Introduction to Gromacs 5

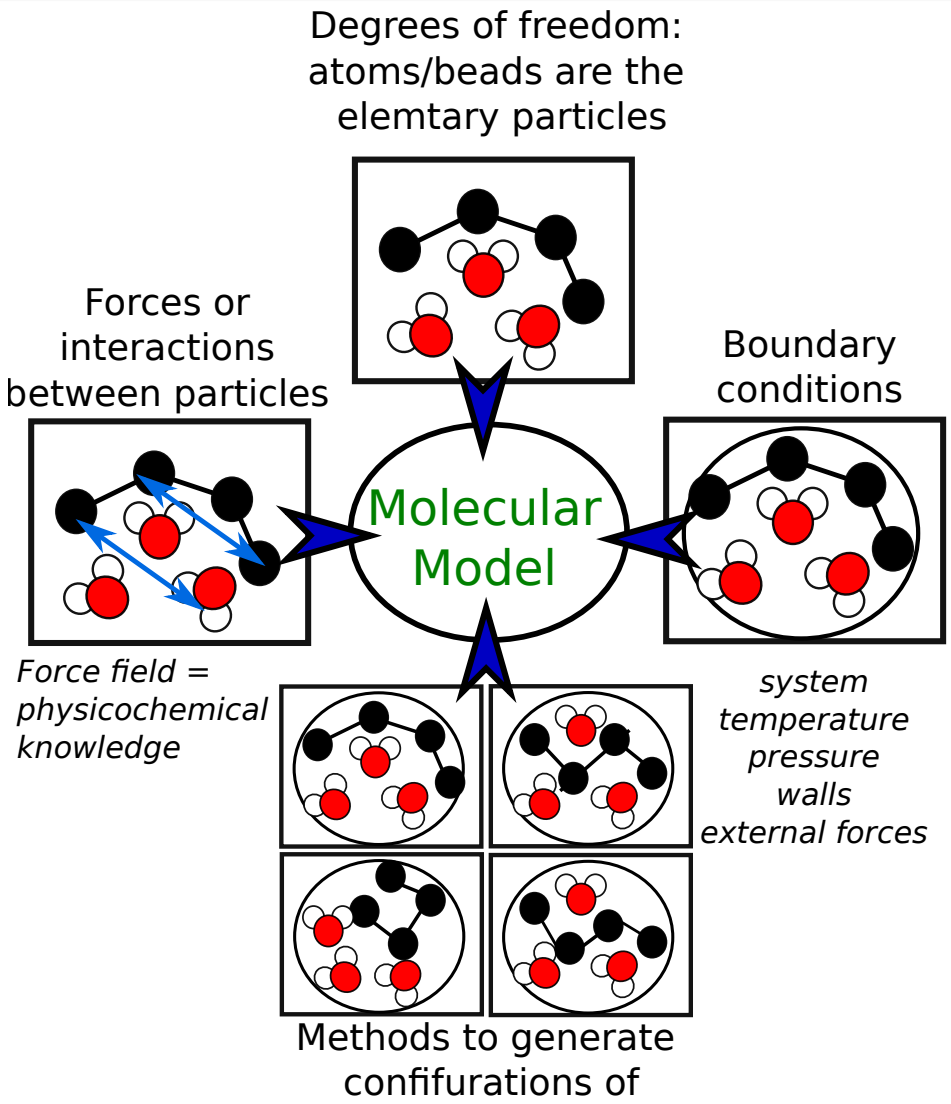
Hector Martinez-Seara

IOCB and TUT

23/11/2015

1. Molecular mechanics (MD)
2. What is gromacs and when to use it?
3. How to use gromacs?
4. Membrane simulation using CHARMMGUI
5. Analysis trajectories

How "classical" molecular dynamics work?



- 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).

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

What is a force Field?

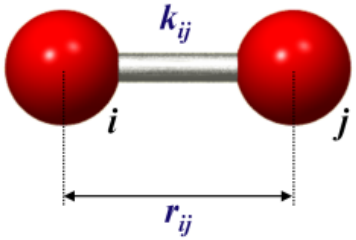
- Force fields are normally assumed to consist of pairwise-additive potentials
  - More accurate descriptions may involve the use of three-body terms or non-additive potentials.
- A compromise between their accuracy and simplicity is essential. Most force fields can be described as:

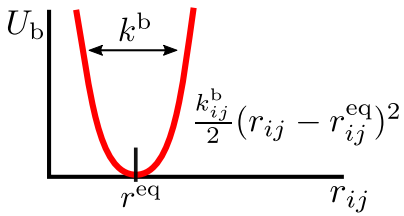
$$U = U_{bond} + U_{ang} + U_{tors} + U_{vdw} + U_{coul} \quad (2)$$

  - 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.

