

AODI Molecular dynamics workshop

Ross Amory

School of Chemistry
University of Nottingham

Table of Contents

Introduction
Me

Theory
Docking
Molecular Dynamics
Free Energy
Hardware
Software

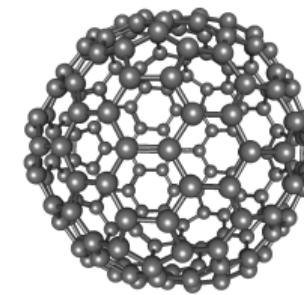
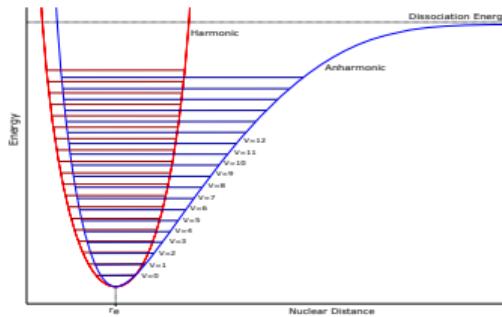
Workshop
Background
Tasks
Structures
Docking
Parameterisation
Molecular Dynamics
Analysis

Who am I?

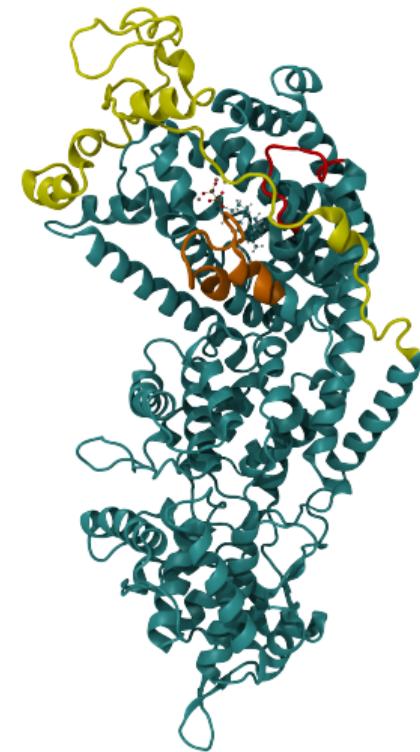
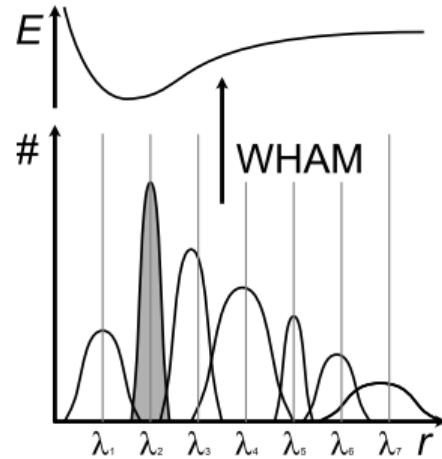
- Ross Amory
- Integrated Masters in Chemistry at the University of Nottingham
- Masters with Prof. Nicholas Besley
- PhD Student at the University of Nottingham
- Supervisors are
 - ▶ Prof. Chris Hayes,
 - ▶ Prof. Jonathan Hirst,
 - ▶ Dr. Christof Jager
 - ▶ Dr. David Rogers



Masters Project:



- Calculate an anharmonic correction for Raman Sepctroscopy
- Implement 2nd order numerical derivatives in Fortran90
- Calculate Raman Spectrum for Carbon Fullerenes using DFT methods
- Compare empirical code to DFT methodology

