Methods of Molecular Dynamics and Monte Carlo Introduction to Gromacs 2016

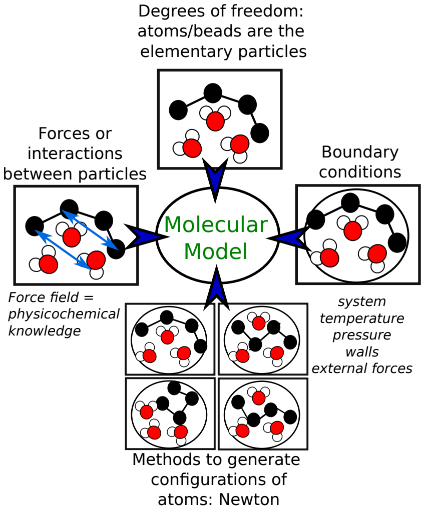
Hector Martinez-Seara, Lukasz Cwiklik

Intitute of Organic Chemistry and Biochemistry
Czech Academy of Sciences

07/12/2016

1. Molecular mechanics (MD)
   - Basics
   - Force field
   - Workflow
2. What is gromacs and when to use it?
   - Description
   - Technical characteristics
3. How to use gromacs?
   - Gromacs general usage
   - Instructions Tutorial
   - Task01: Water box simulation
   - Task02: GFP protein in water simulation
   - Task03: Membrane simulation using CHARMMGUI
   - Task04: Simulation of a transmembrane protein using Martini
4. Appendixes
   - Commands cheatsheet

How ”classical“ molecular dynamics work?



Degrees of freedom: atoms/beads are the elementary particles

Forces or interactions between particles

Boundary conditions

Molecular Model

Force field = physicochemical knowledge

system temperature pressure walls external forces

Methods to generate configurations of atoms: Newton

- Ideal for biological systems where important interactions do not involve reforming of bonds.
- The time evolution of the system could be obtained by numerically integrating the Newton’s equations of motion.
- The heart of the simulation is the force field.

What is Molecular Dynamics?

- Molecular Mechanics (MD) is a methodology that allows to follow the position of any given number of atoms or particles (beads) (1 ... N) with the time assuming that their trajectories can be described accurately in the classical context (Newton's Laws).

$$m_i \frac{d^2\mathbf{r_i}}{dt^2} = \mathbf{F_i} = -\nabla_{\mathbf{r_i}} U(\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_N}) \tag{1}$$

What is a force Field?

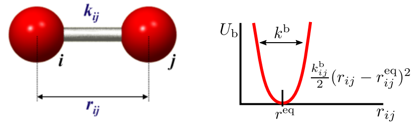
- Force fields are normally assumed to consist of pairwise-additive potentials
    - More accurate descriptions may involve the use of three-body terms or non-additive potentials.
- A compromise between their accuracy and simplicity is essential. Most force fields can be described as:

$$U = U_{bond} + U_{ang} + U_{tors} + U_{vdw} + U_{coul} \tag{2}$$

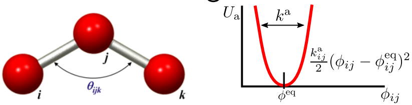
  - Bonded interactions: bond stretching, angle bending and torsional dihedral energy contributions for atoms separated by 1, 2 and 3 bonds.
  - Non Bonded interactions: Van der Waals and Coulomb energy contributions. These forces apply to all atom pairs.

Typical potentials used in biosystems

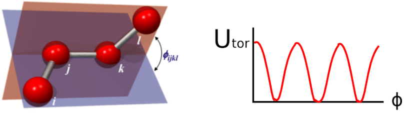
Bond



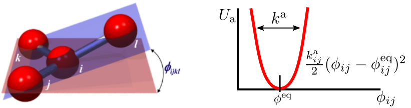
Angle



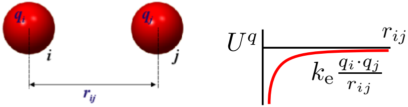
Dihedral



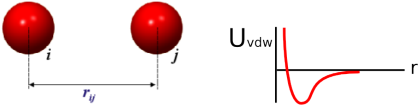
Improper



Coulomb

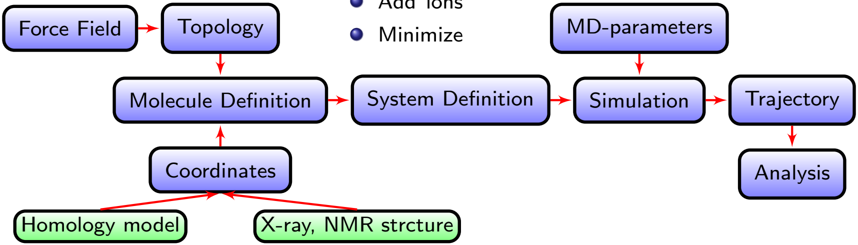


Van der Waals



Typical Molecular dynamics workflow

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



What is GROMACS?

In short

GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.

Area of usage

It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the nonbonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

Major characteristics of GROMACS - I

- Extremely high performance compared to all other MD programs.
- Excellent CUDA-based GPU acceleration on GPUs.
- User-friendly, with topologies and parameter files written in clear text format.
- There is no scripting language - just simple command line options.
- Well documented.
- Both input files and trajectories are independent of hardware used to generate them.

Major characteristics of GROMACS - II

- Provides a very compact way of storing trajectory data.
- Offers a large selection of flexible tools for trajectory analysis - you won’t have to write any code to perform routine analyses.
- Output trajectories are supported by most of the typical third parties codes which are used to visualize and analyze the code.
- High parallelization capabilities, *e.g.* MPI, OPENMP, hybrid mode.
- The package includes a fully automated topology builder for proteins, even multimeric structures.
- GROMACS is Free Software, available under the GNU Lesser General Public License.

Major force fields for MD simulations of biological systems in gromacs

All Atom

ff	Protein	Membrane	Sugars	DNA/RNA
charmm	charmm36	charmm36	charmm36	charmm36
amber	amber99sb-ildn	slipids	glycam06h	amber99sb-ildn
oplsaa	oplsaa	Macrog	(very basic)	no

United Atom

ff	Protein	Membrane	Sugars	DNA/RNA
gromos	yes	yes	(very basic)	yes

Coarse Grained

ff	Protein	Membrane	Sugars	DNA/RNA
martini	yes	yes	(very basic)	yes

The *gmx* prefix

gmx prefix

Starting from gromacs 5 you should invoke all the tools using the *gmx* prefix. Starting from version 5.1 this is in fact the only way to access gromacs tools

Executing *gmx*

gmx [-[no]h] [-[no]quiet] [-[no]version] [-[no]copyright] [-nice <int>] [-[no]backup]

The most useful *gmx* command

gmx help

Listing the available gromacs tools

The most useful *gmx* command

gmx help commands

```
LIST OF AVAILABLE COMMANDS

Usage: gmx [<options>] <command> [<args>]

Available commands:
  anadock    Cluster structures from Autodock runs
  anaeig     Analyze eigenvectors/normal modes
  analyze    Analyze data sets
  angle      Calculate distributions and correlations for angles and dihedrals
  bar        Calculate free energy difference estimates through Bennett's acceptance ratio
  bundle     Analyze bundles of axes, e.g., helices
  check      Check and compare files
  ...        ...
```

Gromacs v5.1.4 has around 100 tools already at your service!!

When you should use gromacs tools?

A free advise

Do not reinvent the wheel. Check before starting creating your own tools. The chances that gromacs has what you need is really high. And likely it will do it better and faster that you can!! This will save a lot of your valuable time.

Running a gromacs tools

The most useful *gmx* command

gmx help <command>

gmx help solvate

```
SYNOPSIS
gmx solvate [-cp [<.gro.g96...>]] [-cs [<.gro.g96...>]] [-p [<.top>]] [-o [<.gro.g96...>]] [-box <vector>] [-radius <real>]
[-scale <real>] [-shell <real>] [-maxsol <int>] [-[no]vel]

DESCRIPTION
gmx solvate can do one of 2 things:
1) Generate a box of solvent. Specify -cs and -box. Or specify -cs and -cp with a structure file with a box, but without atoms.
2) Solvate a solute configuration, e.g. a protein, in a bath of solvent
...

OPTIONS
Options to specify input files:
-cp [<.gro/.g96/...>] (protein.gro) (Opt.) Structure file: gro g96 pdb brk ent esp tpr
-cs [<.gro/.g96/...>] (spc216.gro) (Lib.) Structure file: gro g96 pdb brk ent esp tpr
Options to specify input/output files:

...
```

Tasks: General Instructions

General Instructions

- Only use the folder provided to perform the tasks included in this course.
  - *cd $HOME/task_material/*
- Do not change the file names used in the tasks.
  - In real life this is not necessary but it is to minimize wasting time during this course.
- All exercises provide solution files prefixed with "EX".
  - Never modify these files. They are provided as reference only.
  - Before using them copy them: *cp EX-$FILENAME $FILENAME*
- If you are stuck ask for help immediately to the professors.