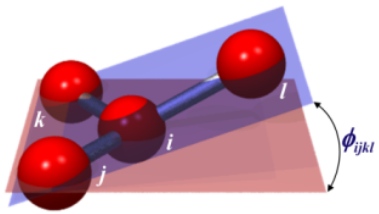
Typical potentials used in biosystems

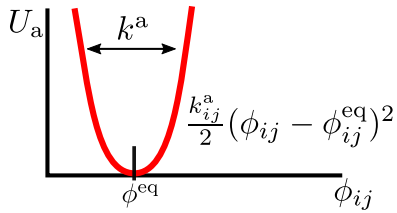
Bond



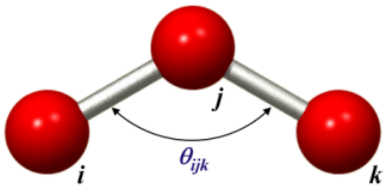


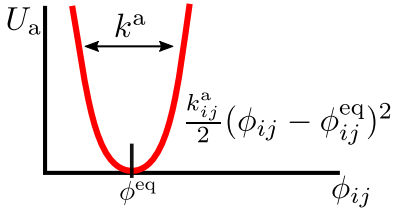
Improper



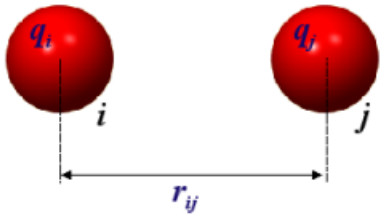


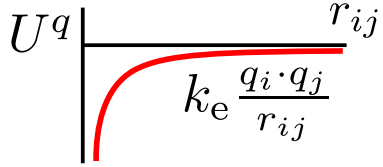
Angle



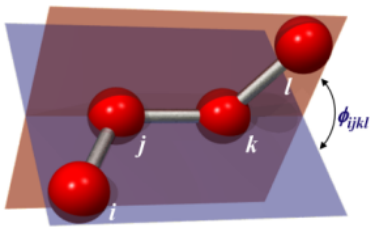


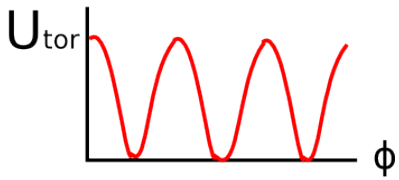
Coulomb



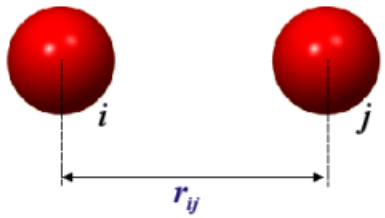


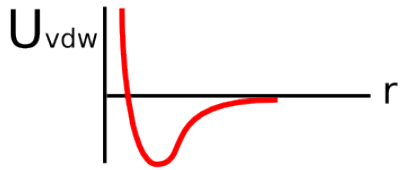
Dihedral





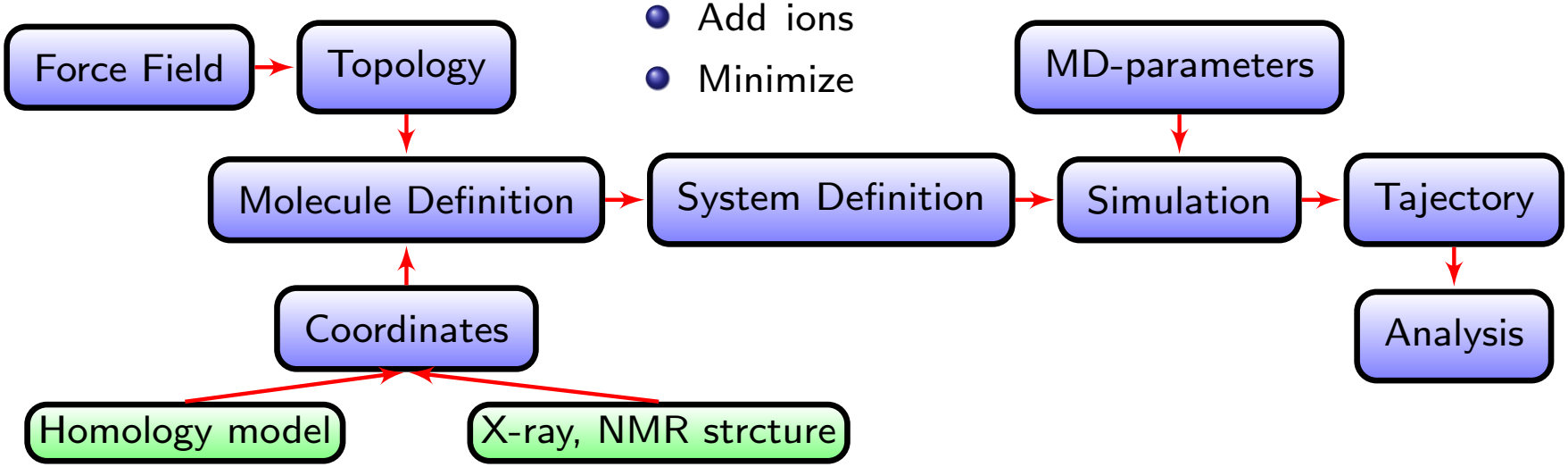
Van der Waals





Typical Molecular dynamics work flow

- 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 v5.1.1 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:

...

Exercice: water box - Preparation

preparation of the working folder

1. *mkdir -p 01-Exercice_water/*
2. *cd 01-Exercice_water/*

Exercise: Let's make a box of TIP3P/SPC water

Generating water box:

- Using gmx solvate generate a TIP3P water box of 5x10x7 namometers
- Visualize it in vmd: *vmd out.gro*

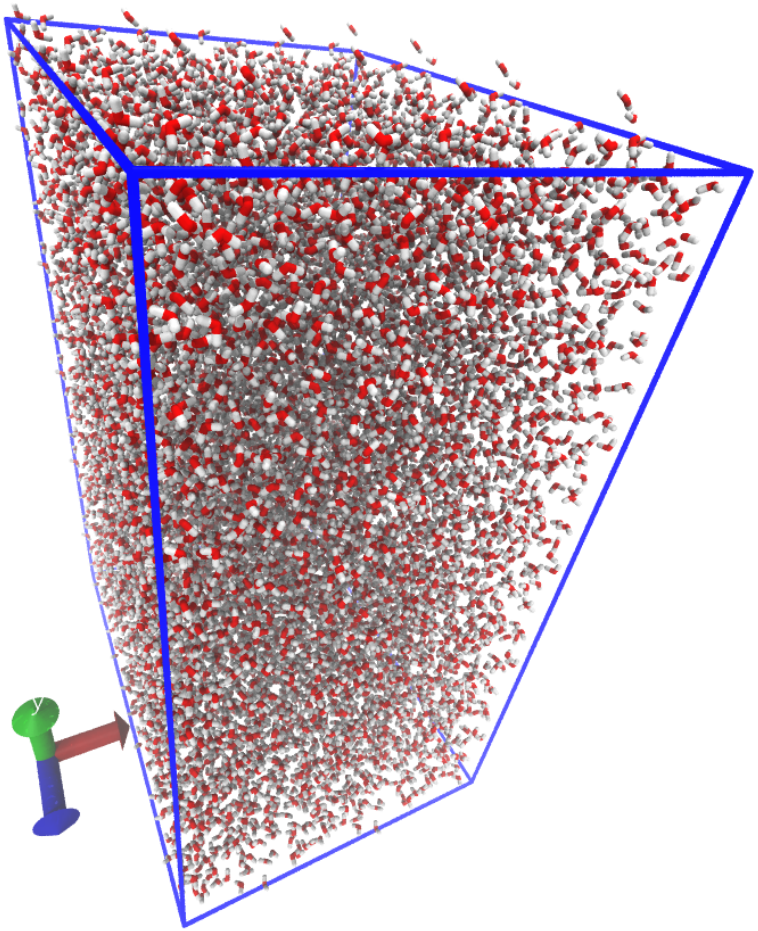
Exercise: Let's make a box of TIP3P/SPC water

Generating water box:

- Using gmx solvate generate a TIP3P water box of 5x10x7 namometers
- Visualize it in vmd: *vmd out.gro*

Solution:

1. *gmx solvate -cs -box 5 10 7*
2. *vmd out.gro*



How to simulate a protein in gromacs - I

Obtaining protein structure

RCSB Protein Data Bank

http://www.rcsb.org/

Exercice: Simulation Green Fluorecence protein - Preparation

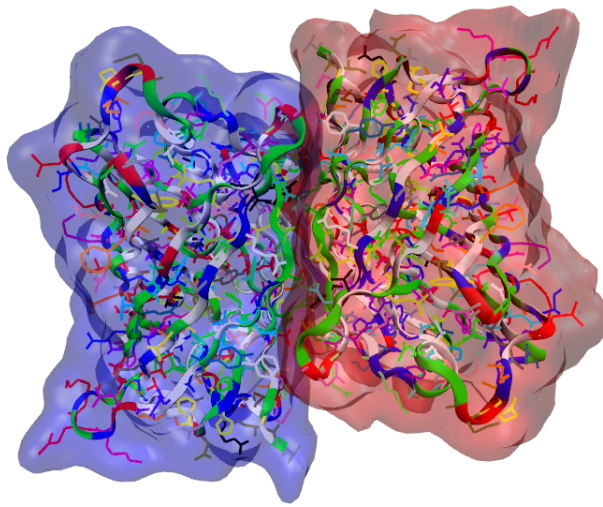
preparation of the working folder

1. *mkdir -p 02-Exercice_GFP/01-GFP_topology*
2. *cd 02-Exercice_GFP/01-GFP_topology*

Exercise: Obtaining the structure of the green fluorescence protein (GFP)

Obtaining green fluorescence protein (GFP) - 1GFL

- Download 1GFL from http://www.rcsb.org/ (save as 00-1GFL.pdb)
- Open 00-1GFL.pdb in your favorite editor
- Visualize it in vmd