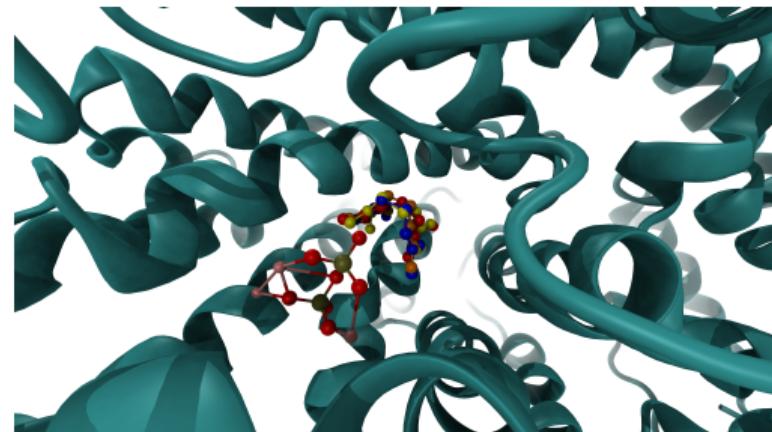


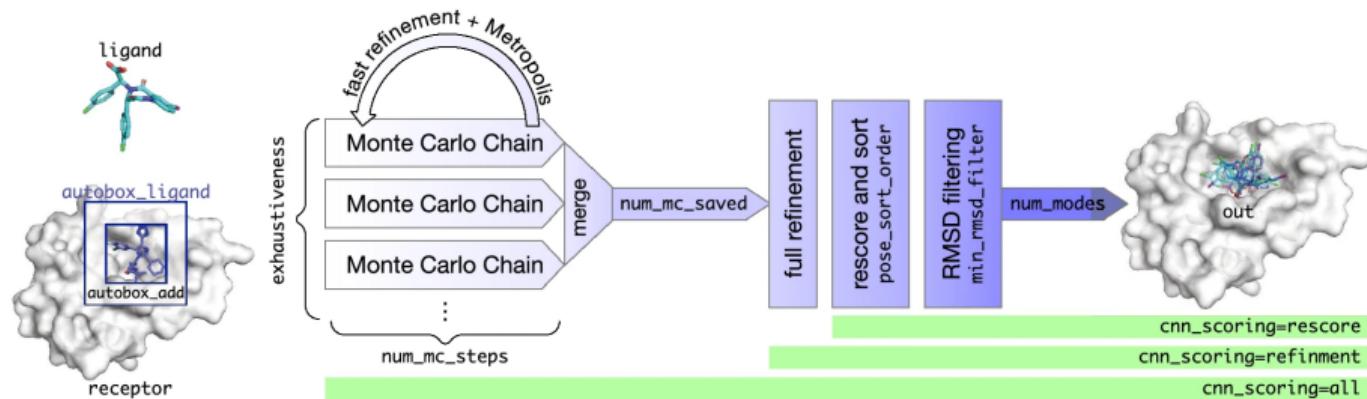
- Molecular dynamics simulations of Terpene Synthases
- Higher level QM/MM Umbrella sampling to obtain Potential Energy Surfaces

- **Self-taught** in Molecular Dynamics
- Experience of the entire workflow:
 - ▶ Homology Modelling with AlphaFold and YASARA
 - ▶ Docking using AD4 and Gnina
 - ▶ Simulations using AMBER
 - ▶ QM/MM Simulations with AMBER and QChem
 - ▶ Analysis using GBSA, Umbrella Sampling, clustering ect.



- Estimates binding affinity between a ligand and protein
- Samples translational, rotational and conformations space of the ligand
- Uses a numerical scoring function
- Highly parameterised meaning accuracy is system dependant





- Uses Monte Carlo methods to sample conformational space
- Initially scores using empirical scoring function
- Uses Convolutional Neural Networks to rescore and improve performance

McNutt, A.T., Francoeur, P., Aggarwal, R. et al. GNINA 1.0: molecular docking with deep learning. J Cheminform 13, 43 (2021)

Molecular Dynamics employs an empirical force field derived from classical physics in order to describe the system.

$$\begin{aligned}E_{Total} &= E_{Bonded} + E_{Non-Bonded} \\E_{Bonded} &= E_{Bond} + E_{Angle} + E_{Dihedral} \\E_{Non-Bonded} &= E_{Electrostatic} + E_{van\ der\ Waals}\end{aligned}\tag{1}$$

- Potential energy of the system is calculated using Equation (1).

Molecular Dynamics employs an empirical force field derived from classical physics in order to describe the system.

$$F(x) = -\Delta E(x) \tag{2}$$

1. Potential energy of the system is calculated using Equation (1).
2. Forces are derived by taking the negative gradient of the potential energy with respect to the atomic coordinates.

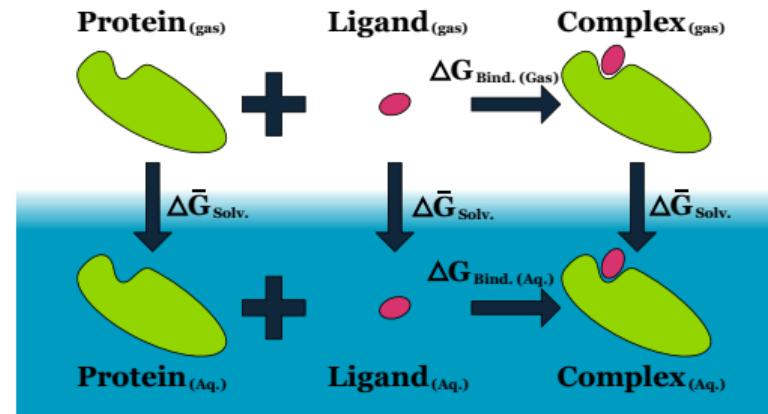
Molecular Dynamics employs an empirical force field derived from classical physics in order to describe the system.

$$\begin{aligned} F &= ma \\ x_{i+1} &= x_i + \delta_t v_i \end{aligned} \tag{3}$$

1. Potential energy of the system is calculated using Equation (1).
2. Forces are derived by taking the negative gradient of the potential energy with respect to the atomic coordinates.
3. Velocities are then calculated using Newtons Laws of motion and the positions are updated with respect to the time step.

