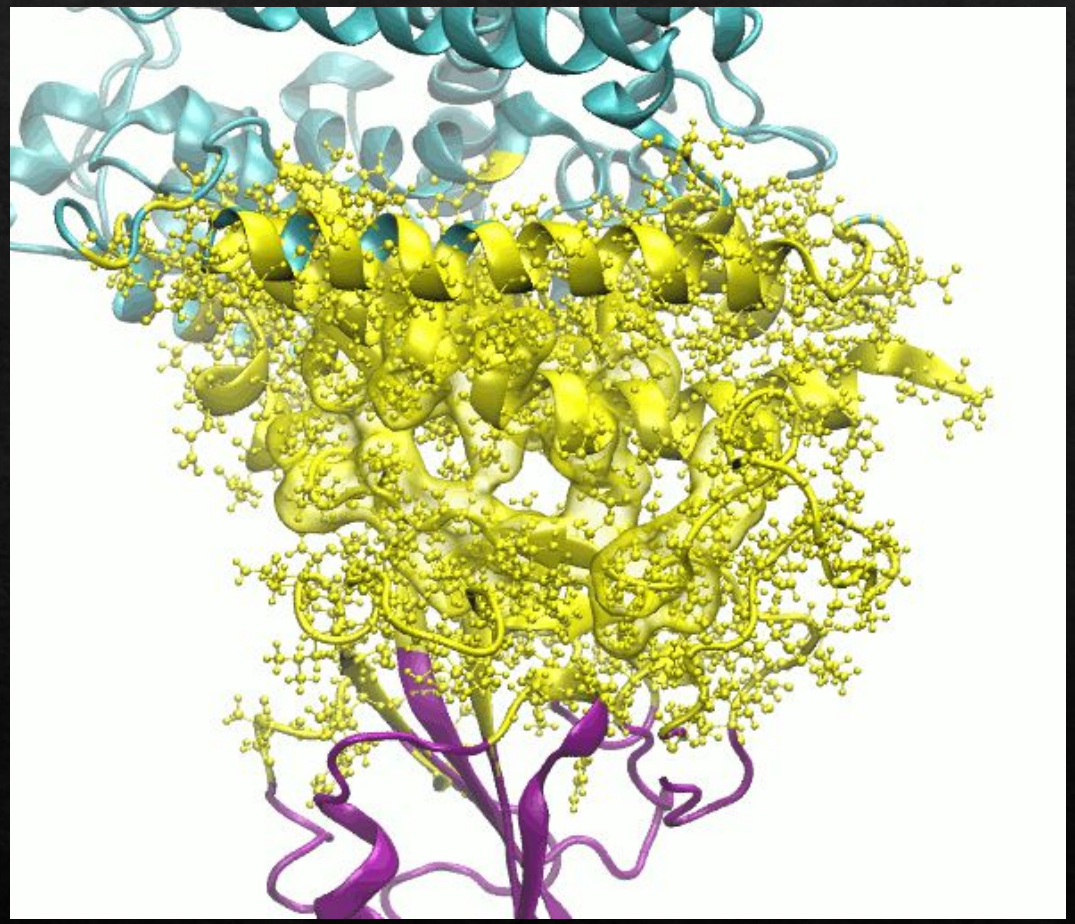


GROMACS: An Introduction

Micholas Dean Smith, PhD MRSC

Research Assistant Professor

University of Tennessee, Knoxville & UT/ORNL Center for Molecular Biophysics

What is GROMACS?

- High Performance Molecular Dynamics Engine
- Main developers located at Royal Institute of Technology and Uppsala University, Sweden
- ~2 Decades of continuous development
- Free, LGPL 2.1

GROMACS

Groningen Machine for Chemical Simulations

Image from GROMACS Reference Manual. http://manual.gromacs.org/2018-current/manual-2018.8.pdf. Accessed: 9-1-2020

GROMACS Installation

- Required Dependencies: gcc 5.1 or higher, cmake 3.9.6 or higher
- Optional Dependencies (highly recommended): fftw (http://fftw.org/)*, cuda (version >= 9.0, NVIDIA compute capacity 3.0 or better, https://developer.nvidia.com/cuda-toolkit) or OpenCL (optional, version >= 1.3, https://www.khronos.org/opencl/), MPI (must adhere to MPI standard 1.3, openMPI recommended, https://www.open-mpi.org/)
- Really Optional Dependencies: Grace (https://plasma-gate.weizmann.ac.il/Grace/) or qtGrace (https://sourceforge.net/projects/qtgrace/); zlib; BLAS/LAPACK