Task01: Simulation of a water box with/without ions

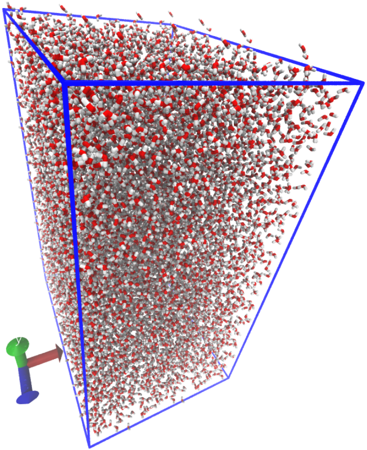
Purpose of the task

- To introduce the general workflow of a Molecular dynamics simulation. Many concepts introduced here are shared among MD engines. Only the commands used differ.
- To show all the stages required to produce scientific quality data when using MD simulations
  1. **Minimization** - To remove bad contacts produced in the assembling of the system
  2. **Equilibration** - Simulation to allow the system to the correct equilibrium simulation ensemble.
  3. **Production** - The stage where we produce the real data from our system.

Generation and visualization of a water box

Water box generation:

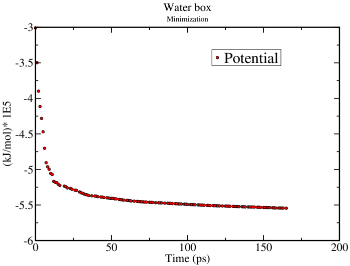
- Using *gmx solvate* generate a TIP3P water box of 5x10x7 namometers
  1. *cd task01-water_box*
  2. *gmx solvate -cs -box 5 10 7 -o water.gro*
- Inspect the content of the gro file in: *e.g. vim, nano, emacs,...*
- Visualize the gro file: *vmd water.gro*
  1. Change the representation to orthographic
  2. Draw the simulation box:
     - Execute *pbc box* in the console execute



Water box simulation - Minimization

Minimization protocol:

1. *cp EX-01-min.mdp 01-min.mdp*
2. *cp EX-00-water.top 00-water.top*
   (change number of waters if required)
3. *gmx grompp -c 00-water.gro -f 01-min.mdp*
   -p 00-water.top -o 01-water-min.tpr
   -po 01-mdout-min.mdp
4. *gmx mdrun -v -deffnm 01-water-min*
5. *echo "Potential" |gmx energy*
   -f 01-water-min.edr -o 01-energy-min.xvg
6. *xmgrace -free -nxy 01-energy-min.xvg*

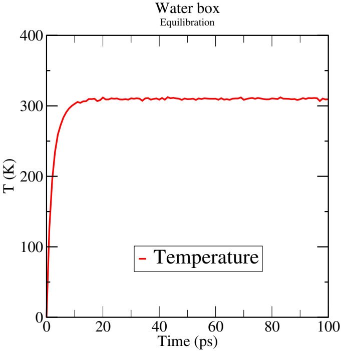


Important step to avoid possible initial bad contacts

Water box simulation - MD Equilibration

Equilibration protocol:

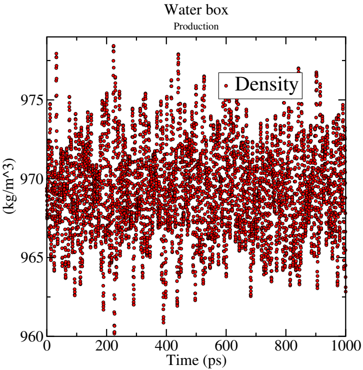
1. *cp EX-01-water-min.gro 01-water-min.gro*
2. *cp EX-02-eq.mdp 02-eq.mdp* (notice that we use berendsen)
3. *gmx grompp -c 01-water-min.gro -f 02-eq.mdp*
   -p 00-water.top -o 02-water-eq.tpr
   -po 02-mdout-eq.mdp
4. *gmx mdrun -v -deffnm 02-water-eq*
5. *echo "Temperature" |gmx energy*
   -f 02-water-eq.edr -o
   02-temperature-eq.xvg
6. *xmgrace -free -nxy 02-temperature-eq.xvg*



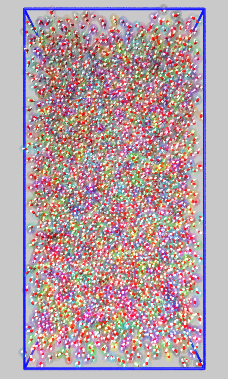
Water box simulation - MD Simulation

Production simulation protocol:

1. *cp EX-02-water-eq.gro 02-water-eq.gro*
2. *cp EX-03-md.mdp 03-md.mdp*
3. *gmx grompp -c 02-water-eq.gro -f 03-md.mdp*
   -p 00-water.top -o 03-water-md.tpr
   -po 03-mdout-md.mdp
4. *gmx mdrun -v -deffnm 03-water-md*
5. *echo "Density" | gmx energy -f*
   03-water-md.edr -o 03-density-md.xvg
6. *xmgrace -free -nxy 03-density-md.xvg*



Water box simulation - Visualization MD Simulation



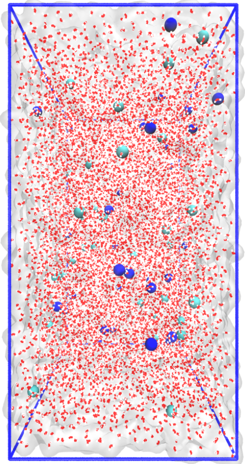
Visualization protocol:

1. Fix broken molecules across periodic boundary conditions
   - *gmx trjconv -f 03-density-md.xtc -s 03-density-md.tpr -o 03-density-md_mol.xtc -pbc mol*
2. Visualize water box in VMD
   - *vmd 02-water-eq.gro 03-water-md_mol.xtc*
   - Visualize the hydrogen bond formation destruction during the simulation.
     NOTE: Select "Update Selection Every Frame"

Water box with ions - Add ions to water

Salt addition protocol:

1. *cp EX-03-water-md.gro 03-water-md.gro*
2. *cp EX-00-water.top 04-ions.top*
3. *gmx grompp -c 03-water-md.gro -f 03-md.mdp -p 00-water.top -o 04-water-genion.tpr -po 04-mdout-genion.mdp*
4. *gmx genion -s 04-water-genion.tpr -p 04-ions.top -o 04-NACL-solution-genion.gro -conc 0.150* (Select "SOL")
5. *vmd 04-NACL-solution-genion.gro*
   - Create separate representations for water (lines) and ions (VDW)



Water box with ions - Simulation

Simulation protocol (use the mdp files from the water simulation):

1. Minimization:
    1. *gmx grompp -f 01-min.mdp -c 04-NACL-solution-genion.gro -p 04-ions.top -o 05-ions-min.tpr -po 05-ions-min_mdout.mdp*
    2. *gmx mdrun -v -deffnm 05-ions-min*
2. Equilibration:
    1. *gmx grompp -f 02-eq.mdp -c 05-ions-min.gro -p 04-ions.top -o 06-ions-eq.tpr -po 06-ions-eq_mdout.mdp*
    2. *gmx mdrun -v -deffnm 06-ions-eq*
3. Production simulation:
    1. *gmx grompp -f 03-md.mdp -c 06-ions-eq.gro -p 04-ions.top -o 07-ions-md.tpr -po 07-ions-md_mdout.mdp*
    2. *gmx mdrun -v -deffnm 07-ions-md*

Water box with ions - Visualization

Visualization protocol of 07-ions-md.xtc trajectory:

1. Fix broken molecules across periodic boundary conditions:
   - *gmx trjconv -f 07-ions-md.xtc -s 07-ions-md.tpr -o 07-ions-md_mol.xtc -pbc mol*
2. Visulaize trajectory with VMD
   - *vmd 06-ions-eq.gro 07-ions-md_mol.xtc*
   - Create separate representations for water (lines) and ions (VDW)
   - Visualize the hydrogen bond formation destruction during the simulation
   - Analyze visually the first hydration shell of a selected cation and anion. NOTE: Select "Update Selection Every Frame"

Analyzing simulation

Time dependencies: energy, temperature, pressure

1. *gmx energy -f 07-ions-md.edr -o PROPERTY.xvg*
2. *xmgrace -free -nxy PROPERTY.xvg*

Radial distribution function of water around Na^+ and Cl^-

1. *gmx rdf -f 07-ions-md.xtc -s 07-ions-md.tpr -o 07-ions-md-ana-rdf-NA.xvg -ref NA -sel SOL*
2. *gmx rdf -f 07-ions-md.xtc -s 07-ions-md.tpr -o 07-ions-md-ana-rdf-CL.xvg -ref CL -sel SOL*
3. xmgrace 07-ions-md-ana-rdf-NA.xvg 07-ions-md-ana-rdf-CL.xvg -free -legend load

Water diffusion coefficient - Mean Squared displacement (MSD):

1. *gmx msd -f EX-07-ions-md.xtc -s EX-07-ions-md.tpr -o EX-07-ions-md-ana-msd-water.xvg -rmcomm* (How does compare with the experimental value: $2.9x10^{-5}cm^2/s$)

Task02: Simulation Green Fluorecence protein (GFP)

Purpose of the task

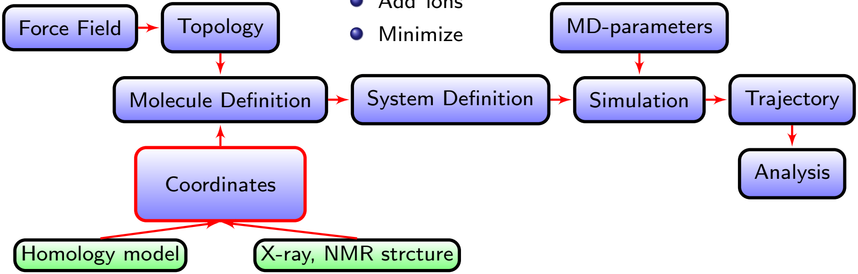
- In this task we will simulate the GFP protein in solution (PDB: 1GFL). Molecular dynamics is an excellent tool to study protein behavior. For these reasons many MD works actually focus on proteins.
- Most of the concepts introduced here can be applied to the simulation of other biomolecules with only small adaptations/modifications.

Change to the new task directory

- *cd ../task02-GFP*

Where to start?

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Obtaining the protein structure

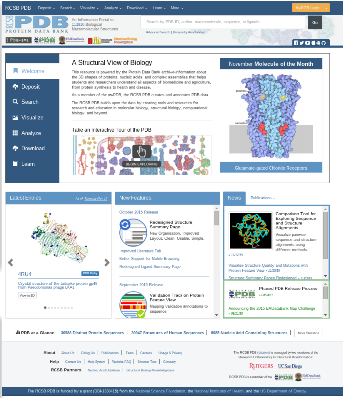
When simulating biomolecules establishing a reasonable position of its constituent atoms is not a trivial task:

- They usually contain hundreds or even million of atoms.
- Their connectivity, stereochemistry and spacial distribution is very complicated.

Obtaining protein structure

RCSB Protein Data Bank

http://www.rcsb.org/



An Information Portal to 113816 Biological Macromolecular Structures

A Structural View of Biology

This resource is powered by the Protein Data Bank archive-information about the 3D shapes of proteins, nucleic acids, and complex assemblies that helps students and researchers understand all aspects of biomedicine and agriculture, from protein synthesis to health and disease.

As a member of the wwPDB, the RCSB PDB curates and annotates PDB data.

The RCSB PDB builds upon the data by creating tools and resources for research and education in molecular biology, structural biology, computational biology, and beyond.

Take an Interactive Tour of the PDB

November Molecule of the Month

Glutamate-gated Chloride Receptors

Latest Entries As of Tuesday Nov 17

4RU4

Crystal structure of the tailspike protein gp49 from Pseudomonas phage LKA1

New Features

October 2015 Release

Redesigned Structure Summary Page

New Organization. Improved Layout. Clean. Usable. Simple.

Improved Literature Tab

Better Support for Mobile Browsing

Redesigned Ligand Summary Page

September 2015 Release

Validation Track on Protein Feature View

Mapping validation annotations to sequence

News Publications

Comparison Tool for Exploring Sequence and Structure Alignments

Visualize pairwise sequence and structure alignments using different methods.

» 11/17/15

Visualize Structure Quality and Mutations with Protein Feature View » 11/10/15

Structure Summary Pages Redesigned » 11/03/15

Phased PDB Release Process » 08/24/15

Announcing the 2015 EMDataBank Map Challenge » 08/11/15

PDB at a Glance | 36988 Distinct Protein Sequences | 30047 Structures of Human Sequences | 8085 Nucleic Acid Containing Structures

About: About Us | Citing Us | Publications | Team | Careers | Usage & Privacy

Help: Contact Us | Help System | Website FAQ | Browser Test | Glossary

RCSB Partners: Nucleic Acid Database | Structural Biology Knowledgebase

The RCSB PDB (citation) is managed by two members of the Research Collaboratory for Structural Bioinformatics:

RUTGERS UC San Diego

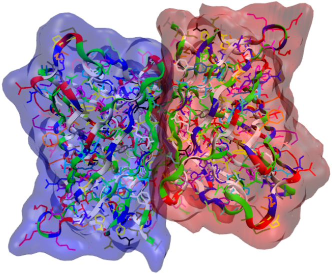
RCSB PDB is a member of the wwPDB EMDataBank

The RCSB PDB is funded by a grant (DBI-1338415) from the National Science Foundation, the National Institutes of Health, and the US Department of Energy.

Download the pdb of the GFP protein

Obtaining GFP protein - 1GFL

- Download 1GFL from http://www.rcsb.org/ (save it as 01-GFP-topology/00-1GFL.pdb)
- Open 00-1GFL.pdb in your favorite text editor
- Visualize it in vmd
  - Display secondary structure information by changing color and representation
  - Display different amino acids types

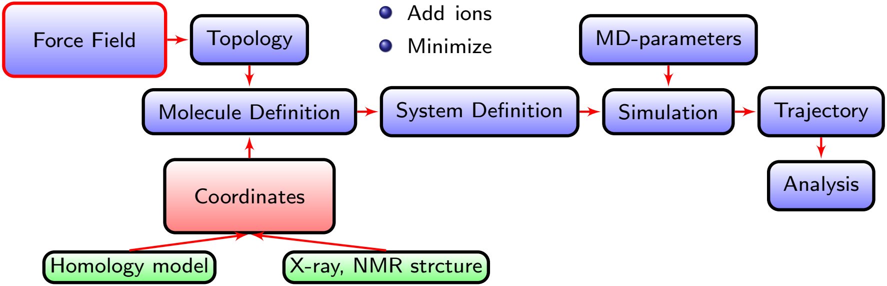


Observations:

1. There are two different chains
2. There are few crystallized waters
3. Read REMARKS 350 465 470!!

Selecting a force field

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Selection of a decent protein force field

All atom

charmm36, amber99sb-ildn, oplsaa, charmm27

United atom

GROMOS96 54a7

Coarse-grained

Martini

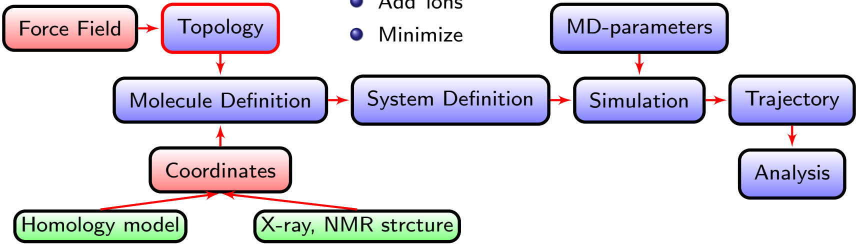
The force field to be selected for this task

Not good but seldom used.

Never good – You MUST know what are you doing.

