward before:

- Knowing the commands to create a PDB file in your environment
- Knowing the commands to fix a protein structure
- Checking the files that you create at each step

2. Create protein system topology

2. Create your characters

2. Create protein system topology

Create protein system topology

Building GROMACS topology corresponding to the protein structure.
Force field used in this tutorial is **amber99sb-ildn**: AMBER **parm99** force field with **corrections on backbone** (sb) and **side-chain torsion potentials** (ildn). Water molecules type used in this tutorial is **spc/e**.
Adding **hydrogen atoms** if missing. Automatically identifying **disulfide bridges**.

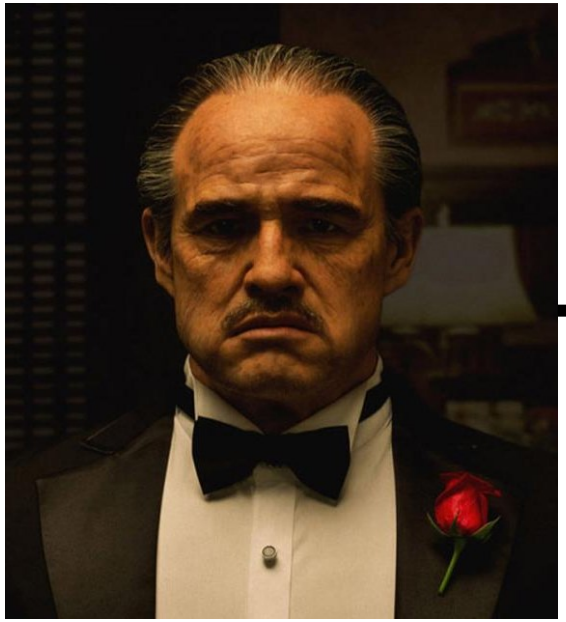
Generating two output files:

- **GROMACS structure** (gro file)
- **GROMACS topology** ZIP compressed file containing:
  - *GROMACS topology top file* (top file)
  - *GROMACS position restraint file/s* (itp file/s)

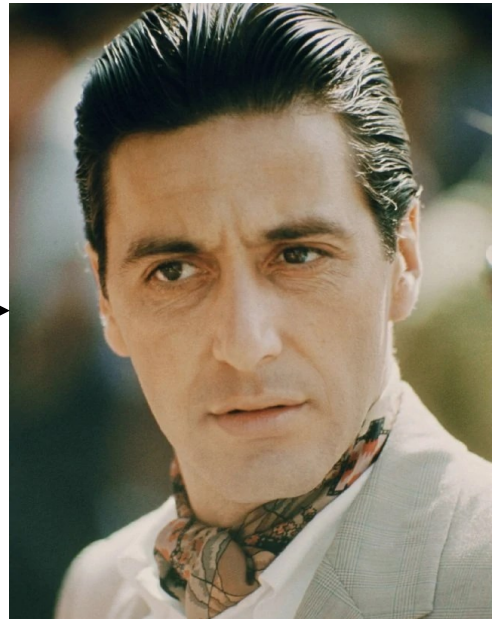
```
# Create system topology
# Import module
from biobb_md.gromacs.pdb2gmx import pdb2gmx

# Create inputs/outputs
output_pdb2gmx_gro = pdbCode+'_pdb2gmx.gro'
output_pdb2gmx_top_zip = pdbCode+'_pdb2gmx_top.zip'

# Create and launch bb
pdb2gmx(input_pdb_path=fixed_pdb,
        output_gro_path=output_pdb2gmx_gro,
        output_top_zip_path=output_pdb2gmx_top_zip)
```

2. Create protein system topology

2. Create your characters



Father/son



Vito Corleone

- Scary
- Dangerous
- Has a lot of money

Michael Corleone

- Scary
- Dangerous
- Has a lot of money

2. Create protein system topology

2. Create protein system topology

When you create the topology of the system, you define the characteristics of all the atoms that will be involved in the simulation, just like characters of a movie.