GROMACS Installation: A typical build

- From GROMACS 5.0 onward the build production was moved to cmake. General build instructions:

  wget http://ftp.gromacs.org/pub/gromacs/gromacs-2020.3.tar.gz Grab the code

  tar –xvf gromacs-2020.3.tar.gz

  cd gromacs-2020.3; mkdir build; cd build; ccmake ../ Personal recommendation: use the cmake interactive gui mode the first time you build so you can see all of options and check that cmake is finding the right libraries

  make –j 8; make –j 8 install

  make check Run regression tests

GROMACS Installation: cmake flags of interest

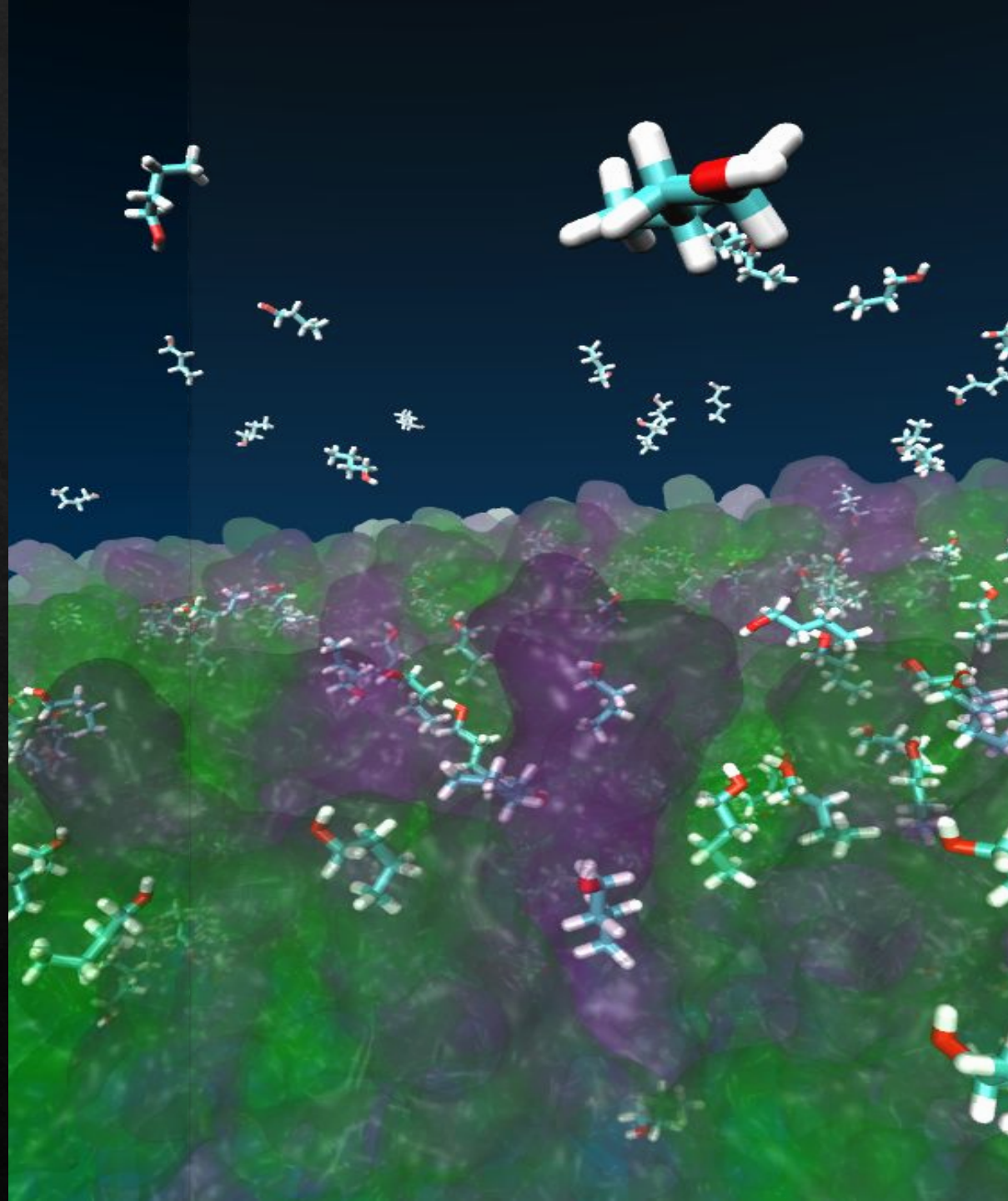
- cmake flags to be mindful of:
  - -DBUILD_TESTING needed if you want to run the reg. tests. Be sure to also set: -DREGRESSIONTEST_DOWNLOAD=ON); Alternatively, you can also get the regression test from: http://manual.gromacs.org/current/download.html
    - Detailed instructions on how to run the regression tests are located at: http://manual.gromacs.org/current/install-guide/index.html#helping-cmake-find-the-right-libraries-headers-or-programs)
  - -DGMX_GPU If using GPU acceleration be sure this is on
  - -DGMX_MPI If using MPI to allow for multi-node runs be sure to turn off –DGMX_THREAD_MPI
  - -DGMX_SMI Confirm that this matches your target architecture to optimize performance
  - -DGMX_BUILD_OWN_FFTW=ON Instructs GROMACS to download, configure, and install its own FFT3W with appropriate precision. When using a local FFT library (MKL or FFT3W) be sure to check local precision matches target precision of your GROMACS build
  - -DGMX_DOUBLE For MD simulations with gromacs double precision is not typically used.