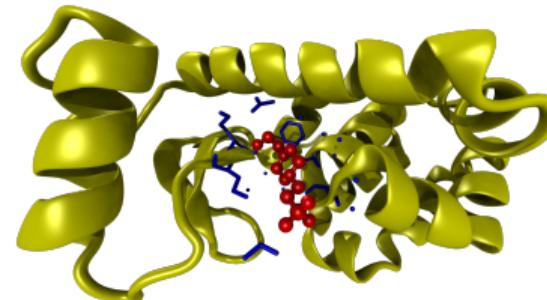
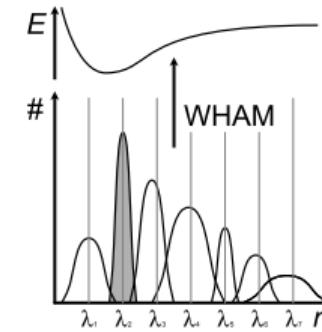
- Dynamics energies are heavily dominated by solvent-solvent interactions
- Impossible to directly extract binding energies
- Use thermodynamics cycles to approximate relative energies

- Postprocessing technique to approximate free energy
- Uses implicit solvent
- Estimates entropy



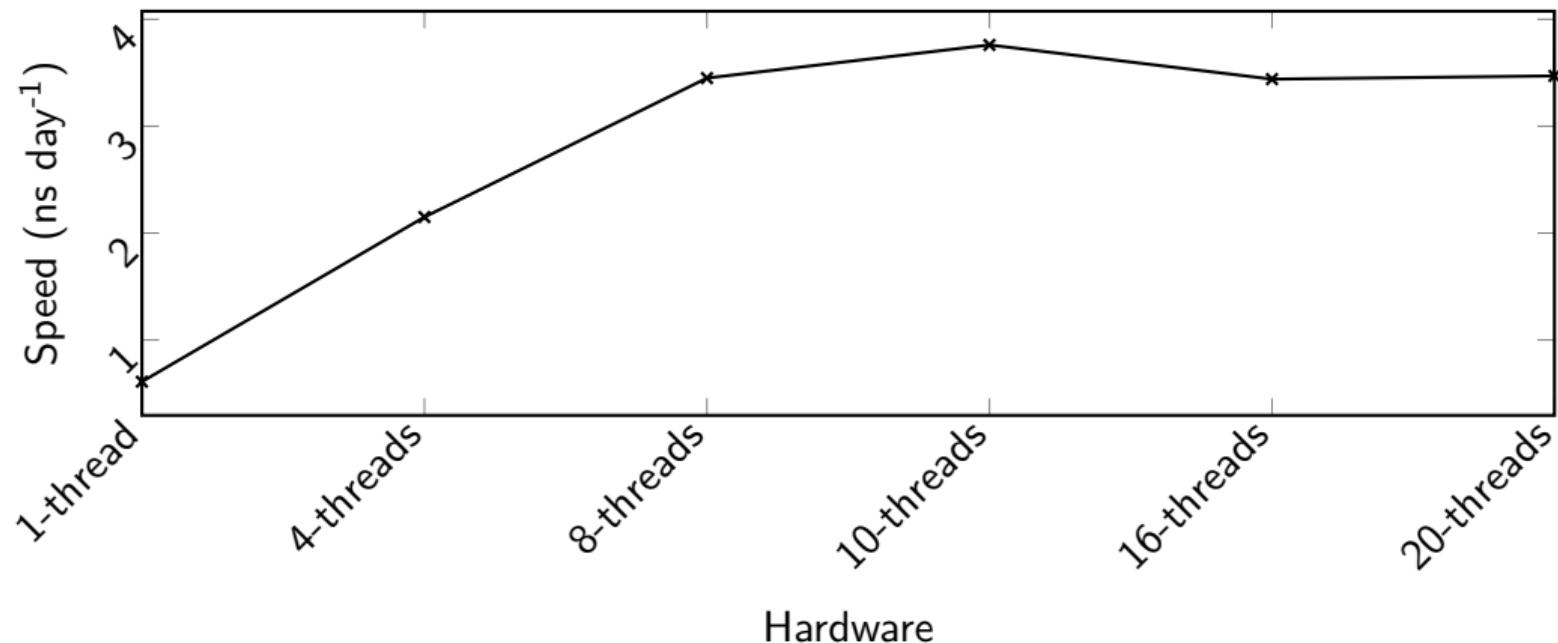
$$\Delta G_{bind} = \Delta \bar{G}_{complex} - \Delta \bar{G}_{prot.} - \Delta \bar{G}_{lig.} \quad (4)$$

- Umbrella Sampling
- Free Energy Perturbation
- QMMM hybrid methods
- Thermodynamic Integration