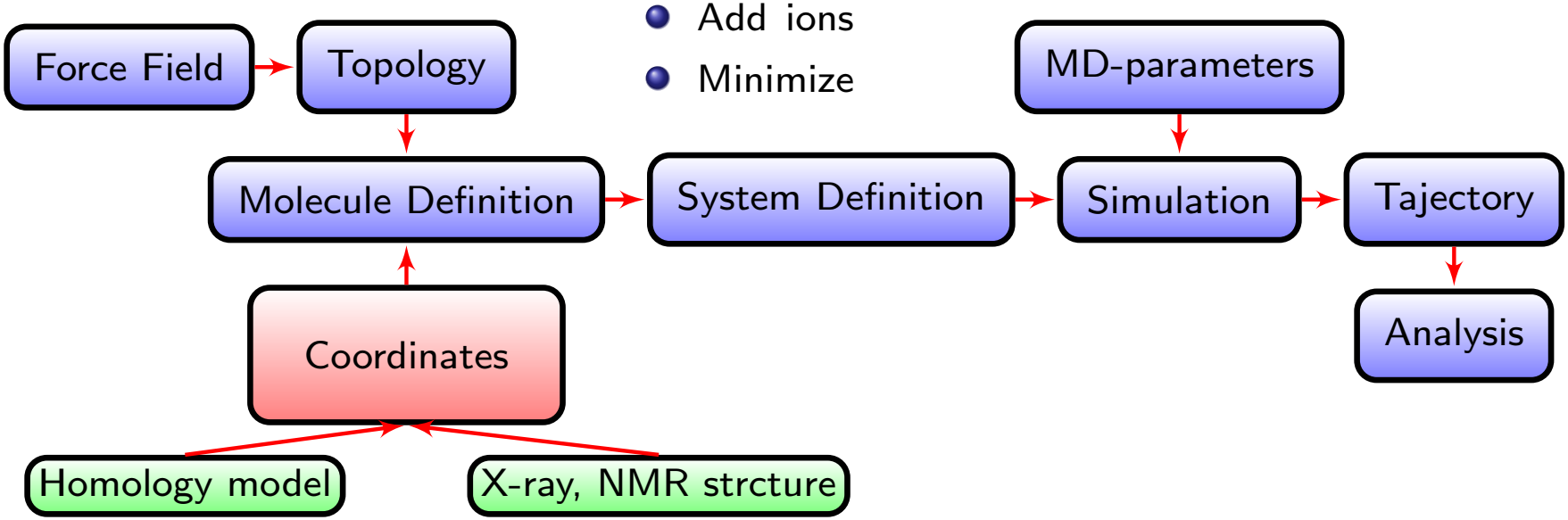


Observations:

1. There are two different chains
2. There are few crystallized waters
3. read REMARKS 350 365 470!!

Typical Molecular dynamics work flow

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



Selection of a decent force field

All atom

charmm36, amber99sb-ildl, oplsaa, charmm27

United atom

GROMOS96 54a7

Coarse-grained

Martini

Not good but seldom used.

Never good – You MUST know what are you doing.

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

Exercise: GFP topology - Raw attempt

Generate topology for 00-1GFL.pdf

Try to generate the topology for 1GFL using *pdb2gmx*

Exercise: GFP topology - Raw attempt

Generate topology for 00-1GFL.pdf

Try to generate the topology for 1GFL using *pdb2gmx*

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 andor residue

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

Exercise: 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 >02-1GFLA_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.

Exercise: GFP topology - pdb4amber

What is *pdb4amber*

- Tool shiped with ambertools which prepare pdbs for further processing using ambertools, *e.g. tleap*.
- 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 02-1GFLA_raw.pdb -o 03-1GFLA_pdb4amber.pdb –reduce –dry

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

Exercise: GFP topology - Correct missing atoms

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

1. *tleap*
2. *source leaprc.ff14SB*
3. *x = loadPdb 03-1GFLA_pdb4amber.pdb*
4. *savepdb x 04-1GFLA.pdb*
5. *quit*

- This method will not correct for missing residues. For this you can use Modeller.

Exercise: GFP topology - Generate working topology

Generate topology GFP

Try to generate the topology for 04-1GFLA.pdb using *pdb2gmx*

Exercise: GFP topology - Generate working topology

Generate topology GFP

Try to generate the topology for 04-1GFLA.pdb using *pdb2gmx*

Normal execution - Problems with hydrogens (FAILS!!)

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

Normal execution but ignoring pdb hydrogens (WORKS!!)

gmx pdb2gmx -f 04-1GFLA.pdb -o 04-1GFLA.gro -i 04-1GFLA_posres.itp -p 04-1GFLA.top -n 04-1GFLA.ndx -ignh

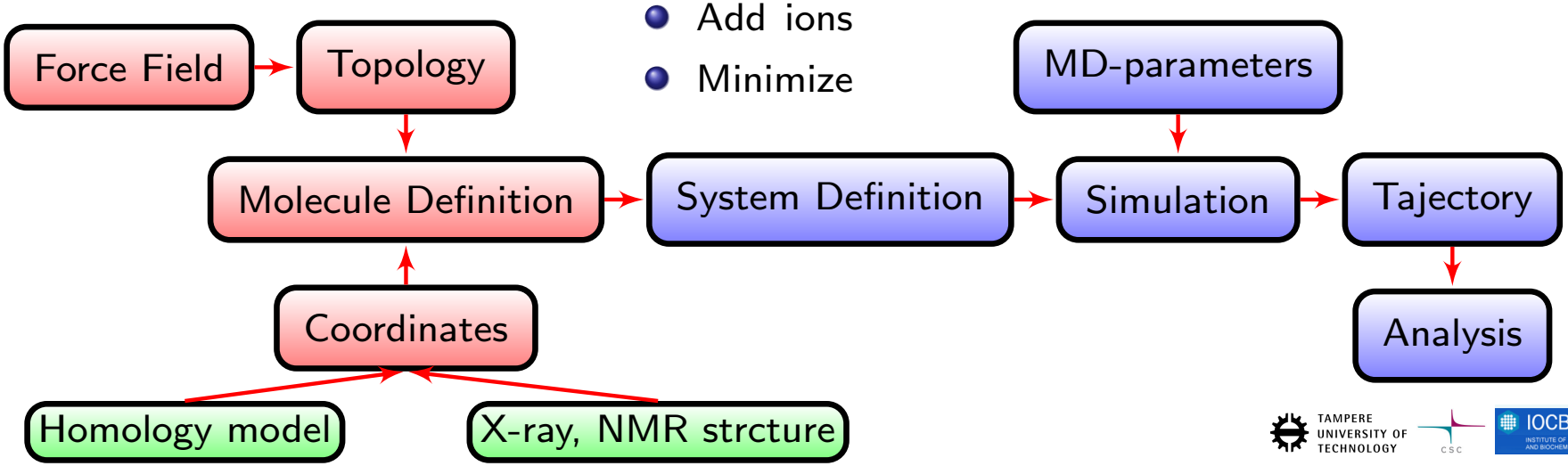
- Notice: Total charge -5.000 e

Now we know how to generate protein topologies

– – – – – – – – – – – *PLEASE NOTE* – – – – – – – – – – –

You have successfully generated a topology from: 04-1GFLA.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 topology files look like?