GROMACS Installation: points of caution and tips

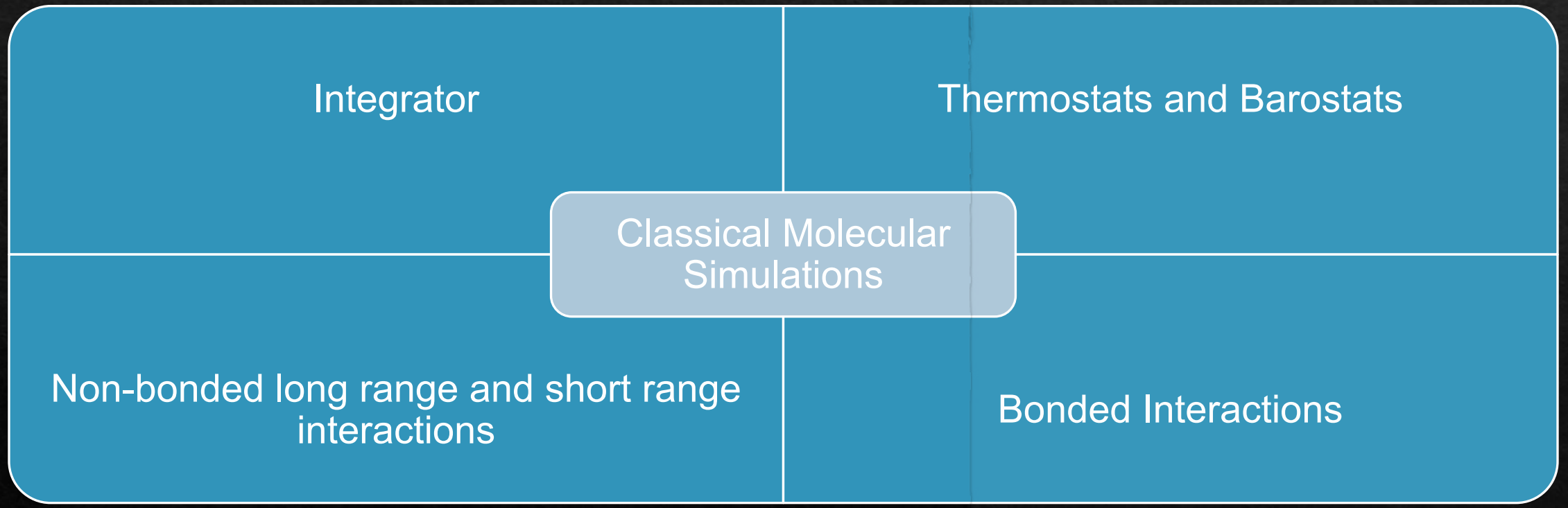
- cmake is great….when it wants to be
  - Doesn’t always play nice with module-based environment management
  - Make sure you use a clean cache when doing a fresh build
- Gromacs specific advice:
  - Stick to gnu compilers
  - CUDA > OpenCL

So now what?