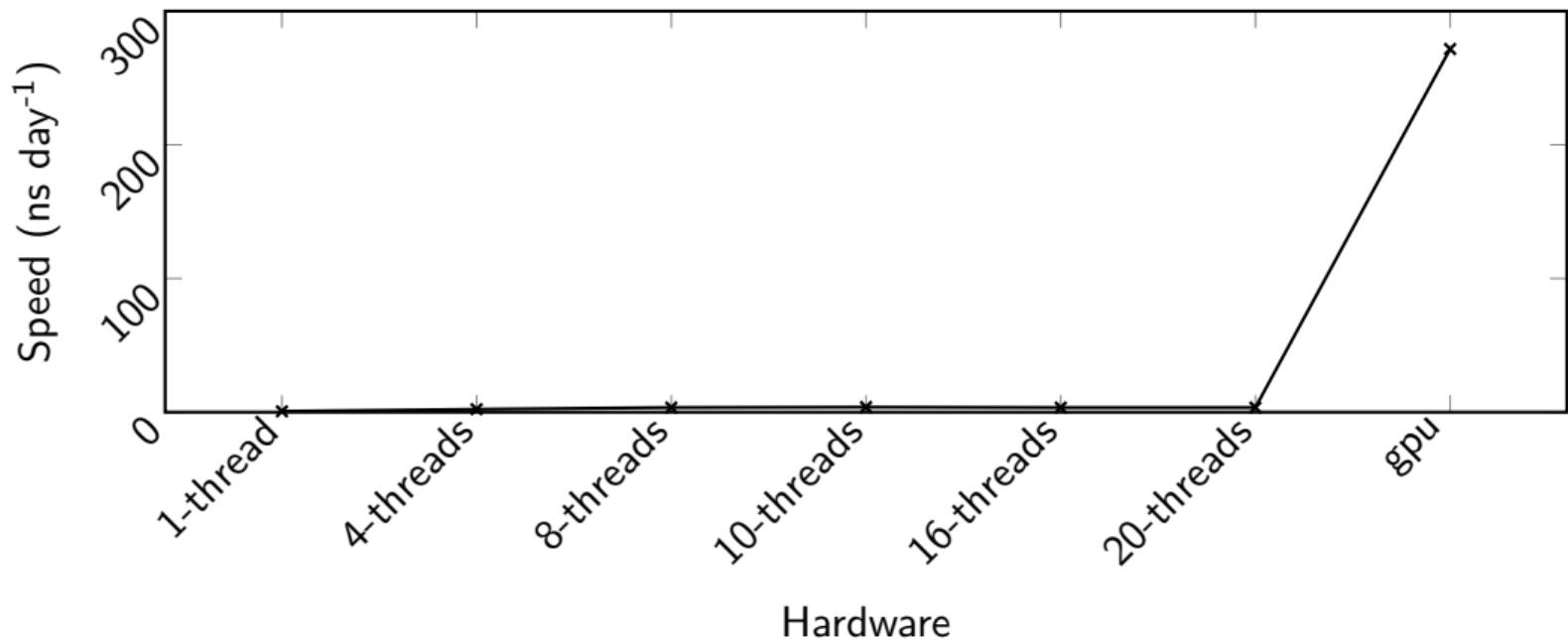
Molecular dynamics scales incredibly well with increasing computational power.

Hardware scaling of MD simulations



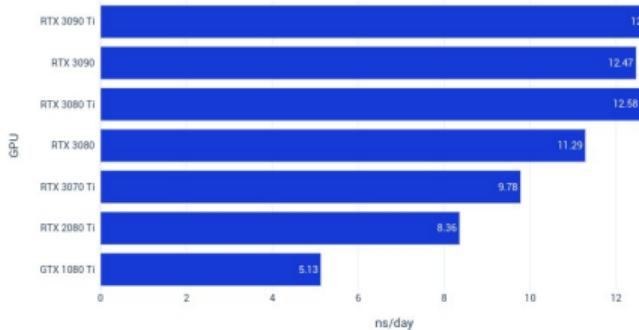
Molecular dynamics scales incredibly well with increasing computational power.

Hardware scaling of MD simulations



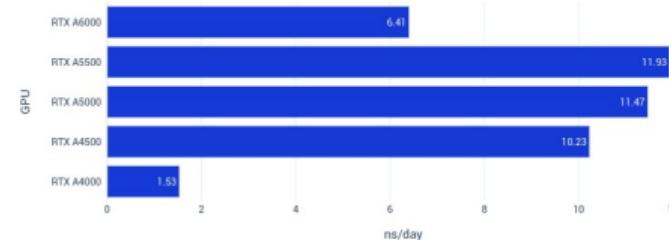
GPU Benchmark

Gromacs v2022 RIB, Ribosome in Water, 2 M atoms,
ns/day (HIGHER IS BETTER)



 Puget Systems

Gromacs v2022 RIB, Ribosome in Water, 2 M atoms,
ns/day (HIGHER IS BETTER)



 Puget Systems

Benchmark data: Puget Systems

Section	Software	Availability
Docking	gnina	Open Source
Preparation	acPype	Free
Dynamics	Amber (AmberTools)	Free*
Analysis	AmberTools	Free
Visualisation	VMD	Free for academic use