The top files

#include "amber99sb-ildn.ff/forcefield.itp" ← Forcefield definition

#include "1GFLA.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
1GFLA                 1 ← Number of proteins
SOL                1000 ← Number of waters (We do not have any yet)
```

Exercice: GFP topology - Prepare system top file

Prepare system topology file: system_nowater.top

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

#include "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: You can use 04-1GFLA.top as template. Just remove molecule section.

How do itp files look like?

The itp files - Topologies of Molecules

[moleculetype]
; 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
; residue 1 ALA rtp NALA q +1.0
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 ← *e.g.* last number one means simple harmonic potential in this case
...
[angles]
; ai aj ak funct c0 c1 c2 c3
2 1 3 1 ← The lack of explicit parameters *c0 c1 ...* means that defaults from #include "amber99sb-ildn.ff/forcefield.itp" 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 "04-1GFLA_posres.itp" ← Restrains file for our protein"
#endif

Exercice: GFP topology - Extract 1GFL molecule topology file

Prepare GFP topology file: 1GFL.itp

- Extract the 1GFL.itp from 04-1GFLA.top
  - *cp 04-1GFLA.top 1GFL.itp*
  - Remove until "[moleculetype]" from 1GFL.itp
  - Change name "[Moleculetype]" from "Protein" to "1GFL"
  - Remove from "; Include water topology from 1GFL.itp" until the end of the file
  - *cp 04-1GFLA_posres.itp 1GFL_posres.itp*

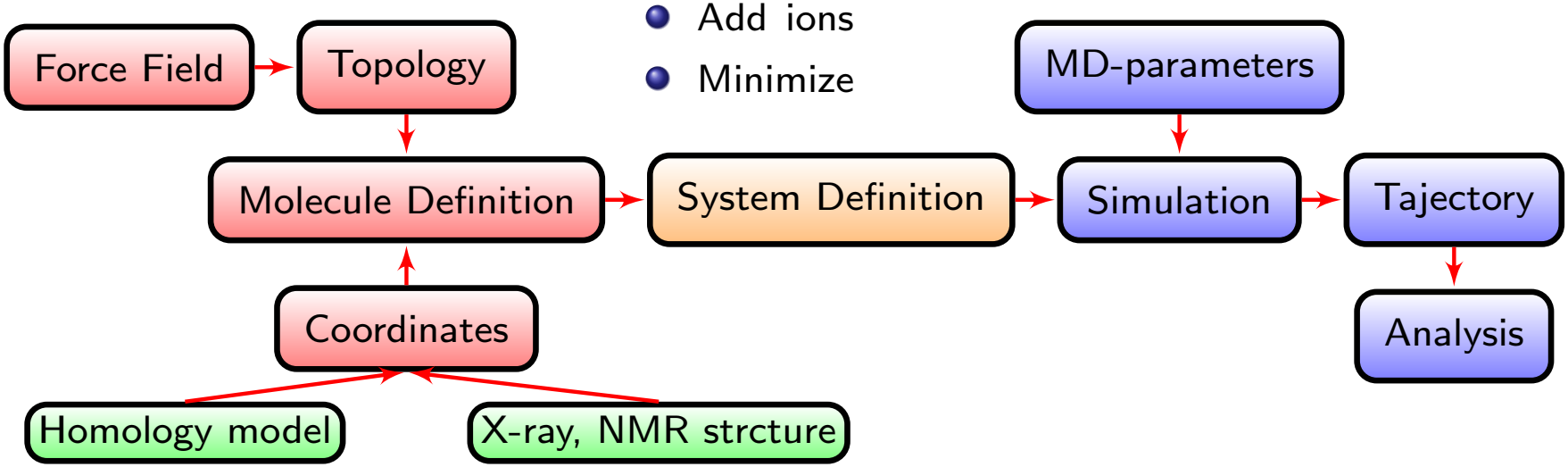
Exercice: GFP topology - The 1GFL.itp topology file

1GFL.itp