- Classical (Newtonian) mechanics treatment of small molecules
- Permits the study of molecular systems composed of hundreds to millions of atoms
- Sample model systems include: biocomposites, membranes, proteins, and liquids

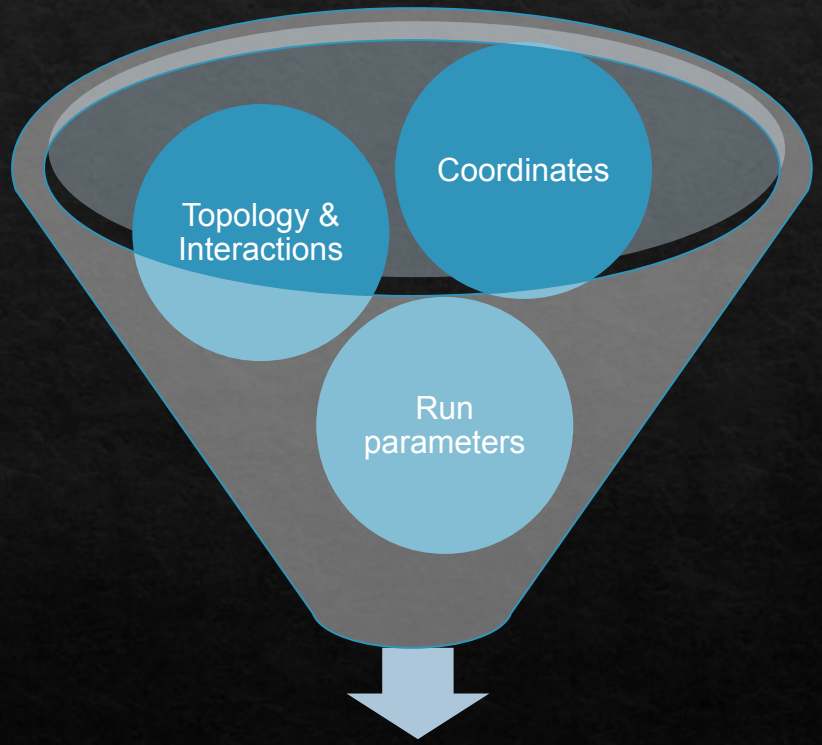


Classical Molecular Dynamics Simulations (MD)

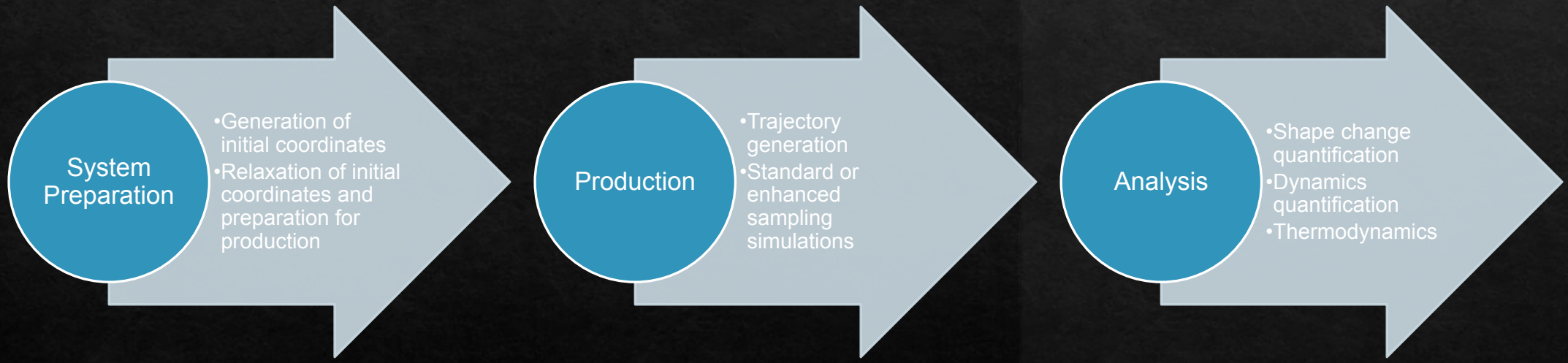


Key requirements to MD simulations with GROMACS

- GROMACS requires three different input files:
  - 1) Run-parameters
    - Gromacs *mdp files
  - 2) System (atomic) Coordinates
    - *.pdb or *gro formatted plain-text files
  - 3) System Topology & Interaction Potentials
    - Gromacs specific *.top and/or *.itp files



The GROMACS MD workflow



Run parameter files

- Typical MD simulations follow three phases
  - Energy minimization
  - Relaxation
  - Production
- Production Special Cases:
  - Umbrella sampling
  - (Temperature) Replica Exchange

```
integrator              = steep
emtol                   = 1000.0
nsteps                  = 5000
nstlist                 = 10
cutoff-scheme           = Verlet
rlist                   = 1.2
vdwtype                 = Cut-off
vdw-modifier            = Force-switch
rvdw_switch             = 1.0
rvdw                    = 1.2
coulombtype             = pme
rcoulomb                = 1.2
;
constraints             = h-bonds
constraint_algorithm    = LINCS
```