nr	type	resnr	residue	atom	cgnr	charge	mass	typeB
residue	1 LYS	rtp NLYS	q +2.0					
1	N3	1	LYS	N	1	0.0966	14.01	
2	H	1	LYS	H1	2	0.2165	1.008	
3	H	1	LYS	H2	3	0.2165	1.008	
4	H	1	LYS	H3	4	0.2165	1.008	
5	CT	1	LYS	CA	5	-0.0015	12.01	
6	HP	1	LYS	HA	6	0.118	1.008	
7	CT	1	LYS	CB	7	0.0212	12.01	
8	HC	1	LYS	HB1	8	0.0283	1.008	
9	HC	1	LYS	HB2	9	0.0283	1.008	
10	CT	1	LYS	CG	10	-0.0048	12.01	
11	HC	1	LYS	HG1	11	0.0121	1.008	
12	HC	1	LYS	HG2	12	0.0121	1.008	
13	CT	1	LYS	CD	13	-0.0608	12.01	
14	HC	1	LYS	HD1	14	0.0633	1.008	
15	HC	1	LYS	HD2	15	0.0633	1.008	
16	CT	1	LYS	CE	16	-0.0181	12.01	
17	HP	1	LYS	HE1	17	0.1171	1.008	
18	HP	1	LYS	HE2	18	0.1171	1.008	
19	N3	1	LYS	NZ	19	-0.3764	14.01	
20	H	1	LYS	HZ1	20	0.3382	1.008	
21	H	1	LYS	HZ2	21	0.3382	1.008	
22	H	1	LYS	HZ3	22	0.3382	1.008	

Bonds between atoms:

16	19	1
19	20	1
19	21	1
19	22	1

2. Create protein system topology

2. Create your characters

2. Create protein system topology

In this section you will learn to:

- Create gromacs files (.gro) and topology files (.top)
- Understand gromacs and topology files

2. Create protein system topology

2. Create your characters

2. Create protein system topology

Don’t move forward before:

- Knowing how you can transform a PDB file into gromacs and topology files .
- Find in the gromacs file (.gro) the coordinates for the atoms of the amino group of the Lysine 1. Compare this coordinates to the ones in the PDB file.
- Find in the topology file (.top) the charge and mass of the atoms from the amino group of Lysine 1. What is the charge of this amino group?
- Find in the topology file (.top) how the atoms from the amino group of Lysine 1 are connected.

3. Create the solvent box

3. Create the stage

3. Create the solvent box

Create solvent box

Define the unit cell for the **protein structure MD system** to fill it with water molecules.
A **cubic box** is used to define the unit cell, with a **distance from the protein to the box edge of 1.0 nm**. The protein is **centered in the box**.

Fill the box with water molecules

Fill the unit cell for the **protein structure system** with water molecules.
The solvent type used is the default **Simple Point Charge water (SPC)**, a generic equilibrated 3-point solvent model.

Adding ions

Add ions to neutralize the **protein structure** charge

- Step 1: Creating portable binary run file for ion generation
- Step 2: Adding ions to **neutralize** the system