```
[ 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 "1GFL_posres.itp"
#endif
```

Now we know how to generate protein topologies

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



Exercice: GFP topology - Construct simulation system

Prepare system definition folder

1. *mkdir ../02-system_preparation*
2. *cp 1GFL.itp ../02-system_preparation*
3. *cp 1GFL_posres.itp ../02-system_preparation*
4. *cp system_nowater.top ../02-system_preparation/00-system_nowater.top*
5. *cp 04-1GFLA.gro ../02-system_preparation/00-system_nowater.gro*
6. *cd ../02-system_preparation/*

Exercice 01-02: GFP in water - Solvate GPF

Immerse the protein in a box of water

- *gmx editconf -f 00-system_nowater.gro -o 01-system_nowater_center.gro -c -box 6 6 6*
- *cp 00-system_nowater.top 02-system_water.top*
- *gmx solvate -cp 01-system_nowater_center.gro -p 02-system_water.top -o 02-system_water.gro -cs*
- vmd 02-system_water.gro

Exercice 01-02: GFP in water - Minimize GPF in water

A typical minimization mdp file: 03-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"

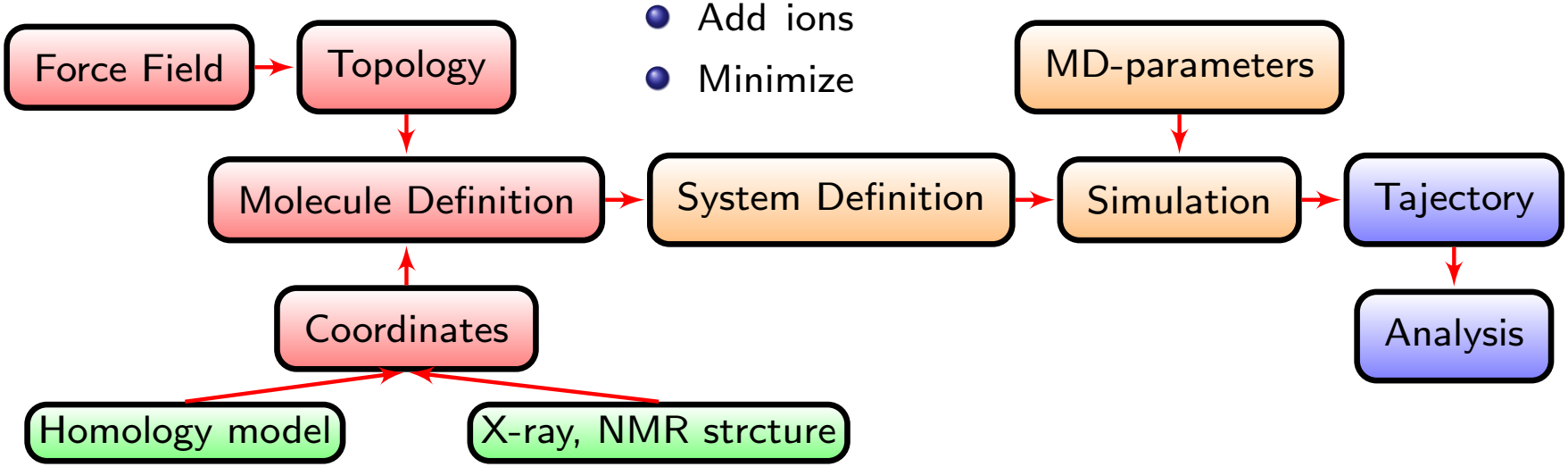
Exercice 01-02: GFP in water - Minimize GPF in water

Our first simulation a minimization

1. *gmx grompp -f 03-min.mdp \← Minimization parameters*
   -p 02-system_water.top \← Topology system
   -c 02-system_water.gro \← Coordinates atoms
   -o 03-system_water_min.tpr \← Output file for mdrun
   -po 03-system_water_min_mdout.mdp ← Full list minimizaton parameters)
2. *gmx mdrun -deffnm 03-system_water_min -v*
3. *vmd 03-system_water_min.gro*

What to do after adding waters and minimize?

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



Exercice 01-02: GFP in water - Neutralization of excess charges and adjustment medium ionic force

What was this note about