Standard run parameter file

Output and integration options

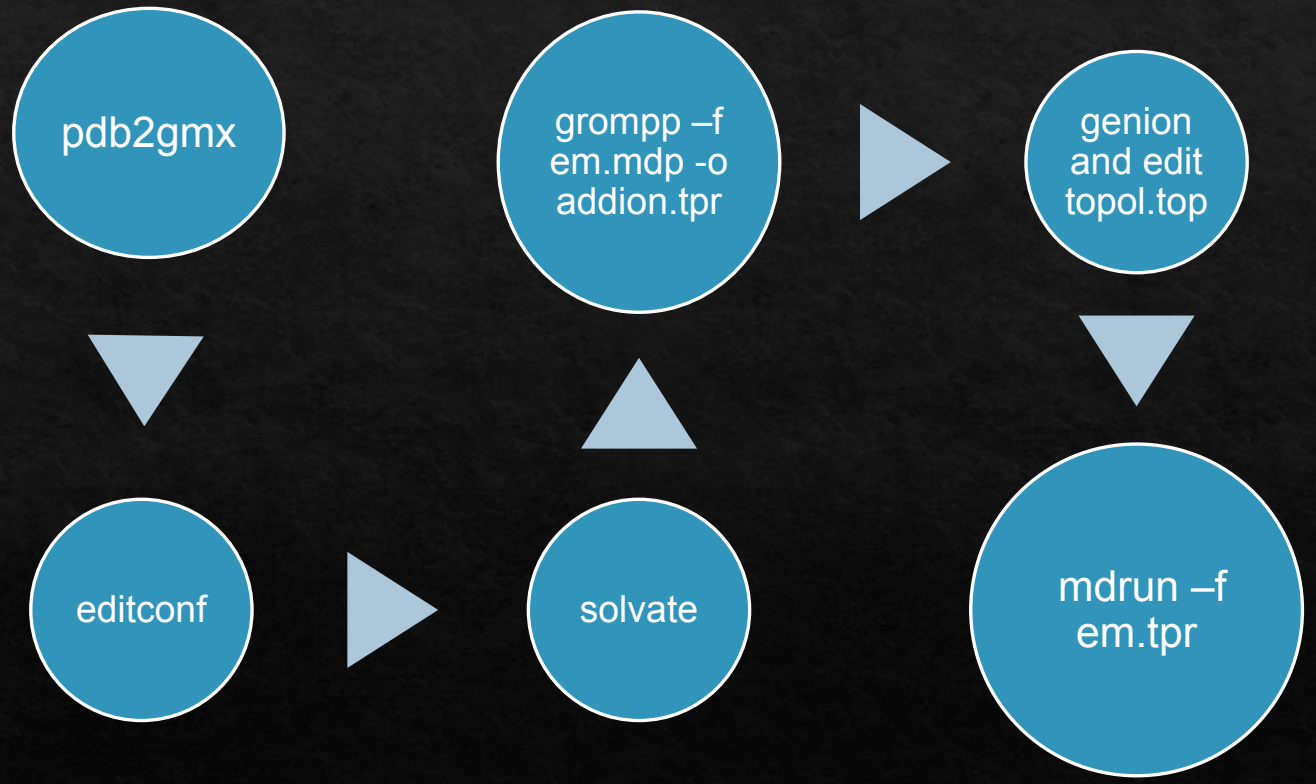
Thermostat/barostat options

Force-field specific cut-offs options

Initialization of velocities

```
define                  = -DPOSRES
integrator              = md
dt                      = 0.001
nsteps                  = 125000
nstxtcout               = 5000
nstvout                 = 5000
nstfout                 = 5000
nstcalcenergy           = 100
nstenergy               = 1000
nstlog                  = 1000
cutoff-scheme           = Verlet
nstlist                 = 20
rlist                   = 1.2
coulombtype             = pme
rcoulomb                = 1.2
vdwtype                 = Cut-off
vdw-modifier            = Force-switch
rvdw_switch             = 1.0
rvdw                    = 1.2
tcoupl                  = Nose-Hoover
tc_grps                 = SYSTEM
tau_t                   = 1.0
ref_t                   = 303.15
constraints             = h-bonds
constraint_algorithm    = LINCS
nstcomm                 = 100
comm_mode               = linear
comm_grps               = SYSTEM
gen_vel                 = yes
gen_temp                = 303.15
gen_seed                = -1
refcoord_scaling        = com
```

Formatting Coordinates and Obtaining Topologies



Great, so now how do I run a simulation?