Task02: Generating the protein topology

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



gmx pdb2gmx - your best/worst friend

gmx pdb2gmx

Tool provided by gromacs to convert coordinate files to topology and force field compliant coordinate files. Mainly used for proteins.

gmx pdb2gmx reads a .pdb (or .gro) file, reads some database files, adds hydrogens to the molecules and generates coordinates in GROMACS format and a topology in GROMACS format. These files can subsequently be processed to generate a run input file.

Normal execution

gmx pdb2gmx -f protein.pdb

GFP topology - Raw attempt

Try to generate the topology for 00-1GFL.pdb using *pdb2gmx* using amber99sb-ildn using tip3p water model

Forced execution

```
gmx pdb2gmx -f 00-1GFL.pdb -missing -o 00-1GFL -p 00-1GFL -i 00-1GFL
```

WARNING: atom O is missing in residue SER 65 in the pdb file.

WARNING: Residue 65 named SER of a molecule in the input file was mapped to an entry in the topology database, but the atom O used in an interaction of type improper in that entry is not found in the input file. Perhaps your atom and/or residue naming needs to be fixed.

Normal execution

```
gmx pdb2gmx -f 00-1GFL.pdb
```

WARNING: atom O is missing in residue SER 65 in the pdb file.

Fatal error: There were 1 missing atoms in molecule Protein_chain_A, if you want to use this incomplete topology anyhow, use the option -missing

OPTION:

-[no]missing (no) Continue when atoms are missing, dangerous

GFP topology - Preprocess original pdb

Clean 00-1GFL.pdb

1. Remove all HETATM lines
   grep -v "HETATM" 00-1GFL.pdb > 01-1GFL_noHETATM.pdb
   - Be aware that some nonstandards residues might be input as HETATM, *e.g.* Selenomethionine.
2. Remove chain B from pdb (for convenience of the exercise)
   grep -v " B " 01-1GFL_noHETATM.pdb >02-1GFL_A_raw.pdb
   - Notice that B is surrounded by one space in each side

- This is a very crude method. It is often better to open the pdb with a raw text editor (e.g. vi, emacs, nano) and do these tasks manually.

GFP topology - Clean pdb ("Extra")

pdb4amber

- Tool shiped with ambertools which prepare pdbs for further processing in *tleap* which add missing heavy atoms.
- Among its capabilities it can remove waters, create Disulfide bonds, estimate the protonation state bases on the hydrogen atoms in the histidines ...

Prepare pdb for ambertools using pdb4amber

pdb4amber -i 02-1GFL_A_raw.pdb -o 03-1GFL_pdb4amber.pdb --reduce --dry

- Notice it says: "Missing Heavy Atoms (Renumbered Residues!) None"
- Try to make *pdb2gmx* using *-ignh*

GFP topology - Add missing heavy atoms ("Extra")

Correct missing atoms with *tleap* (part of ambertools suite)

1. *tleap*
2. *source leaprc.ff14SB*
3. *x = loadPdb 03-1GFL_pdb4amber.pdb*
4. *savepdb x 04-1GFL.pdb*
5. *quit*

- This method will not correct for missing residues. For this you can use Modeller which can also be used to correct for missing atoms.

GFP topology - Generate working topology

Generate topology GFP

Use *gmx pdb2gmx* to generate the topology for 04-1GFL.pdb using tip3p water model and amber99sb-ildn force field

Normal execution - Problems with hydrogens (FAILS!!)

```
gmx pdb2gmx -f 04-1GFL.pdb -o 04-1GFL.gro -i 04-1GFL.itp -p 04-1GFL.top
```

(Hydrogen order is force field dependent)

Normal execution but ignoring pdb hydrogens (WORKS!!)

```
gmx pdb2gmx -f 04-1GFL.pdb -o 04-1GFL.gro -i 04-1GFL_posres.itp -p 04-1GFL.top -n 04-1GFL.ndx -ignh
```

- Notice that the total charge of the protein is -5 e

ÚOCHB IOCB PRAGUE

GFP topology - Isolate 1GFL molecule topology

For convenience and clarity we usually isolate molecules topologies from the system topology. This allows the recycling of the generated topologies and simplify the system topology file.

Prepare GFP topology file: 1GFL.itp

- Extract the 1GFL.itp from 04-1GFL.top using text editor
  - *cp 04-1GFL.top 04-1GFL.itp*
  - Open 1GFL.itp in using your favorite raw text editor (vim, nano, emacs . . .)
  - Remove until "[Moleculetype]" from 1GFL.itp
  - Change name "[Moleculetype]" from "Protein" to "1GFL"
  - Remove everything from "; Include water topology"
  - Save the newly created 1GFL topology file

The molecule topology file - How itp files look like?

The itp files - Topologies of Molecules

```
[ moleculetype ] ← Tag that marks the beginning of a molecule definition
; Name nrexcl ← This is a comment which starts ";"
Protein 3 ← Do not compute long range interaction for atoms 3 bonds apart
[ atoms ]
; nr type resnr residue atom cgnr charge mass typeB chargeB massB
1 N3 1 ALA N 1 0.1414 14.01 ; qtot 0.1414
. . .
[ bonds ] ← Funct type depends on the force field used
; ai aj funct c0 c1 c2 c3
1 2 1 ← last number, i.e. 1, refers to a simple harmonic potential. The function used for bond by amber99sb-ildn
. . .
[ angles ]
; ai aj ak funct c0 c1 c2 c3
2 1 3 1 ← The lack of explicit parameters c0 c1 ... means that default parameters from #include
"amber99sb-ildn.ff/forcefield.itp" file will be used
. . .
[ dihedrals ]
; ai aj ak al funct c0 c1 c2 c3 c4 c5
2 1 5 6 9
. . .
; Include Position restraint file
#ifdef POSRES ← Topologies can conditionally include other parameters, i.e. define=-DPOSRES in mdp file"
#include "04-1GFL_posres.itp" ← Restrains file for our protein"
#endif
```

GFP topology - The 04-1GFL.itp topology file

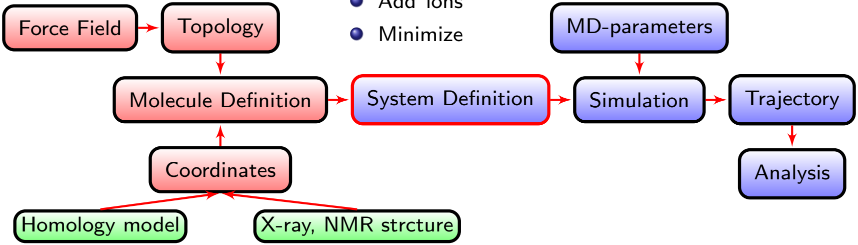
```
[ moleculetype ]
; Name nrexcl
1GFL 3

[ atoms ]
; nr type resnr residue atom cgnr charge mass typeB chargeB massB
; residue 1 ALA rtp NALA q +1.0
1 N3 1 ALA N 1 0.1414 14.01 ; qtot 0.1414
2 H 1 ALA H1 2 0.1997 1.008 ; qtot 0.3411
3 H 1 ALA H2 3 0.1997 1.008 ; qtot 0.5408

...

3576 3593 3591 3592 4
3591 3595 3593 3594 4
3595 3606 3605 3607 4

; Include Position restraint file
#ifdef POSRES
#include ”EX-04-1GFL_posres.itp”
#endif
```

Protein definition achieved

— — — — — — — — — — — PLEASE NOTE — — — — — — — — — — —

We have successfully generated a topology from: 04-1GFL.pdb. The Amber99sb-ildn force field and the tip3p water model are used.

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



How do system topology files look like?

The top files

```
#include ”amber99sb-ildn.ff/forcefield.itp” ← Force field definition

#include ”Protein.itp” ← ; Include protein topologies

#include ”amber99sb-ildn.ff/tip3p.itp” ← Include water topology

#include ”amber99sb-ildn.ff/ions.itp” ← Include topology for ions

[ system ]
; Name
GREEN FLUORESCENT PROTEIN

[ molecules ]
; Compound        #mols
PROTEIN               1 ← Number of proteins
SOL                1000 ← Number of waters (We do not have any yet!!)
```

Prepare system topology file

Prepare system topology file: 05-1GFL_nowater.top

```
#include ”amber99sb-ildn.ff/forcefield.itp”

#include ”04-1GFL.itp”

#include ”amber99sb-ildn.ff/tip3p.itp”

#include ”amber99sb-ildn.ff/ions.itp”

[ system ]
; Name
GREEN FLUORESCENT PROTEIN

[ molecules ]
; Compound        #mols
1GFL                 1
```

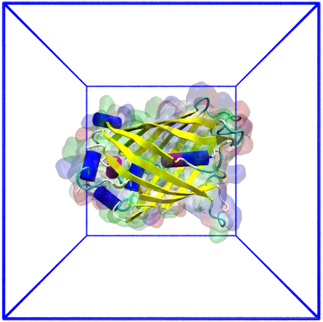
NOTE: cp EX-05-1GFL_nowater.top 05-1GFL_nowater.top or use EX-04-1GFL.top as template.