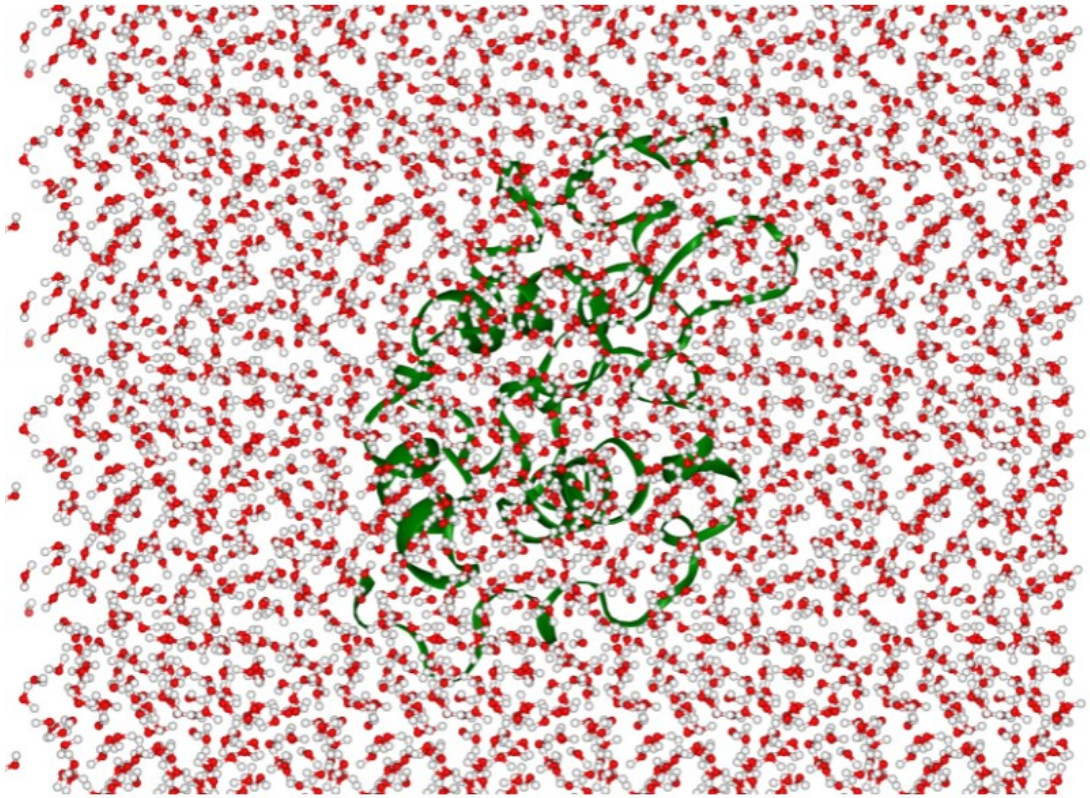
3. Create the solvent box

3. Create the stage

3. Create the solvent box

3. Create the solvent box

The stage for proteins is always an aqueous environment, usually having a physiological concentration of ions



3. Create the solvent box

3. Create the stage

3. Create the solvent box

In this section you will learn to:

- Create a solvent box and fill it with water and your protein
- Understand the gromacs and topology files that are made after you include water molecules
- Understanding binary portable run files
- Using binary portable run files to add ions to the system

3. Create the solvent box

3. Create the stage

3. Create the solvent box

Don’t move forward before:

- Knowing the commands to create a box, inlude water molecules and include ions.
- Comparing the gromacs and topology files for the system without water, with water and with water and ions. What are their differences?
- Taking a look to the binary portable run file. How does it look like? Do you know why these files are important?

4. Energy minimization

4. Put the characters on stage

4. Energy minimization

Energetically minimize the system

Energetically minimize the **protein system** till reaching a desired potential energy.

- Step 1: Creating portable binary run file for energy minimization
- Step 2: Energetically minimize the **system** till reaching a force of 500 kJ mol-1 nm-1.
- Step 3: Checking **energy minimization** results. Plotting energy by time during the **minimization** process.

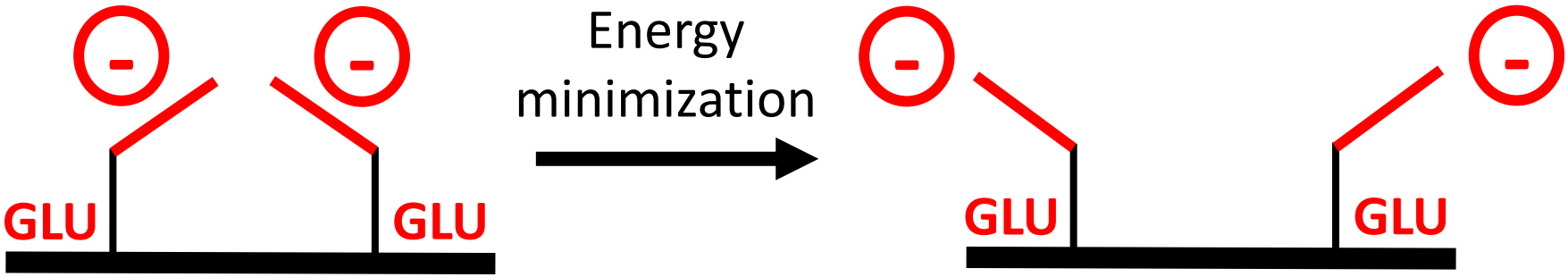
4. Energy minimization

4. Put the characters on stage

4. Energy minimization

4. Energy minimization

PDB structures often contain artifacts or conformations that are not energetically favorable. Energy minimization helps to adopt conformations with low energy to make the simulation stable.