- *gmx grompp -f 03-min.mdp -p 02-system_water.top -c 02-system_water.gro -o 03-system_water_min.tpr -po 03-system_water_min_mdout.mdp*

  NOTE 1 [file 02-system_water.top, line 20]:

  System has non-zero total charge: -5.000000

- It means that our protein has more negative than positive groups. 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.* Na^+Cl^- ions.

Exercice 01-02: 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 02-system_water.top 04-system.top*
  - *gmx grompp -f 03-min.mdp -c 03-system_water_min.gro -p 02-system_water.top -o 04-system_genion.tpr -po 04-system_genion_mdout.mdp*
- Output coordinate file should be called "system_genion.gro"

Exercice 01-02: 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 02-system_water.top 04-system.top*
  - *gmx grompp -f 03-min.mdp -c 03-system_water_min.gro -p 02-system_water.top -o 04-system_genion.tpr -po 04-system_genion_mdout.mdp*
- Output coordinate file should be called "system_genion.gro"

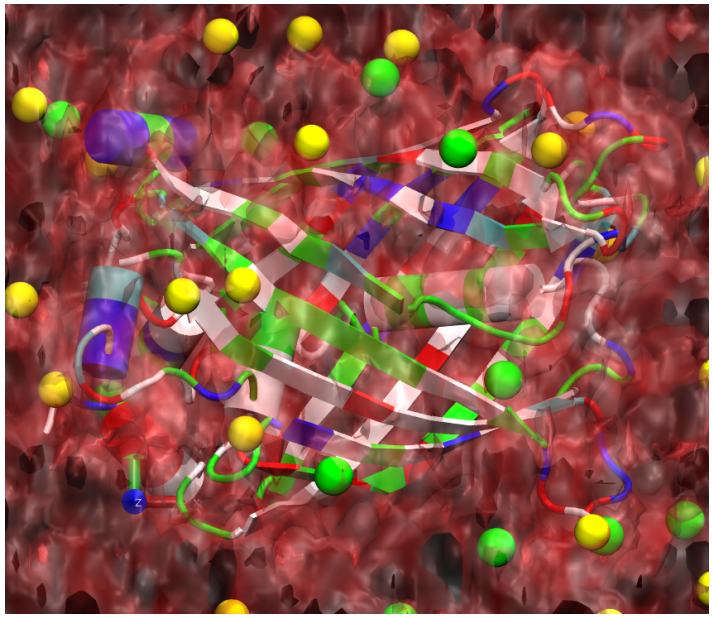
Solution

- *gmx genion -s 04-system_genion.tpr -p 04-system.top -o 04-system_genion.gro -neutral -conc 0.15*
  - Note: It does not really provide 150 mMol!!

Exercice 01-02: GFP in water - Minimize GPF system

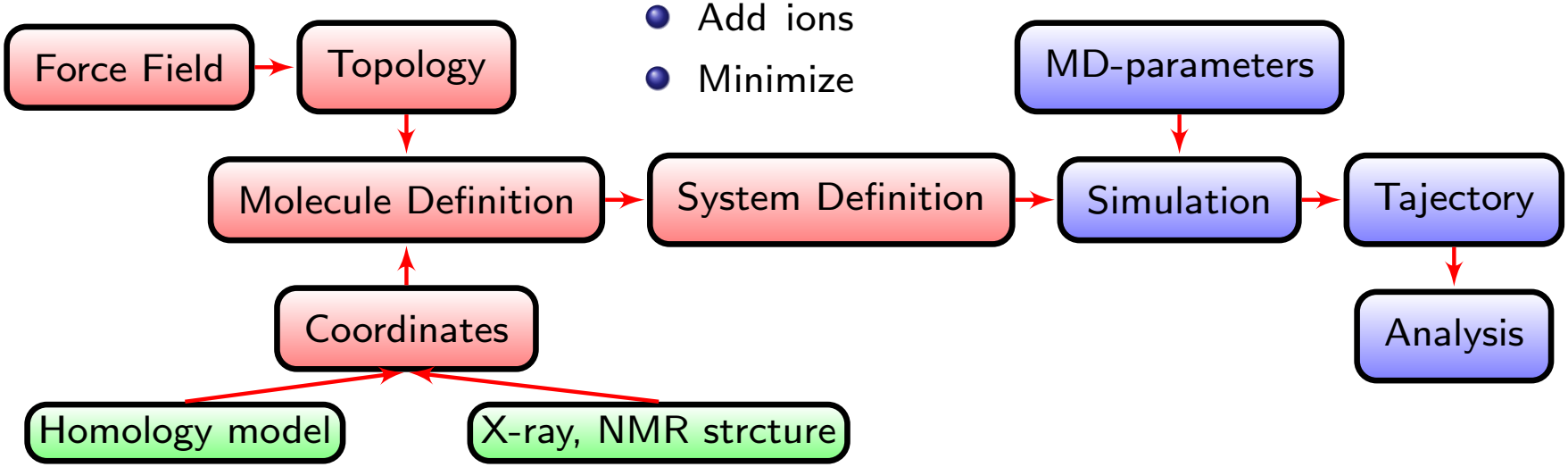
Minimize system after ions additions

1. *gmx grompp -f 03-min.mdp*
   -p 04-system.top
   -c 04-system_genion.gro
   -o 05-system_min.tpr
   -po 05-system_min_mdout.mdp
2. *gmx mdrun -deffnm 05-system_min -v*
3. *vmd 05-system_min.gro*



System ready for equilibration using MD

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



Exercice 01-02: GFP in water - Equilibration parameters

Equilibration file: 06-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: 06-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