Table: Software

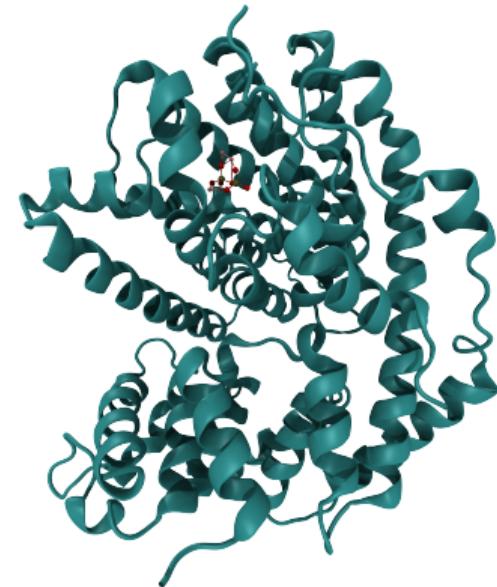
* AmberTools is free to use, but full Amber requires a licence. This is only used for GPU acceleration.

The system

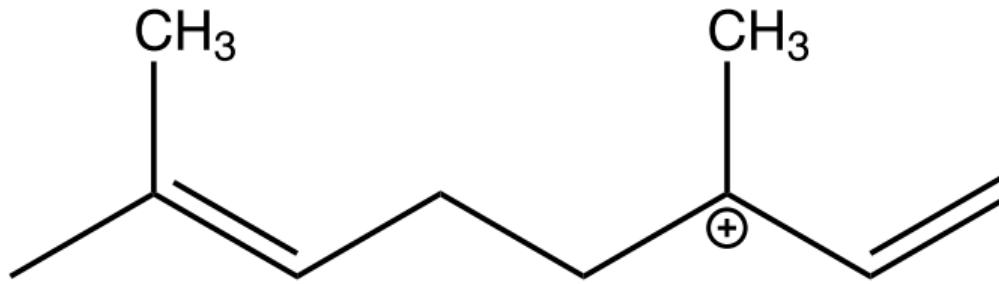
Terpene Synthase “(+)-Bornyl Diphosphate Synthase”

- 4511 Atoms (Not including solvent or ligand)
- Dynamical system that exists at room temp.
- Protein is not in isolation (Needs solvent)
- Reactions can require nanosecond simulation lengths

Therefore DFT or ab-initio methods are too expensive!
Molecular dynamics is therefore a much more sensible option.



The ligand



- Initial intermediate Geranyl Diphosphate - Bornyl Diphosphate synthetic route
- Monoterpene carbocation
- Natural ligand for this protein
- Unsaturated

Workshop overview

The structure of the workshop is as follows:

- Download starting structures
- Dock a ligand
- Parameterise a ligand
- Setup a dynamics simulation
- Run a dynamics simulation
- Perform analysis on a simulation
- Approximate the free energy



This workshop is available as a PDF on GitHub:

https://github.com/pcyra2/MD_workshop/blob/main/main.pdf

The slides are also available on GitHub:

https://github.com/pcyra2/MD_workshop/blob/main/Slides/main.pdf

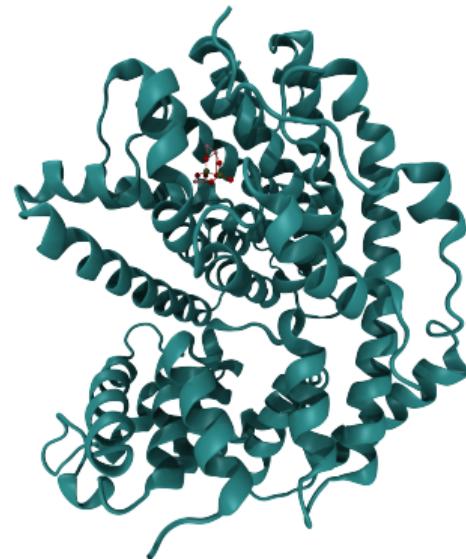
Obtaining the Protein

There are many repositories for protein crystal structures:

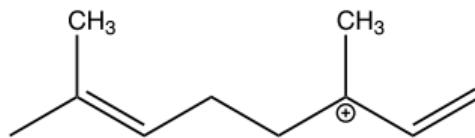
- <https://www.rcsb.org/>
- <https://www.uniprot.org/>
- <https://www.wwpdb.org/>

Our protein has the PDB code 1N23:

<https://www.rcsb.org/structure/1N23>



Obtaining the Ligand

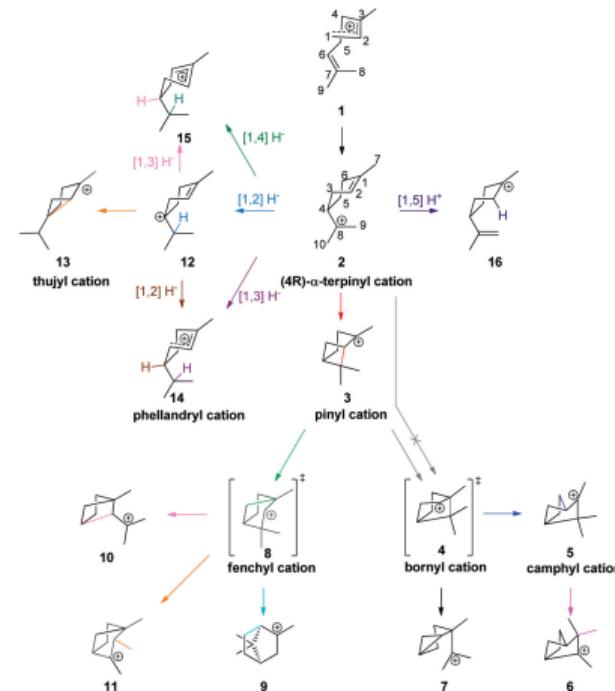


- Structure 1 from the reaction pathway
- Other intermediates from this reaction scheme should also work

M. Weitman, D. Major, Challenges Posed to Bornyl Diphosphate Synthase:

Diverging Reaction Mechanisms in Monoterpene, J. Am. Chem. Soc.,

2010, 132 (18), 6349-6360





```
"gnina -r 1n23_monomer_NoLig.pdb -l LIG.mol2 --autobox_ligand 2BN.pdb -o docked.mol2 --no_gpu"
```

- Use ligand bound in crystal structure to dock the ligand to the active site
- “--no_gpu” only required if no GPU is present

Using “AnteChamber PYthon Parser interfacE” to parameterise a non-standard ligand.

- Ligand is flexible
- Ligand is charged
- Ligand has a mix of atom types

acPype tutorial: <https://alanwilter.github.io/acpype/>

Further information: <https://ambermd.org/tutorials/basic/tutorial4b/index.php>

The standard steps of a dynamics simulation are as follows:

1. Minimisation
 - ▶ 0 K
2. Heat
 - ▶ 0 K to 300 K, Constant volume
3. Equilibration
 - ▶ 300 K, Constant volume and pressure
4. Production dynamics
 - ▶ 300 K, Constant volume and pressure

Molecular dynamics simulations require an initial parameter and structure file to run.

We need to:

1. Create separate files for the Mg²⁺ and Diphosphate ions
2. Load parameters for the protein, ligands and solvent
3. Join these files, plus the docked ligand
4. Solvate the system
5. Neutralise the system
6. Generate parameter and coordinate files

Box shape

- Solvent molecules dominate simulation time
- Reducing amount of solvent speeds up simulations
- Want the periodic box to be optimised to reduce solvent molecules
- Truncated octahedron is generally a good shape for “round” proteins

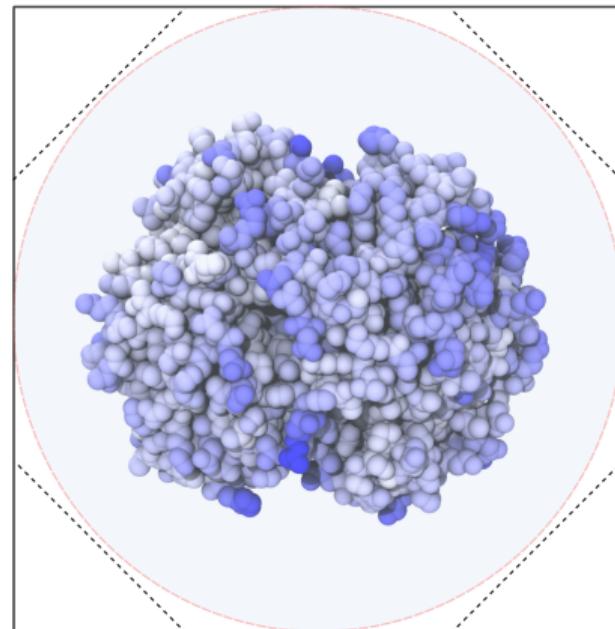
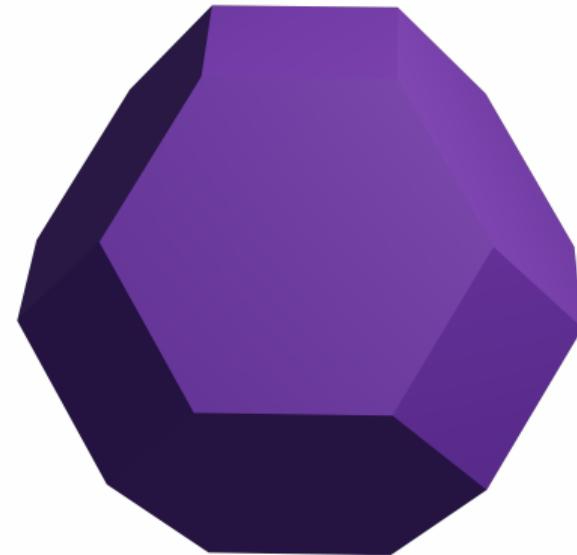