GFP in vacuum - Create simulation box

Put GFP protein in a simulation box

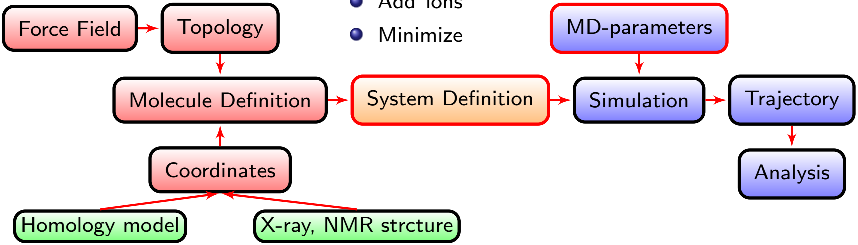
1. *gmx editconf -f 04-1GFL.gro -o 05-1GFL_nowater_center.gro*

   -box 6 6 6
2. *vmd 05-1GFL_nowater_center.gro*



Generation of coordinates and topologies for protein simulation in vacuum

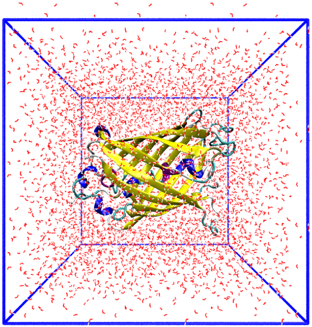
- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



GFP in water - Solvate GFP

Immerse the GFP protein in a box of water

- *cp 05-1GFL_nowater.top 06-1GFL_water.top*
- *gmx solvate -cp 05-1GFL_nowater_center.gro -p 06-1GFL_water.top -o 06-1GFL_water.gro -cs*
- *vmd 06-1GFL_water.gro*



GFP in water - The parameter file minimization

A typical minimization mdp file: 07-min.mdp

define = -DPOSRES ← Protein restrained

integrator = steep ← We will do a minimization

emtol = 1000.0 ← min energy difference KJ/mol

nsteps = 5000 ← max number steps

nstlist = 10 ← how often update neighbors list

cutoff-scheme = Verlet ← Search neighbor method for nonbonded-interaction

rlist = 1.2 ← force field dependent

vdwtype = Cut-off ← Calculate vdw interactions with atoms inside "rvdw"

vdw-modifier = Potential-shift-Verlet ← force field dependent

rvdw_switch = 1.0 ← force field dependent

rvdw = 1.2 ← force field dependent

coulombtype = pme ← Usage particle mesh Ewald

rcoulomb = 1.2 ← force field dependent

coulomb-modifier = Potential-shift-Verlet ← force field dependent

constraints = none ← No constrains will be used

constraint_algorithm = LINCS ← Useless as "constrains = none"

GFP in water - Minimization

Removal of bad contacts between water and a constrained GFP

1. *gmx grompp -f 07-min.mdp \\← Minimization parameters*
   -p 06-1GFL_water.top \\← Topology system
   -c 06-1GFL_water.gro \\← Coordinates atoms
   -o 07-1GFL_water_min.tpr \\← Output file for mdrun
   -po 07-1GFL_water_min_mdout.mdp ← Full list minimizaton parameters)

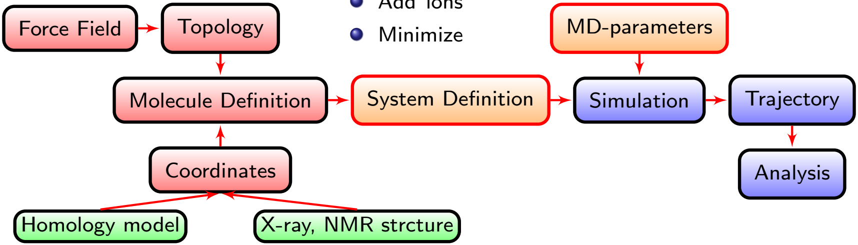
2. *gmx mdrun -deffnm 07-1GFL_water_min -v*

3. *vmd 07-1GFL_water_min.gro*

ÚOCHB IOCB PRAGUE

What to do after adding waters and minimize?

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



GFP in water - Neutralization of excess charges and medium ionic force adjustment

What was this note about

- *gmx grompp -f 07-min.mdp -p 06-1GFL_water.top -c 06-1GFL_water.gro -o 07-1GFL_water_min.tpr -po 07-1GFL_water_min_mdout.mdp*

  NOTE 1 [file 06-1GFL_water.top, line 20]:

  System has non-zero total charge: -5.000000
- It means that GFP protein has more negatively charged amino acids than positive ones. We need to add cations to neutralize this excess of charge, *e.g.* NA^+ ions.
- Also in biological systems the medium contain salt, *e.g.* $\mathrm{Na}^+\mathrm{Cl}^-$ ions.

GFP in water - Adding ions *gmx genion*

Add salt to the system using gmx solvate

- Neutralize and add 150 mMol $Na^{+}Cl^{-}$ ions.
- Do first:
  - *cp 06-1GFL_water.top 08-1GFL_solvated.top*
  - *gmx grompp -f 07-min.mdp -c 07-1GFL_water_min.gro -p 06-1GFL_water.top -o 08-1GFL_genion.tpr -po 08-1GFL_genion_mdout.mdp*
- Output coordinate file should be called "08-1GFL_genion.gro"

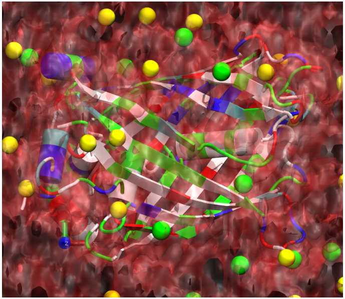
Solution

- *gmx genion -s 08-1GFL_genion.tpr -p 08-1GFL_solvated.top -o 08-1GFL_genion.gro -neutral -conc 0.15*
  - Select "SOL"
  - Note: It does not really provide 150 mMol!!

GFP in water - Minimize GPF system

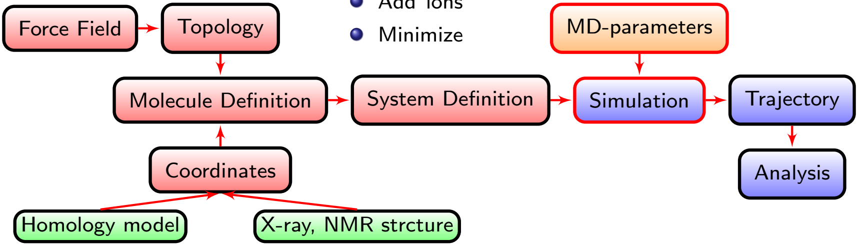
Minimize system after ions additions

1. *gmx grompp -f 07-min.mdp*
   -p 08-1GFL_solvated.top
   -c 08-1GFL_genion.gro
   -o 09-1GFL_solvated_min.tpr
   -po 09-1GFL_solvated_min_mdout.mdp
2. *gmx mdrun -v*
   -deffnm 09-1GFL_solvated_min
3. *vmd 09-1GFL_solvated_min.gro*



GFP in water - Equilibration process

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



GFP in water - Equilibration parameters

Equilibration file: 10-eq_pr.mdp

```
define = -DPOSRES ;Protein is restrained
integrator = md
dt = 0.002
nsteps = 500000 ; 1 ns
nstlog = 500 ; 1ps
nstenergy = 500 ; 1ps
nstxout-compressed = 500 ; 1ps

cutoff-scheme = Verlet
nstlist = 20
rlist = 1.2
coulombtype = pme
rcoulomb = 1.2
coulomb-modifier = Potential-shift-Verlet
vdwtype = Cut-off
vdw-modifier = Potential-shift-Verlet
rvdw_switch = 1.0
rvdw = 1.2
```