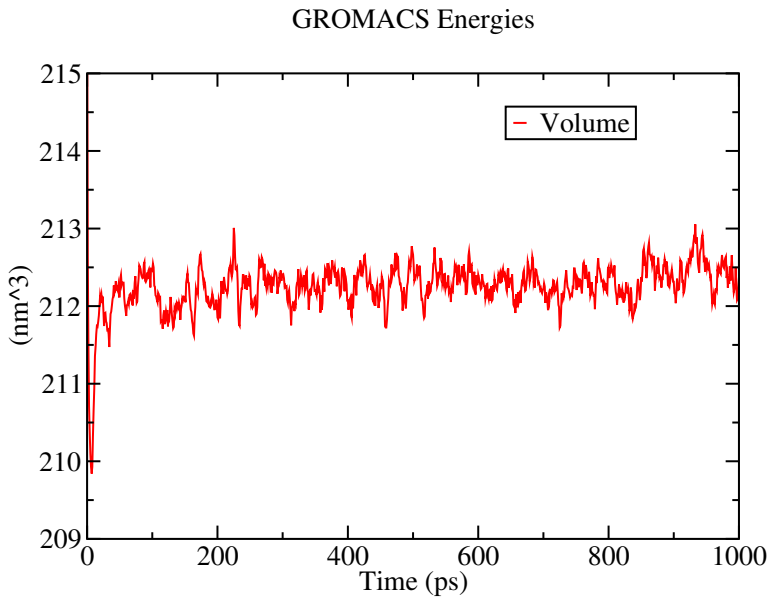
nstcomm = 100
comm_mode = linear

refcoord_scaling = com

Exercice 01-02: GFP in water - Equilibration I

Equilibration restrainded GFP in water

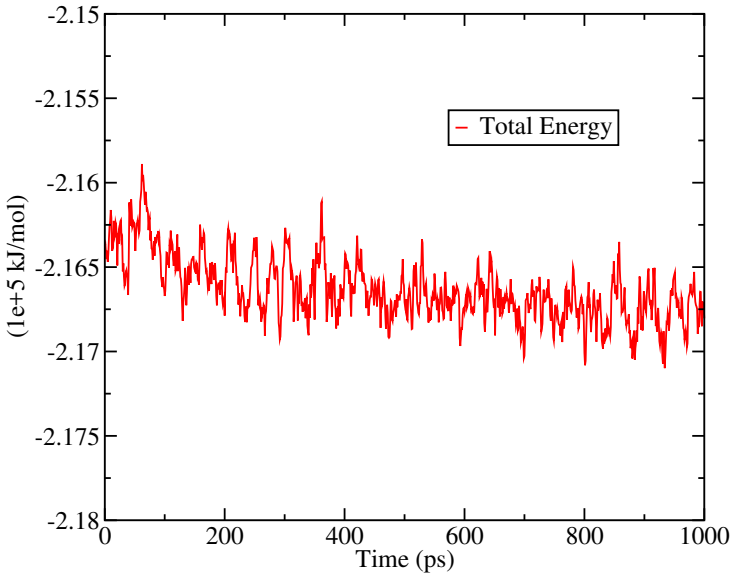
1. *gmx grompp -f 06-eq_pr.mdp*
   -c 05-system_min.gro
   -p 04-system.top
   -o 06-system_eq_pr.tpr
   -po 06-system_eq_pr_mdout.mdp
2. *gmx mdrun -v -deffnm 06-system_eq_pr*
3. *echo "Volume" |gmx energy*
   -f 06-system_eq_pr.edr
   -o 06-system_eq_pr-Volume.xvg
4. *xmgrace 06-system_eq_pr-Volume.xvg*



Exercice 01-02: GFP in water - Equilibration II

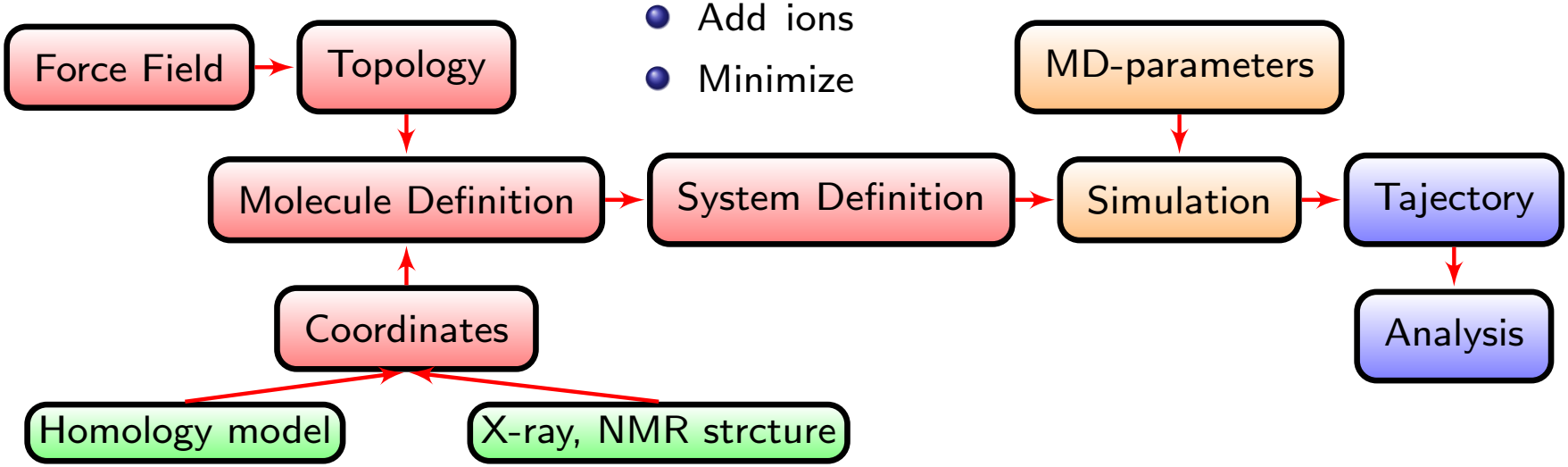
Equilibration free GFP in water

1. *cp 06-eq.mdp 07-eq.mdp*
2. Comment "define = -DPOSRES" line in 07-eq.mdp with ";"
3. Equilibration - no restrains:
   1. *gmx grompp -f 07-eq.mdp -c 06-system_eq_pr.gro -p 04-system.top -o 07-system_eq.tpr -po 07-system_eq_mdout.mdp*
   2. *gmx mdrun -v -deffnm 07-system_eq*
4. *echo "Total-Energy" |gmx energy -f 07-system_eq.edr -o 07-system_eq-Total-Energy.xvg*
5. *xmgrace 07-system_eq-Total-Energy.xvg*



Equilibration of the system

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



Exercice 01-02: GFP in water - Production parameters

Production file: 08-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: 08-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

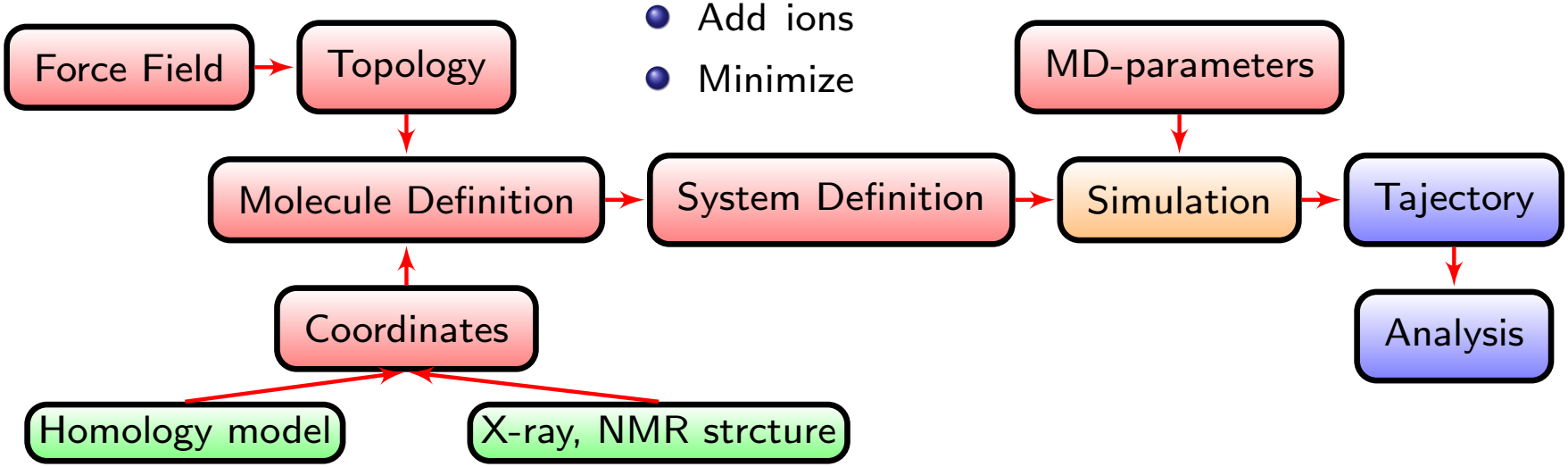
constraints = h-bonds
constraint_algorithm = LINCS

nstcomm = 100
comm_mode = linear

refcoord_scaling = com

System ready for simulation

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



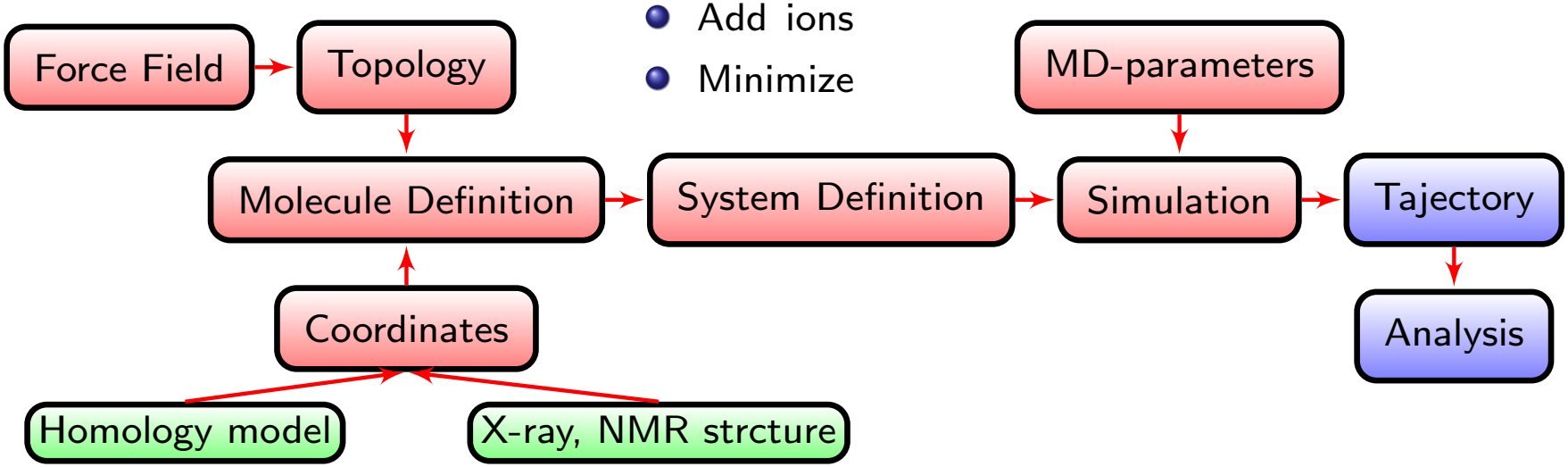
Exercice 01-02: GFP in water - Production simulation

Production simulation GFP in water

1. *cp 07-eq.mdp 08-md.mdp*
2. Edit 08-md.mdp:
   1. tcoupl = nose-hoover