- Generic simulation:
  - gmx_mpi mdrun (or gmx mdrun) –s production.tpr –c production_out.gro –e production.edr –v –cpo state_production.cpt –cpi state_production.cpt –cpt 1 –o production.trr –g production.log
- Multi-copy (same production different random seeds, atomic positions, or different temperatures):
  - for((i=0;i<NumberOfCopies;i++));do mkdir window_0”$i” ; cp production_”$i”.tpr window_0”$i”/production.tpr;done
  - gmx_mpi (or gmx mdrun) –s production.tpr –c production_out.gro –e production.edr –g production.log –cpo state_production.cpt –cpi state_production.cpt –cpt 1 –o production.trr –g production.log –muldir `ls –d window_0*`

Great, so now how do I run a simulation?

- Generic simulation:
  - gmx_mpi mdrun (or gmx mdrun) –s production.tpr –c production_out.gro –e production.edr –v –cpo state_production.cpt –cpi state_production.cpt –cpt 1 –o production.trr –g production.log
- Multi-copy (same production different random seeds, atomic positions, or different temperatures):
  - for((i=0;i<NumberOfCopies;i++));do mkdir window_0”$i” ; cp production_”$i”.tpr window_0”$i”/production.tpr;done
  - gmx_mpi (or gmx mdrun) –s production.tpr –c production_out.gro –e production.edr –g production.log –cpo state_production.cpt –cpi state_production.cpt –cpt 1 –o production.trr –g production.log –muldir `ls –d window_0*`

Great, so now how do I run a simulation?

- So what do those flags mean?
  - -s system input file (gromacs binary topology file tpr format)
  - -o trajectory file (gromacs trr format)
  - -cpo , -cpi, -cpt (-cpo/-cpi -> checkpoint file output/input, cpt binary files -> checkpoint save frequency)
  - -g logfile (plain text necessary for some forms of post-processing T-REMD)
  - -e energy file (records system energy terms and thermostat/barostat behavior, binary gromacs edr format)
  - -multidir list of directories with input tpr and output files
  - -replex special flag for T-REMD simulations, sets step interval for exchange trials

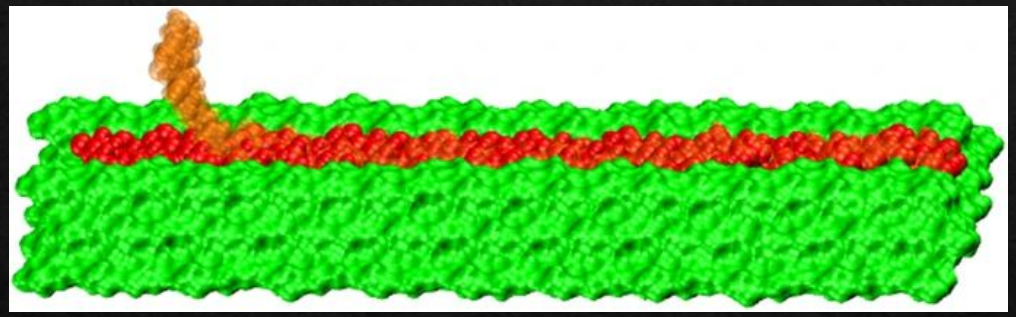
Performance related flags

- gmx mdrun has a number of performance related options
  - -nb non-bonded force calculation on cpu, gpu, auto
  - -pme electrostatics particle mesh ewald, gpu or cpu, or auto
  - -pmefft perform FFT on gpu or cpu (or auto assign) for PME
  - -bonded perform bonded force calculations on gpu, cpu, or auto assign
  - -update perform integration (update) on gpu or cpu
  - -ntomp number of openMP threads per MPI rank
  - -ntomp_pme number of openMP threads per MPI rank dedicated to PME
  - -pin, pinoffset, pinstride let gmx mdrun set thread affinities *Should be used with caution
  - -dlb dynamics load balance for domain decomposition
  - -dd manually set domain decomposition
  - -ddorder rank-order of domain decomposition
  - -gpu-id list of unique gpu ids
  - -gputasks manual mapping of gpu device IDs to PP task on each node