4. Energy minimization

4. Put the characters on stage

4. Energy minimization

In this section you will learn to:

- Create a binary portable run file to execute an energy minimization
- Understand the parameters involved in an energy minimization
- Understand the files that are generated at an energy minimization
- Understand the different energy terms used by force fields to quantify the energy of a system
- Plot the change in potential energy of the system upon energy minimization

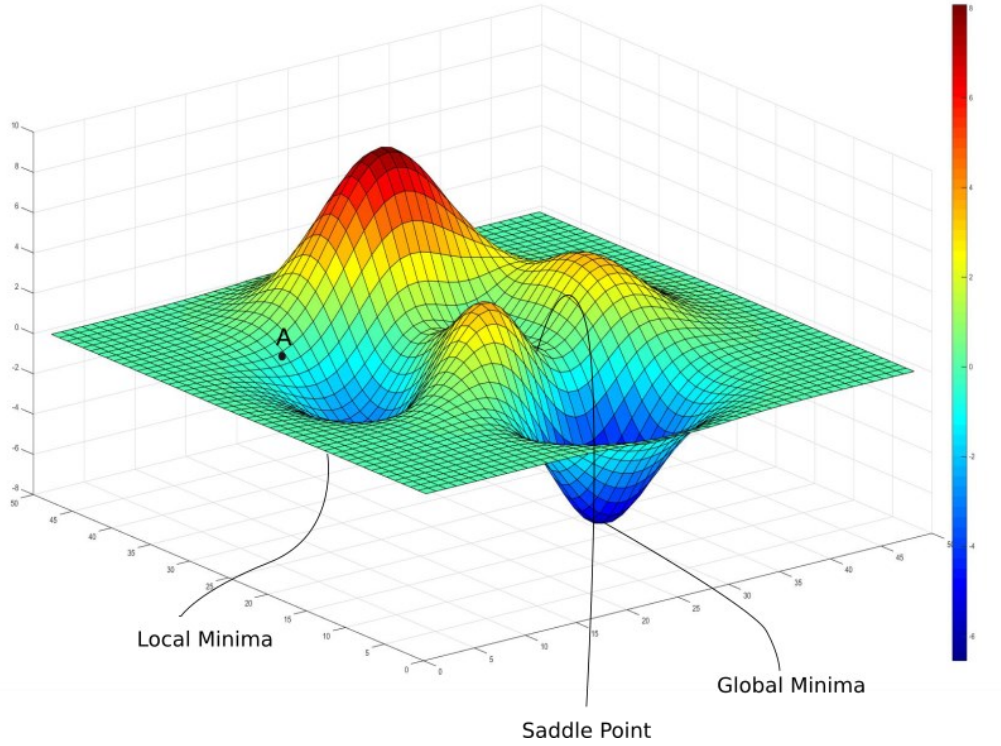
4. Energy minimization

4. Put the characters on stage

4. Energy minimization

Parameters for an energy minimization:

- Algorythm: Steepest descent



4. Energy minimization

4. Put the characters on stage

4. Energy minimization

Parameters for an energy minimization:

- Algorythm: Steepest descent
- Emtol: Energy threhold at which the energy minimization stops
- Emstep: Size of the step in the steepest descent algorythm
- Nsteps: Maximum number of steps made in the energy minimization

4. Energy minimization

4. Put the characters on stage

4. Energy minimization

Files generated by an energy minimization:

- trr files (.trr) and edr files (.edr): binary
- gromacs file (.gro): contains coordenates
- log file (.log): contains the messages sent by the program, the gromacs command used and the energies calculated at different steps

```
        Step           Time
           1        1.00000

Energies (kJ/mol)
        Bond           Angle    Proper Dih.  Improper Dih.          LJ-14
 8.13493e+03    1.83442e+03    5.07958e+03    2.57149e+02    2.42157e+03
  Coulomb-14        LJ (SR)   Coulomb (SR)   Coul. recip. Position Rest.
 1.75029e+04    1.81398e+05   -5.81275e+05    1.73394e+04    1.84602e-01
   Potential Pressure (bar)
-3.47306e+05   -2.24263e+04
```

4. Energy minimization

4. Put the characters on stage

4. Energy minimization

Don’t move forward before:

- Understanding the parameters used for energy minimization. What combination of parameters will involve a more exhaustive energy minimization? What would be their computational cost?
- Comparing the gromacs file after minimization with the gromacs file before minimization. Are they the same? Why?
- Find the potential energy of the system at step 5 of the energy minimization. What file did you checked?
- Why is the energy minimization not creating new topology files?