Figure: https://computecanada.github.io/molmodsim-md-theory-lesson-novice/04-Periodic_Boundary/index.html

Box shape

- Solvent molecules dominate simulation time
- Reducing amount of solvent speeds up simulations
- Want the periodic box to be optimised to reduce solvent molecules
- Truncated octahedron is generally a good shape for “round” proteins





sander file

```
initial minimisation script at 0k
&cntrl
 imin=1,          # initiate minimisation
  ntx=1,          # read coordinates with no velocities
  irest=0,         # don't restart the simulation
  maxcyc=10000,   # 10000 minimisation steps
  ncyc=5000,       # 5000 steps of steepest gradient
  ntpr=10,         # print to mdout every 10 steps
  ntwx=0,          # don't print trajectory file
  cut=8.0,         # non-bonded cut off of 8 A
/
```



sander file

```
heating script from 0 k to 300 k over 200 ps
&cntrl
  imin=0,          # run a dynamics simulation
  ntx=1,          # read coordinates with no velocities
  irest=0,         # don't restart the simulation
  nstlim=100000,   # run simulation for 100000 steps
  dt=0.002,        # each step is separated by 0.002 ps (200 ps total)
  ntf=2, ntc=2,    # constrain bonds with hydrogen
  ntpc=100,        # print to mdout every 100 steps
  ntwx=100,        # print trajectory file every 100 steps
  cut=8.0,         # non-bonded cut off of 8 A
  ntb=1,           # constant volume with periodic boundary conditions
  ntp=0,           # no pressure control
  ntt=3,           # control temperature using Langevin Dynamics
  gamma_ln=2.0,    # Langevin collision frequency
  nmropt=1,        # control NMR restraints
  ig=-1,           # use a random seed
/
&wt type='TEMP0', istep1=0, istep2=100000, value1=0.0, value2=300.0 /
&wt type='END' /
# Heat the system from 0 to 300 k from step 0 to end
```



sander file

```
equilibration script at 300k for 200 ps
&cntrl
  imin=0,          # run a dynamics simulation
  ntx=5,          # read coordinates with no velocities
  irest=1,         # don't restart the simulation
  nstlim=100000,   # run simulation for 100000 steps
  dt=0.002,        # each step is separated by 0.002 ps (200 ps total)
  ntf=2, ntc=2,    # constrain bonds with hydrogen
  ntp=100,         # print to mdout every 100 steps
  ntwx=100,        # print trajectory file every 100 steps
  cut=8.0,         # non-bonded cut off of 8 A
  ntb=2,           # constant volume with periodic boundary conditions
  ntp=1,           # pressure control
  ntt=3,           # control temperature using Langevin Dynamics
  barostat=1,       # Berendsen barostat for pressure control
  gamma_ln=2.0,     # Langevin collision frequency
  ig=-1,            # use a random seed
  temp0=300.0,      # start at 300 k
  tempi=300.0       # end at 300 k (Constant temp)
/
```