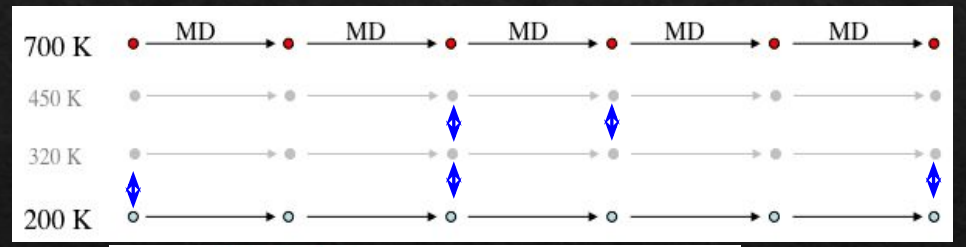
Things to keep in mind for performance

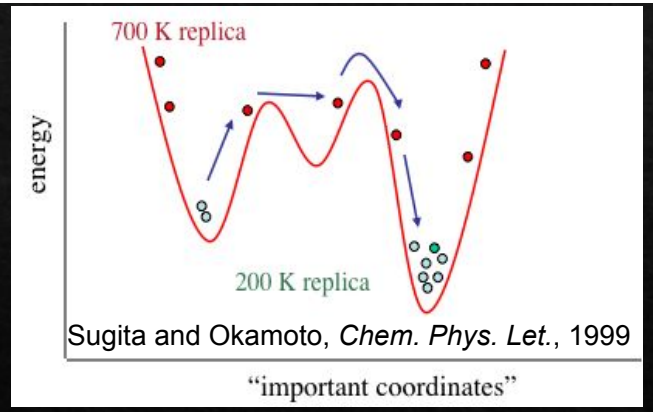
- GPU off-loading is a good thing
- gmx tune_pme can be helpful for quick benchmarking and run tuning:
  - http://manual.gromacs.org/current/onlinehelp/gmx-tune_pme.html
- “Write” just often enough
  - Unless you are tasked with looking at very fast relaxations: don’t write every integration step, every 5 to 10ps (~2500 to 5000 integration timestep of 2fs) is normally sufficient for most structural analysis of proteins.
  - Appending continuations to trajectories vs separate trajectories?
- Multiple MD simulations are typically needed
  - Many single node jobs vs a few multi-node jobs? (Which gives the best bang-for-your-buck)

MD Simulation: Two Special Cases



Pulling with an external force





Sugita and Okamoto, *Chem. Phys. Let.*, 1999

Partly adopted from Dr. Ronald M. Levy

Temperature Replica-Exchange

MD Special Case: Pulling

- Simulations where an external force is applied to a specified set of atoms
- Goal: obtain estimate of binding free-energy (useful for drug discovery, alternative methods also exist)
- Additional terms in mdp (run parameter files) (see: http://manual.gromacs.org/2019/user-guide/mdp-options.html#mdp-pull-ngroups for complete details).
  - pull=yes
  - pull-nstxout
  - pull-nstfout
  - pull-ngroups
  - pull-coord1-type
  - pull-coord1-dim
  - pull-coord1-vec
  - Pull-coord1-k

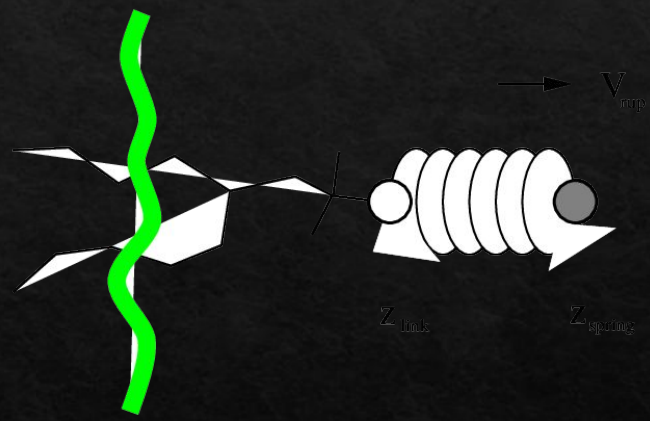
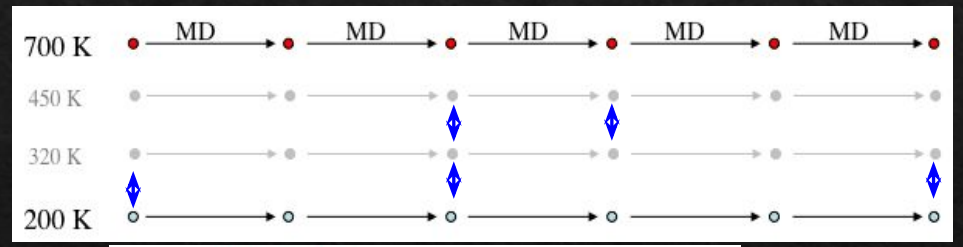
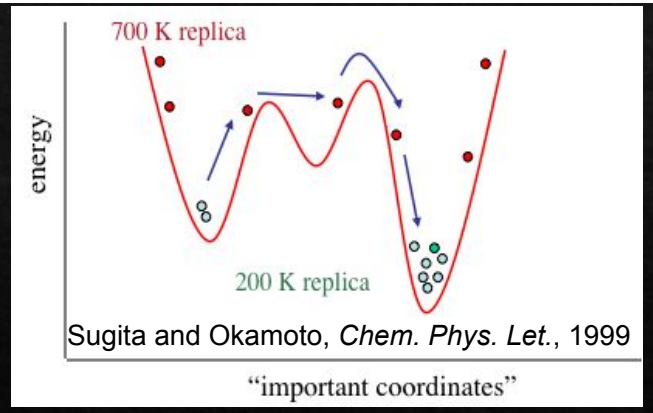