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

Equilibrate the system (NVT)

Equilibrate the **protein system** in **NVT ensemble** (constant Number of particles, Volume and Temperature). Protein **heavy atoms** will be restrained using position restraining forces: movement is permitted, but only after overcoming a substantial energy penalty. The utility of position restraints is that they allow us to equilibrate our solvent around our protein, without the added variable of structural changes in the protein.

- Step 1: Creating portable binary run file for system equilibration
- Step 2: Equilibrate the **protein system** with **NVT** ensemble.
- Step 3: Checking **NVT Equilibration** results. Plotting **system temperature** by time during the **NVT equilibration** process.

Equilibrate the system (NPT)

Equilibrate the **protein system** in **NPT** ensemble (constant Number of particles, Pressure and Temperature).

- Step 1: Creating portable binary run file for system equilibration
- Step 2: Equilibrate the **protein system** with **NPT** ensemble.
- Step 3: Checking **NPT Equilibration** results. Plotting **system pressure and density** by time during the **NPT equilibration** process.

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

5. Equilibrate the system

Along with energy minimization it helps to adjust the protein structure to the simulation environment. It also sets the environment in the desired conditions of temperature, volume and pressure.

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

In this section you will learn to:

- Differentiate between NVT equilibration and NPT equilibration
- Create a binary portable run file to execute different types of equilibration
- Understand some parameters involved in equilibration
- Plot the change of some variables in your system across time

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

NVT equilibration

- Constant number of particles, volume and temperature
- Try to stabilize the system at a fixed temperature
- The protein has strong spatial restrains
- It equilibrates how the solvent interacts with the protein

NPT equilibration

- Constant number of particles, pressure and temperature
- Try to stabilize the system at a fixed density and pressure

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

Parameters for an energy minimization:

- Type of equilibration: NVT or NPT
- Nsteps: Number of steps made in the equilibration

By controlling the number of steps you control the duration of the equilibration:

1 step = 2 fs = $2 \cdot 10^{-15}$s

5000 steps = 10 ps = $10 \cdot 10^{-12}$s

5. Equilibrate the system

5. Rehearse, warm up

5. Equilibrate the system

Don’t move forward before:

- Knowing how to change the number of steps/time of duration of your equilibration.
- Identifying the output files that come out of an equilibration. See that they are the same type of output files that we got from energy minimization.

6. Run the simulation

6. Record the movie

6. Run the simulation

Free Molecular Dynamics Simulation

Upon completion of the **two equilibration phases (NVT and NPT)**, the system is now well-equilibrated at the desired temperature and pressure. The **position restraints** can now be released. The last step of the **protein** MD setup is a short, **free MD simulation**, to ensure the robustness of the system.

- Step 1: Creating portable binary run file to run a **free MD simulation**.
- Step 2: Run short MD simulation of the **protein system**.
- Step 3: Checking results for the final step of the setup process, the **free MD run**. Plotting **Root Mean Square deviation (RMSd)** and **Radius of Gyration (Rgyr)** by time during the **free MD run** step.

6. Run the simulation

6. Record the movie

6. Run the simulation

6. Run the simulation

Here the force field and the molecular dynamics engine cooperate to simulate the movement of atoms every 2 femtoseconds.

Force field (AMBER parm99)

Describes the energy landscape for all the atoms of a chemical system

→

MD engine (gromacs)

Uses energies described by the force field to calculate forces and accelerations of atoms

6. Run the simulation

6. Record the movie

6. Run the simulation

The process of running a simulation can be compared to recording one specific type of movie: a **stop motion movie**

6. Run the simulation

6. Record the movie

6. Run the simulation

The process of running a simulation can be compared to recording one specific type of movie: a **stop motion movie**

Knowledge of realistic movements

↓

Animator

Force field (AMBER parm99)

↓

MD engine (gromacs)

6. Run the simulation

6. Record the movie

6. Run the simulation

In this section you will learn to:

- Understand how the force field and the MD engine collaborate to make the molecular dynamics simulation
- Create a binary portable run file to execute a free MD simulation
- Choose how long is this simulation going to be (number of steps)