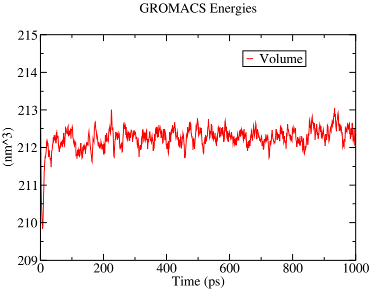
Continuation: 10-eq_pr.mdp

```
tcoupl = berendsen ; nose-hoover
tc_grps = Protein water_and_ions
tau_t = 1.0 1.0
ref_t = 310.00 310.00

pcoupl = berendsen ; Parrinello-Rahman
pcoupltype = isotropic
tau_p = 5.0
compressibility = 4.5e-5
ref_p = 1.0

constraints = h-bonds
constraint_algorithm = LINCS

nstcomm = 100
comm_mode = linear

refcoord_scaling = com
```

GFP in water - Equilibration I

Equilibration restrainded GFP in water

1. gmx grompp -f 10-eq_pr.mdp
   -c 09-1GFL_solvated_min.gro
   -p 08-1GFL_solvated.top
   -o 10-1GFL_solvated_eq_pr.tpr
   -po 10-1GFL_solvated_eq_pr_mdout.mdp
2. gmx mdrun -v -deffnm 10-1GFL_solvated_eq_pr
3. echo "Volume" |gmx energy
   -f 10-1GFL_solvated_eq_pr.edr
   -o 10-1GFL_solvated_eq_pr-Volume.xvg
4. xmgrace 10-1GFL_solvated_eq_pr-Volume.xvg



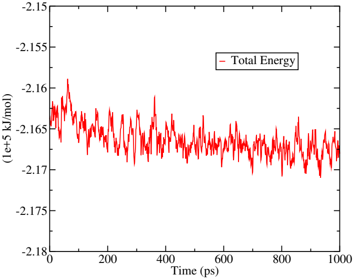
GFP in water - Equilibration II (no restrains)

Equilibration free GFP in water

1. *cp 10-eq_pr.mdp 11-eq.mdp*
2. Comment "define = -DPOSRES" line in 11-eq.mdp with ";"
3. *gmx grompp -f 11-eq.mdp -c 10-1GFL_solvate_eq_pr.gro -p 08-1GFL_solvated.top -o 11-1GFL_solvated_eq.tpr -po 11-1GFL_solvated_eq_mdout.mdp*
4. *gmx mdrun -v -deffnm 11-1GFL_solvated_eq*

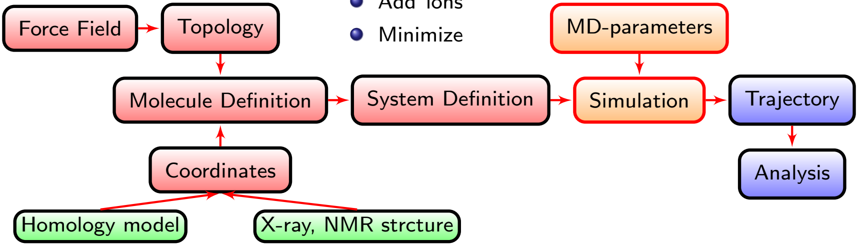
Analysis

1. *echo "Total-Energy" |gmx energy -f 11-1GFL_solvated_eq.edr -o 11-1GFL_solvated_eq-Total-Energy.xvg*
2. *xmgrace 11-1GFL_solvated_eq-Total-Energy.xvg*



Ready for a the production simulation

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



GFP in water - Production parameters

Production file: 12-md.mdp

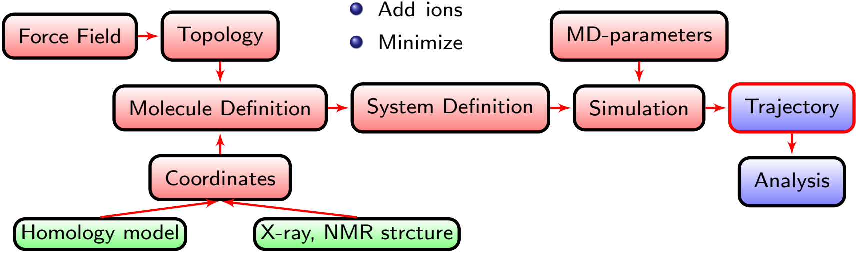
```
define = integrator = md
dt = 0.002
nsteps = 500000 ; 1ns
nstlog = 5000 ; 10ps
nstenergy = 5000 ; 10ps
nstxout-compressed = 5000 ; 10ps

cutoff-scheme = Verlet
nstlist = 20
rlist = 1.2
coulombtype = pme
rcoulomb = 1.2
coulomb-modifier = Potential-shift-Verlet
vdwtype = Cut-off
vdw-modifier = Potential-shift-Verlet
rvdw_switch = 1.0
rvdw = 1.2
```

Continuation: 12-md.mdp

```
tcoupl = nose-hoover
tc_grps = Protein water_and_ions
tau_t = 1.0 1.0
ref_t = 310.00 310.00

pcoupl = Parrinello-Rahman
pcoupltype = isotropic
tau_p = 5.0
compressibility = 4.5e-5
ref_p = 1.0

constraints = h-bonds
constraint_algorithm = LINCS

nstcomm = 100
comm_mode = linear

refcoord_scaling = com
```

System ready for the production simulation

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



GFP in water - Production simulation

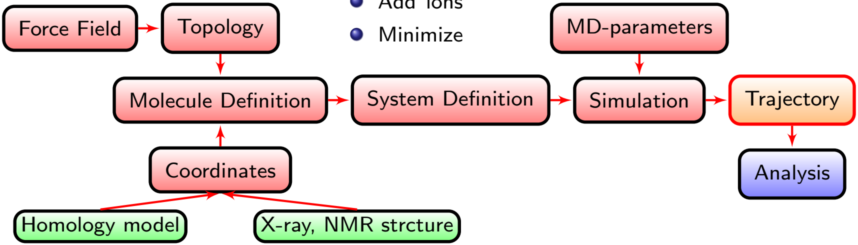
Production simulation GFP in water

1. *cp 11-eq.mdp 12-md.mdp*
2. Edit 12-md.mdp:
    1. tcoupl = nose-hoover
    2. pcoupl = Parrinello-Rahman
3. *gmx grompp -f 12-md.mdp -c 11-1GFL_sovated_eq.gro -p 08-1GFL_solvated.top -o 12-1GFL_solvated_md.tpr*

   -po 12-1GFL_solvated_md_mdout.mdp
4. *gmx mdrun -v -deffnm 12-1GFL_solvated_md*

Simulation ready - Are we done?

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



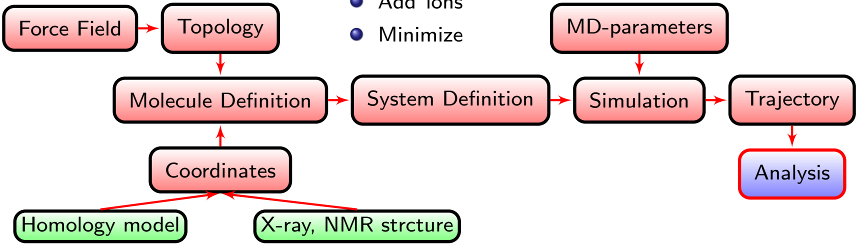
GFP in water - Process trajectory

Process trajectory to visualize in vmd

1. *gmx check -f 12-1GFL_solvated_md.xtc*
2. *gmx trjconv -s 12-1GFL_solvated_md.tpr -f 12-1GFL_solvated_md.xtc -o 12-1GFL_solvated_md_whole.xtc -pbc whole*
3. *vmd 12-1GFL_solvated_md.gro 12-1GFL_solvated_md_whole.xtc*

Simulation ready, time for analysis

- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



Analyzing simulation

Radius of gyration:

1.
```
echo ”Protein” | gmx gyrate -f 12-1GFL_solvated_md_whole.xtc
-s 12-1GFL_solvated_md.tpr -o 12-1GFL_solvated_md_ana_gyrate.xvg
```

Autocorrelation rotational diffusion:

1.
```
printf ”del 0-30 \n a3556|a2224 \n q \n” |gmx make_ndx -f 12-1GFL_solvated_md.tpr
-o 12-1GFL_solvated_md_ana_rotacf.ndx
```
2.
```
gmx rotacf -f 12-1GFL_solvated_md_whole.xtc -s 12-1GFL_solvated_md.tpr
-n 12-1GFL_solvated_md_ana_acf.ndx -o 12-1GFL_solvated_md_ana_rotacf.xvg -d
```

Radial distribution function of Na^+ around GFP

1.
```
gmx rdf -f 12-1GFL_solvated_md_whole.xtc -s 12-1GFL_solvated_md.tpr
-o 12-1GFL_solvated_md_ana_rdf_surf.xvg -surf mol -ref Protein -sel NA
```

Gromacs analysis tools I

anadock	Cluster structures from Autodock runs
anaeig	Analyze eigenvectors/normal modes
analyze	Analyze data sets
angle	Calculate distributions and correlations for angles and dihedrals
bar	Calculate free energy difference estimates through Bennett's acceptance ratio
bundle	Analyze bundles of axes, e.g., helices
chi	Calculate everything you want to know about chi and other dihedrals
cluster	Cluster structures
clustsize	Calculate size distributions of atomic clusters
confrms	Fit two structures and calculates the RMSD
covar	Calculate and diagonalize the covariance matrix
current	Calculate dielectric constants and current autocorrelation function
density	Calculate the density of the system
densmap	Calculate 2D planar or axial-radial density maps
densorder	Calculate surface fluctuations
dielectric	Calculate frequency dependent dielectric constants
dipoles	Compute the total dipole plus fluctuations
disre	Analyze distance restraints
distance	Calculate distances between pairs of positions

Gromacs analysis tools II

do_dssp	Assign secondary structure and calculate solvent accessible surface area
dos	Analyze density of states and properties based on that
dyecoupl	Extract dye dynamics from trajectories
dyndom	Interpolate and extrapolate structure rotations
enemat	Extract an energy matrix from an energy file
freevolume	Calculate free volume
gangle	Calculate angles
gyrate	Calculate the radius of gyration
h2order	Compute the orientation of water molecules
hbond	Compute and analyze hydrogen bonds
helix	Calculate basic properties of alpha helices
helixorient	Calculate local pitch/bending/rotation/orientation inside helices
hydorder	Compute tetrahedrality parameters around a given atom
mdmat	Calculate residue contact maps
mindist	Calculate the minimum distance between two groups
mk_angndx	Generate index files for 'gmx angle'
morph	Interpolate linearly between conformations
msd	Calculates mean square displacements
nmeig	Diagonalize the Hessian for normal mode analysis

Gromacs analysis tools III

nmens	Generate an ensemble of structures from the normal modes
nmtraj	Generate a virtual oscillating trajectory from an eigenvector
order	Compute the order parameter per atom for carbon tails
pairdist	Calculate pairwise distances between groups of positions
polystat	Calculate static properties of polymers
potential	Calculate the electrostatic potential across the box
principal	Calculate principal axes of inertia for a group of atoms
rama	Compute Ramachandran plots
rdf	Calculate radial distribution functions
rms	Calculate RMSDs with a reference structure and RMSD matrices
rmsdist	Calculate atom pair distances averaged with power -2, -3 or -6
rmsf	Calculate atomic fluctuations
rotacf	Calculate the rotational correlation function for molecules
rotmat	Plot the rotation matrix for fitting to a reference structure
saltbr	Compute salt bridges
sans	Compute small angle neutron scattering spectra
sasa	Compute solvent accessible surface area
saxs	Compute small angle X-ray scattering spectra

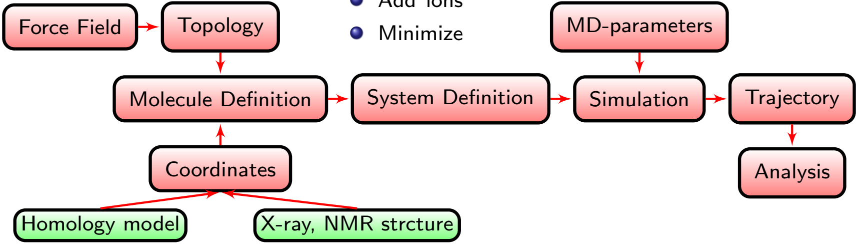
Gromacs analysis tools IV

sham	Compute free energies or other histograms from histograms
solvate	Solvate a system
sorient	Analyze solvent orientation around solutes
spatial	Calculate the spatial distribution function
spol	Analyze solvent dipole orientation and polarization around solutes
tcaf	Calculate viscosities of liquids
traj	Plot x, v, f, box, temperature and rotational energy from trajectories
trjorder	Order molecules according to their distance to a group
vanhove	Compute Van Hove displacement and correlation functions
velacc	Calculate velocity autocorrelation functions

Gromacs v5.1.4 has around 100 tools already at your service!!

GFP simulation complete

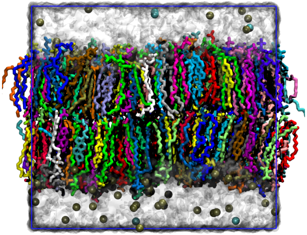
- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize

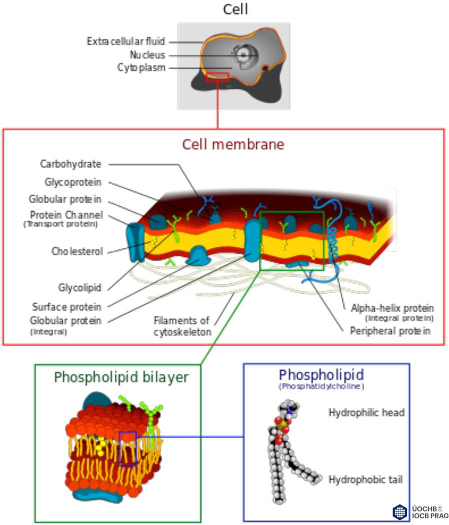


Task03: Membrane simulation using CHARMMGUI

Purpose of the task

- Show how tools can simplify the execution workflow of MD simulations
- Show another type of biological systems where MD can be applied



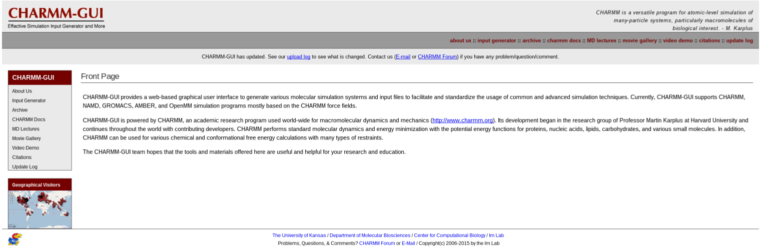


ÚOCHB IOCB PRAGUE

CHARMMGUI for gromacs

http://www.charmm-gui.org/

CHARMMGUI is a free web resource that will help you to generate input files for most common biologically relevant systems. It produces files compatible with Gromacs v5 and uses the modern charmm36 force field



CHARMM-GUI
Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation of many-particle systems, particularly macromolecules of biological interest. - M. Karplus

about us :: input generator :: archive :: charmm docs :: MD lectures :: movie gallery :: video demo :: citations :: update log

CHARMM-GUI has updated. See our upload log to see what is changed. Contact us (E-mail or CHARMM Forum) if you have any problem/question/comment.

CHARMM-GUI
About Us
Input Generator
Archive
CHARMM Docs
MD Lectures
Movie Gallery
Video Demo
Citations
Update Log
Geographical Visitors

Front Page

CHARMM-GUI provides a web-based graphical user interface to generate various molecular simulation systems and input files to facilitate and standardize the usage of common and advanced simulation techniques. Currently, CHARMM-GUI supports CHARMM, NAMD, GROMACS, AMBER, and OpenMM simulation programs mostly based on the CHARMM force fields.

CHARMM-GUI is powered by CHARMM, an academic research program used world-wide for macromolecular dynamics and mechanics (http://www.charmm.org). Its development began in the research group of Professor Martin Karplus at Harvard University and continues throughout the world with contributing developers. CHARMM performs standard molecular dynamics and energy minimization with the potential energy functions for proteins, nucleic acids, lipids, carbohydrates, and various small molecules. In addition, CHARMM can be used for various chemical and conformational free energy calculations with many types of restraints.

The CHARMM-GUI team hopes that the tools and materials offered here are useful and helpful for your research and education.