Image from GROMACS Reference Manual. http://manual.gromacs.org/2018-current/manual-2018.8.pdf. Accessed: 9-1-2020

MD Special Case: T-REMD

- Coupled multi-temperature simulations
- Goal: enhance sampling (potentially useful for drug discovery)
- Demands: many multiple simulations
- No additional terms in mdp (run parameter files)
- Requires careful choice of temperature intervals for proper exchanges





Sugita and Okamoto, *Chem. Phys. Let.*, 1999

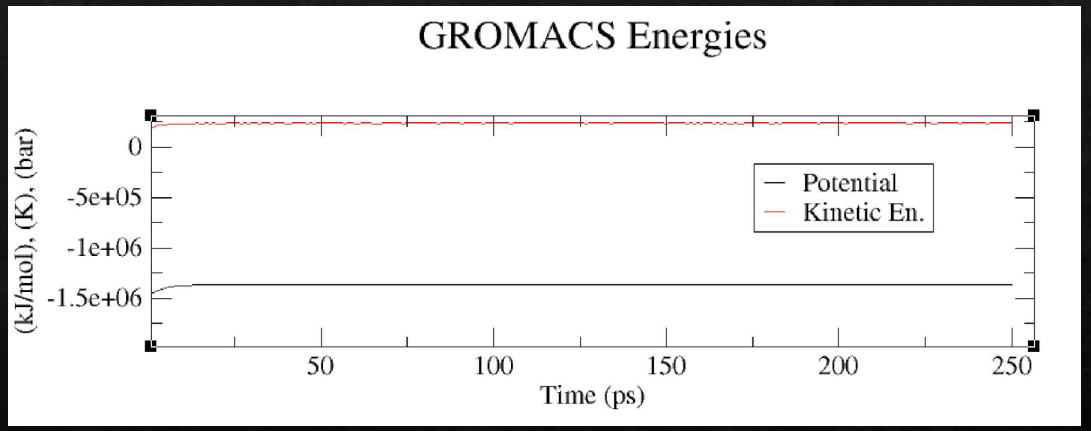
Partly adopted from Dr. Ronald M. Levy

Analysis: Getting something out of MD

- The point of MD is to “measure” something.
  - Diversity of pocket sizes?
  - Hydrogen bonding?
  - How well two different polymers bind to one-another
- In the end: simulations are only 70% of the work
- Main problem: analysis code is generally not optimized.

Analysis: Getting something out of MD

- Main steps:
  - Check system for simulation stability
    - gmx energy
  - Generation of Index files
    - gmx make_ndx or gmx select
  - Calculate molecular features:
    - gmx hbonds
    - gmx sas
    - gmx gyrate



Troubleshooting Crashes

- Two types of crashes:
  - Unstable MD system
    - Check to see if system was properly minimized
    - Confirm thermostat/barostat are not suffering large fluctuations
    - Reduce integration timestep
  - Hardware failure => File corruption
    - Restart with previous checkpoint (default name: state_prev.cpt)
    - gmx check –f (your trajectory here).trr scans trajectory for length and notes last working frame
- Checkpoint often!
- Consider including –noappend with gmx mdrun

Where to go for more help

- GROMACS How-To Guides:
  - http://manual.gromacs.org/current/how-to/index.html
- GROMACS User Manual:
  - http://manual.gromacs.org/documentation/current/user-guide/index.html#user-guide
- GROMACS Forum:
  - https://gromacs.bioexcel.eu/
- Developer’s Guide (for those that want to tinker)
  - http://manual.gromacs.org/current/dev-manual/index.html

Q & A