   2. pcoupl = Parrinello-Rahman
3. *gmx grompp -f 08-md.mdp -c 07-system_eq.gro -p 04-system.top -o 08-system_md.tpr -po 08-system_md.mdp*
4. *gmx mdrun -v -deffnm 08-system_md*

System ready for simulation

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



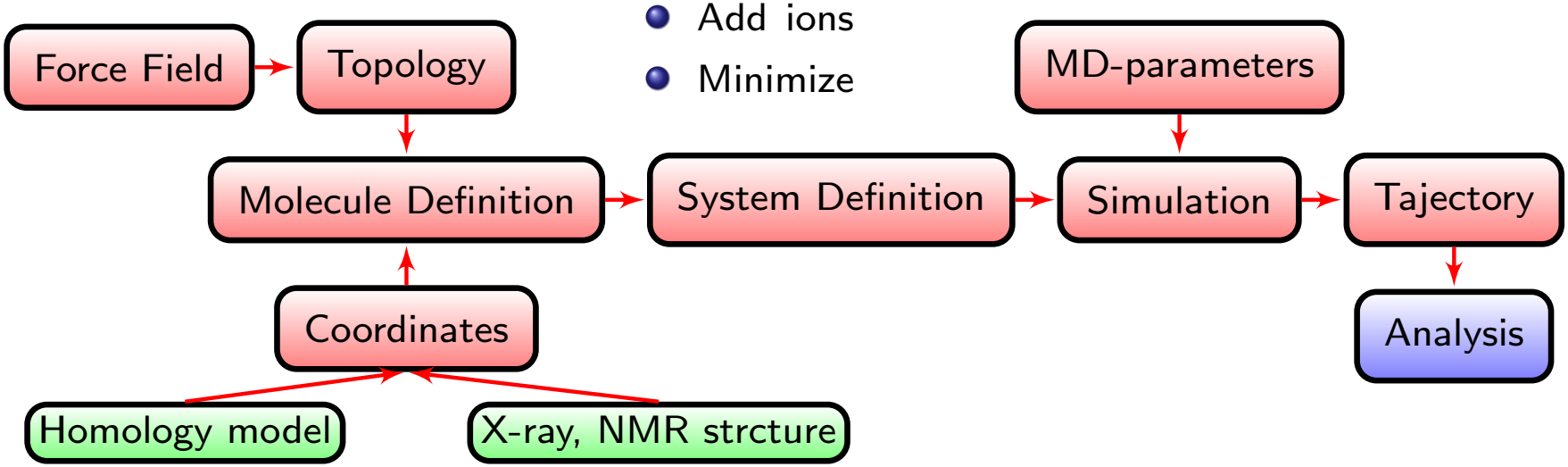
Exercice 01-02: GFP in water - Process trajectory

Process trajectory to visualize in vmd

1. *gmx check -f 08-system_md.xtc*
2. *gmx trjconv -s 08-system_md.tpr -f 08-system_md.xtc -o 08-system_md_whole.xtc -pbc whole*
3. *vmd 08-system_md.gro 08-system_md_whole.xtc*

System ready for simulation

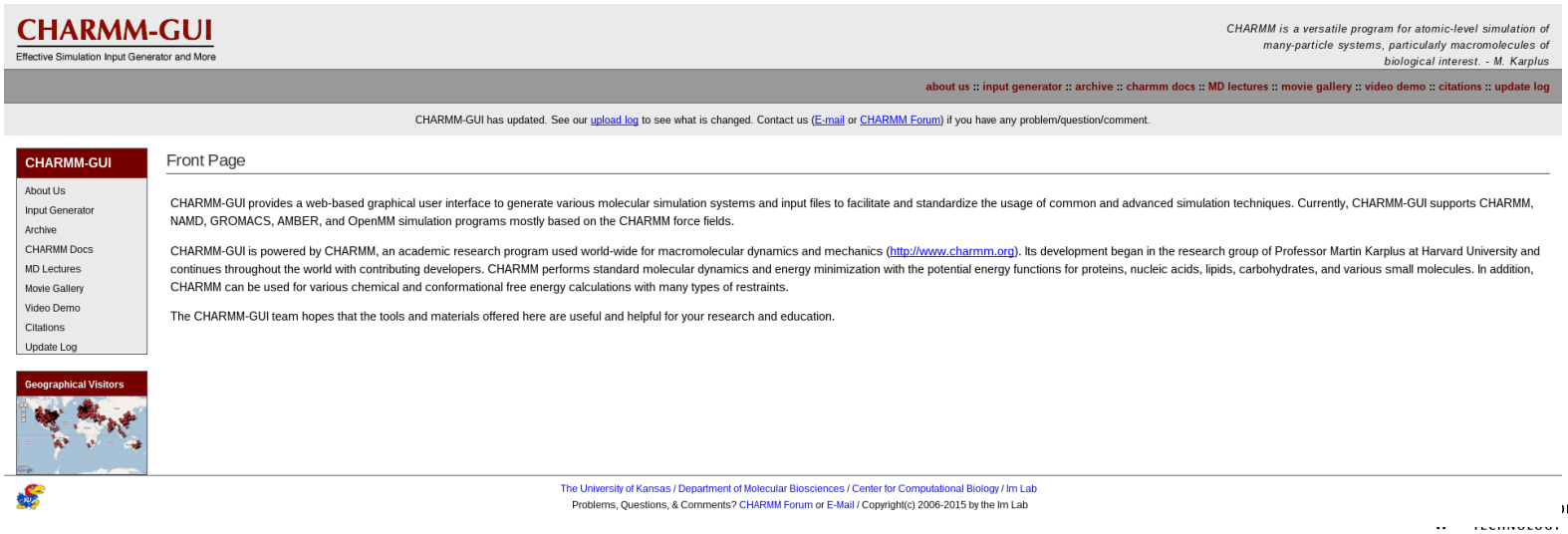
- Define unit cell
- Assemble the system
- Add solvent
- Add ions
- Minimize



CHARMMGUI for gromacs

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

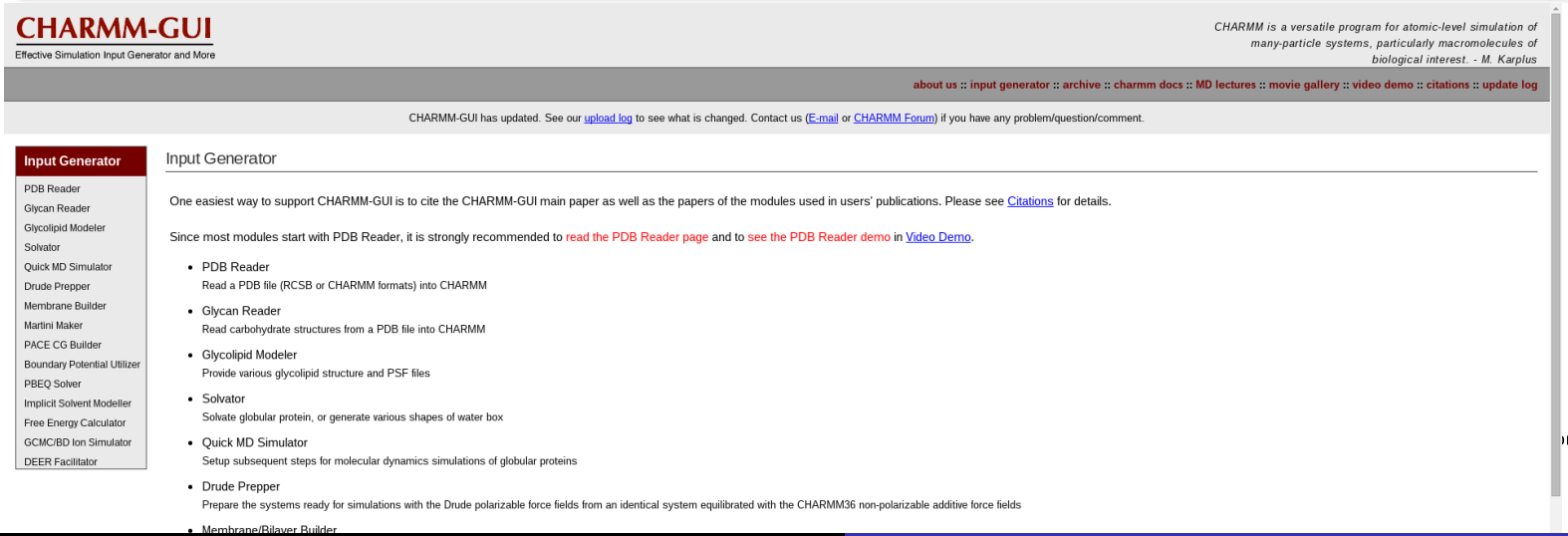
CHARMMGUI is a free web resource that will help you to generate input files for most common biologically relevant systems. It produces files compatible with Gromacs v5 and uses the modern charmm36 force field.



CHARMMGUI is very efficient to generate biological membranes

The Input Generator

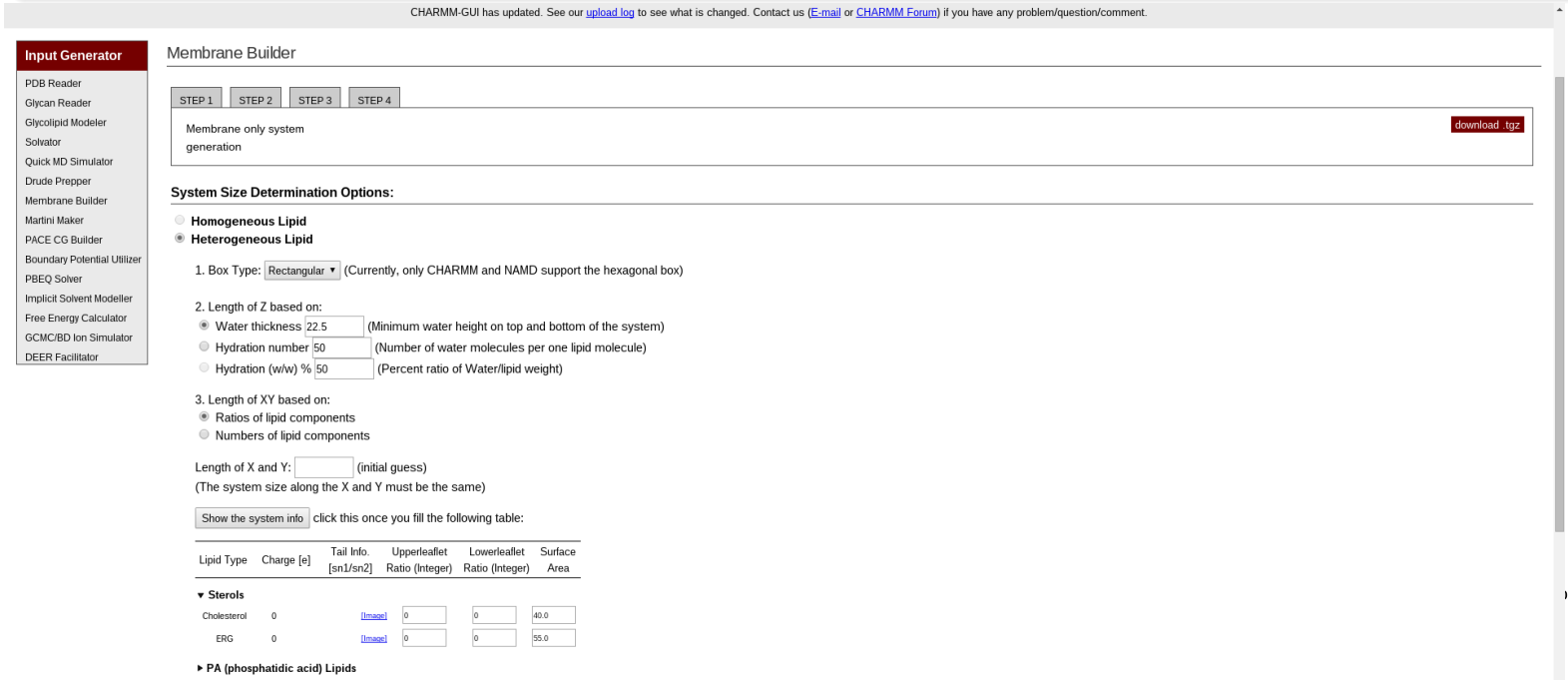
- It provides among other options the possibility to make a personalize membrane.
- Its output can be easily adapted to be used with other force fields not just charmm36.



Exercice: Simulation model plasma membrane

Composition typical plasma membrane

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



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

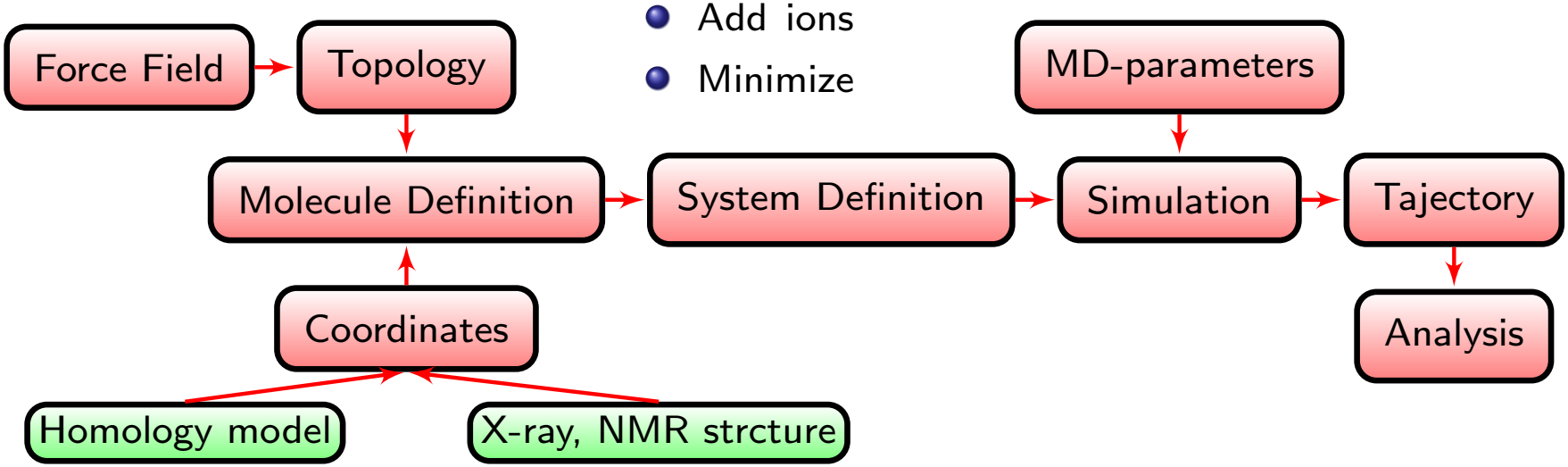
Gromacs analysis tools III

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 v5.1.1 has around 100 tools already at your service!!

System ready for simulation

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