6. Run the simulation

6. Record the movie

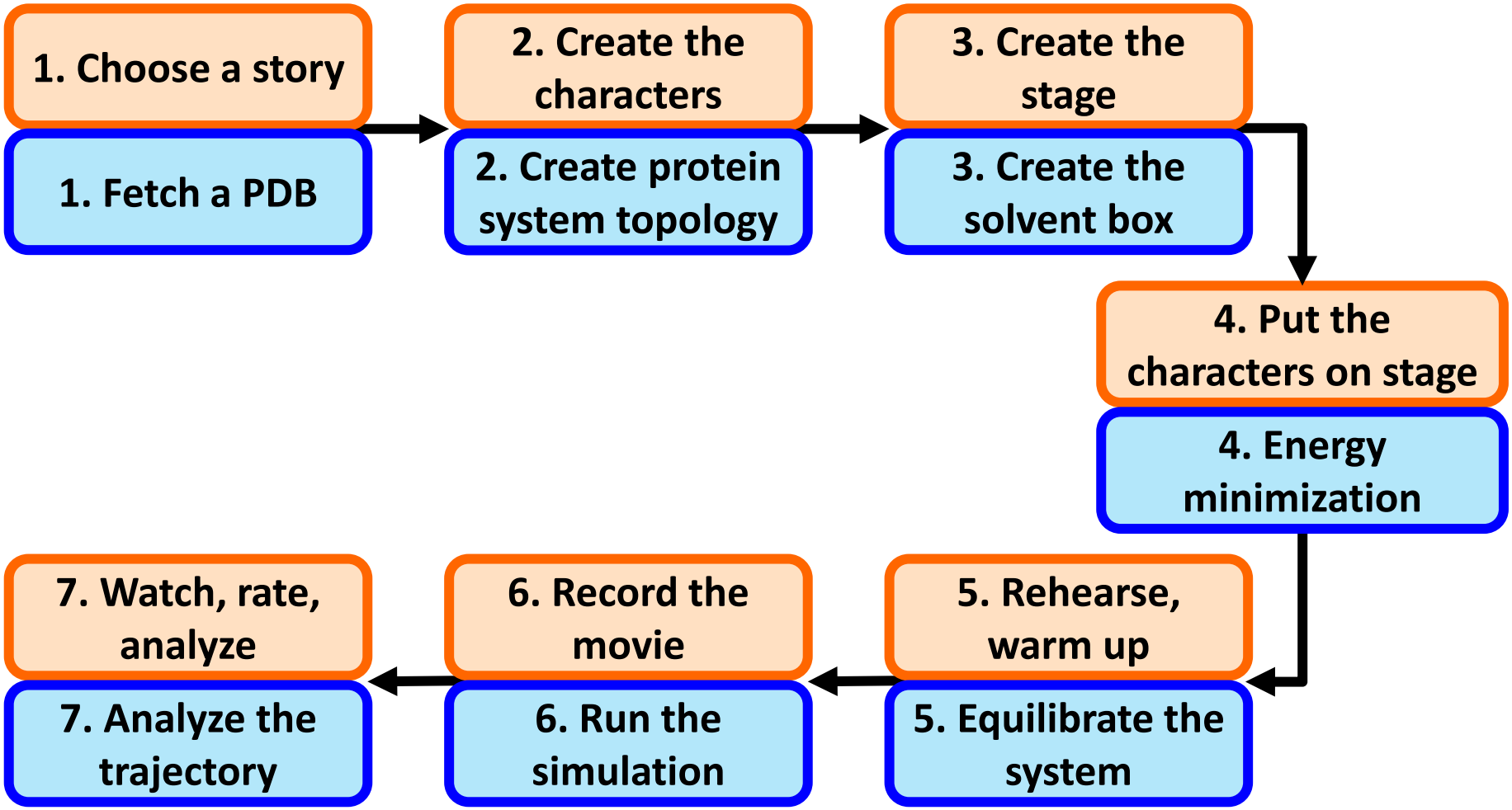
6. Run the simulation

Don’t move forward before:

- Knowing how to change the number of steps/time of duration of your free MD simulation.
- Knowing how to generate and identify binary portable run files.
- Knowing what is the gromacs command that we are using to execute the free MD simulation. In what file can you look for that?

Molecular dynamics set up

Making a molecular dynamics simulation is similar to making a movie



Homework

Make the whole workflow of today for the protein you are working in your project. Next day we will launch a long simulation in a remote cluster.

The simulation should be 5 nanoseconds long

Keep the binary portable file and don't execute it

We will execute the binary portable file in the cluster