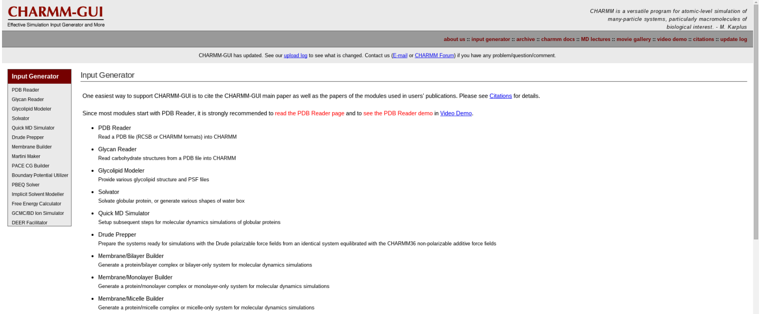
The University of Kansas / Department of Molecular Biosciences / Center for Computational Biology / Im Lab

Problems, Questions, & Comments? CHARMM Forum or E-Mail / Copyright(c) 2006-2015 by the Im Lab

CHARMMGUI is very efficient to generate biological membranes

The Input Generator

- It provides among other options the possibility to make a personalized membrane.
- Its output can be easily adapted to be used with other force fields not just charmm36.



CHARMM-GUI
Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation of many-particle systems, particularly macromolecules of biological interest. - M. Karplus

about us :: input generator :: archive :: charmm docs :: MD lectures :: movie gallery :: video demo :: citations :: update log

CHARMM-GUI has updated. See our upload log to see what is changed. Contact us (E-mail or CHARMM Forum) if you have any problem/question/comment.

Input Generator

One easiest way to support CHARMM-GUI is to cite the CHARMM-GUI main paper as well as the papers of the modules used in users' publications. Please see Citations for details.

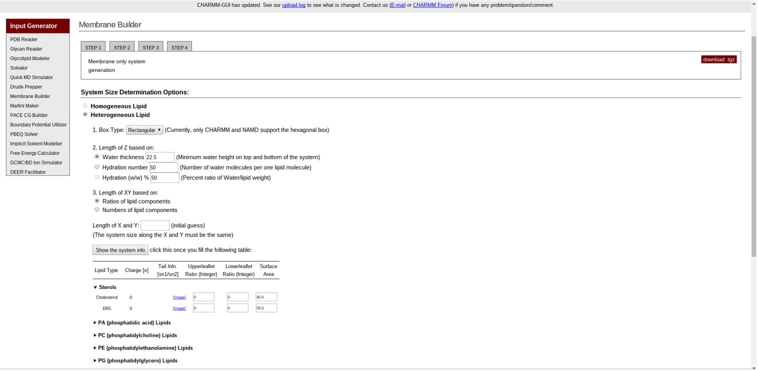
Since most modules start with PDB Reader, it is strongly recommended to read the PDB Reader page and to see the PDB Reader demo in Video Demo.

- PDB Reader
  Read a PDB file (RCSB or CHARMM formats) into CHARMM
- Glycan Reader
  Read carbohydrate structures from a PDB file into CHARMM
- Glycolipid Modeler
  Provide various glycolipid structure and PSF files
- Solvator
  Solvate globular protein, or generate various shapes of water box
- Quick MD Simulator
  Setup subsequent steps for molecular dynamics simulations of globular proteins
- Drude Prepper
  Prepare the systems ready for simulations with the Drude polarizable force fields from an identical system equilibrated with the CHARMM36 non-polarizable additive force fields
- Membrane/Bilayer Builder
  Generate a protein/bilayer complex or bilayer-only system for molecular dynamics simulations
- Membrane/Monolayer Builder
  Generate a protein/monolayer complex or monolayer-only system for molecular dynamics simulations
- Membrane/Micelle Builder
  Generate a protein/micelle complex or micelle-only system for molecular dynamics simulations

Simulation of a model plasma membrane

Produce a plasma membrane with the following parameters

- upper leaflet 60 POPC/60 CHOL /60 PSM
- lower leaflet 65 POPE/60 CHOL /60 POPS
- default options



CHARMM-GUI has updated. See our upload log to see what is changed. Contact us (E-mail or CHARMM Forum) if you have any problem/question/comment.

Membrane Builder

Membrane only system generation

System Size Determination Options:

○ **Homogeneous Lipid**
● **Heterogeneous Lipid**

1. Box Type: Rectangular ▾ (Currently, only CHARMM and NAMD support the hexagonal box)

2. Length of Z based on:
● Water thickness 22.5 (Minimum water height on top and bottom of the system)
○ Hydration number 50 (Number of water molecules per one lipid molecule)
○ Hydration (w/w) % 50 (Percent ratio of Water/lipid weight)

3. Length of XY based on:
● Ratios of lipid components
○ Numbers of lipid components

Length of X and Y: (initial guess)
(The system size along the X and Y must be the same)

Show the system info click this once you fill the following table:

Lipid Type	Charge [e]	Tail Info. [sn1/sn2]	Upperleaflet Ratio (Integer)	Lowerleaflet Ratio (Integer)	Surface Area
▾ **Sterols**					
Cholesterol	0	[Image]	0	0	40.0
ERG	0	[Image]	0	0	55.0

▸ **PA (phosphatidic acid) Lipids**

▸ **PC (phosphatidylcholine) Lipids**

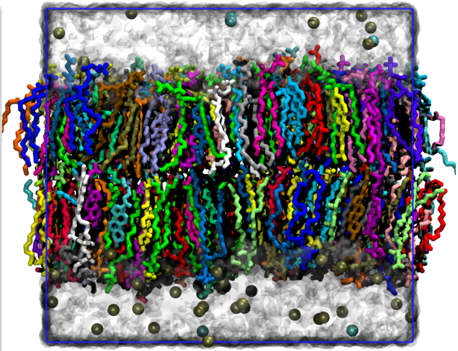
▸ **PE (phosphatidylethanolamine) Lipids**

▸ **PG (phosphatidylglycero) Lipids**

Run a membrane generated by CHARMMGUI

Running CHAMMGUI output

1. *cd ../task03-CHARMMGUI-membrane*
2. *tar -xzvf charmm-gui.tgz*
3. *cd charmm-gui/gromacs*
4. *sed -i 's:gmx_d:gmx:g' README*
   - This step is not really needed if you have installed the double precision version of Gromacs which I recommend for minimization
5. *csh README*



Analyzing membrane simulation

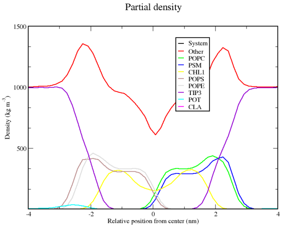
Concatenating trajectories

1.
   ```
   gmx trjcat -f step7*.trr -o total.xtc
   ```

Density profiles along the axis perpendicular membrane plane

1.
   ```
   echo "0 0 1 2 3 4 5 6 7 8 9" | gmx density -f total.xtc -s step7_1.tpr -o dens.xvg -center -d Z -ng 10
   ```

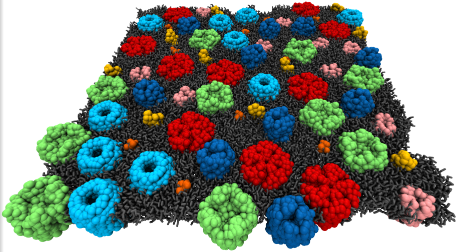
2.
   ```
   xmgrace -free -nxy dens.xvg
   ```



Task04: Simulation of a membrane protein using Martini

Purpose of the task

- Show how tools can simplify the execution workflow of MD simulations
- Show a more biologically relevant system
- Show a coarse grained model for biological systems



Create a membrane protein using martinize.py

Appendix1: Important commands

Gromacs

- *gmx command -h*
- Prepare simulation files:
  - *gmx pdb2gmx*
    (protein itp file form pdb)
  - *gmx grompp*
    (tpr file from top, gro and mdp)
- Run simulation:
  - *gmx mdrun -v -deffnm TPRFILE*
    (run MD from tpr file)
- Post-processing trajectory:
  - *gmx energy -f EDRFILE*
    (Output trajectories observables)
  - *gmx trjconv*
    (process trajectory for analysis)

xmgrace: (Plotting xvg files)

xmgrace -free -nxy XVGFILE

vmd: (Vissualizing trajectories

vmd GROFILE XTCFILE_WHOLE