Run Simulations

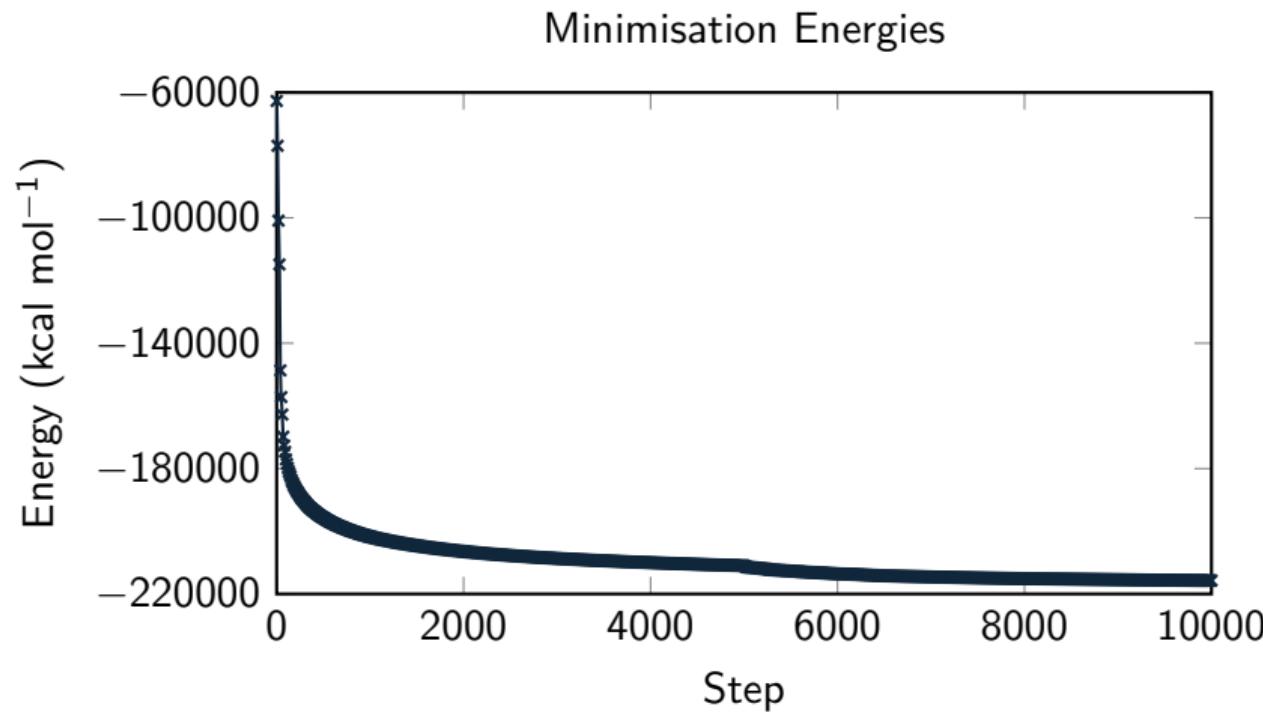
bash command

```
$ # If using the conda install or have no mpi compiled version:  
$ sander -O -i min.in -o min.out -c start.rst7 -p complex.parm7  
    ↵ -inf min.mdinfo -r min.ncrst  
  
$  
$ # If compiled with mpi:  
$ mpirun -np NUM_CORES sander.MPI -O -i min.in -o min.out -c  
    ↵ start.rst7 -p complex.parm7 -inf min.mdinfo -r min.ncrst  
  
$  
$ # If using the full version of amber with GPU support:  
$ pmemd.cuda -O -i min.in -o min.out -c start.rst7 -p complex.parm7  
    ↵ -inf min.mdinfo -r min.ncrst  
$ ### NOTE: Often the pmemd version struggles if large clashes are  
    ↵ present. If you get an error, try switching to sander.
```

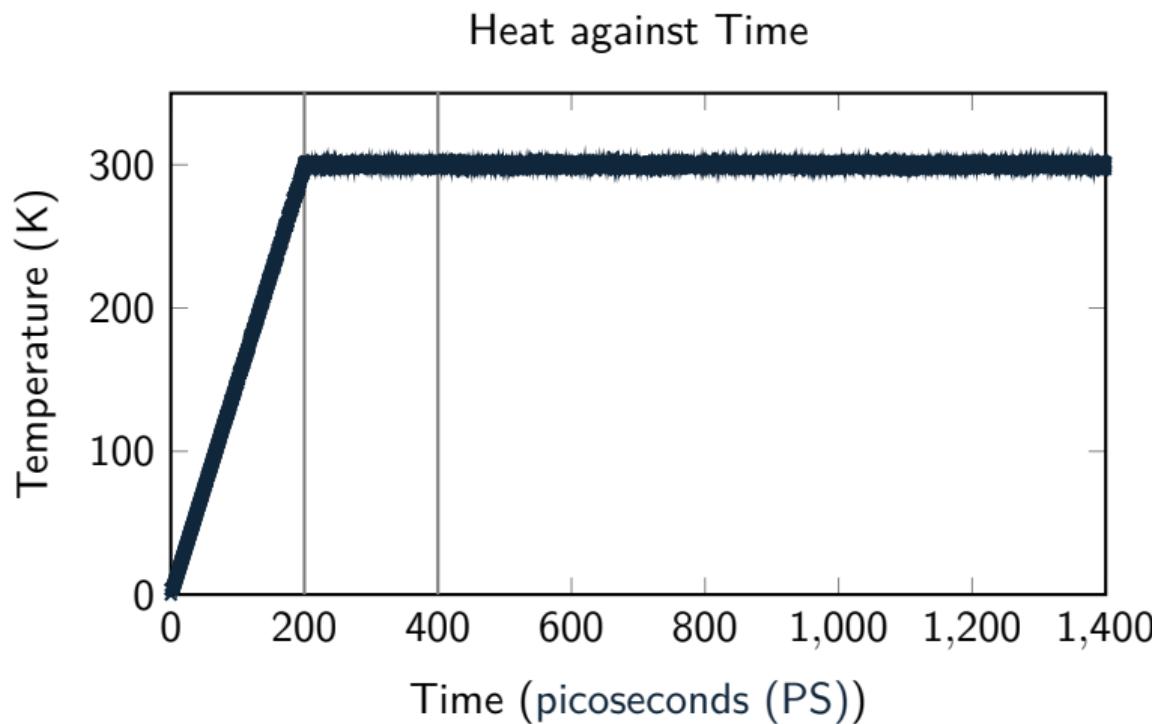
We need to perform analysis to ensure our simulations have run correctly.

You should always check:

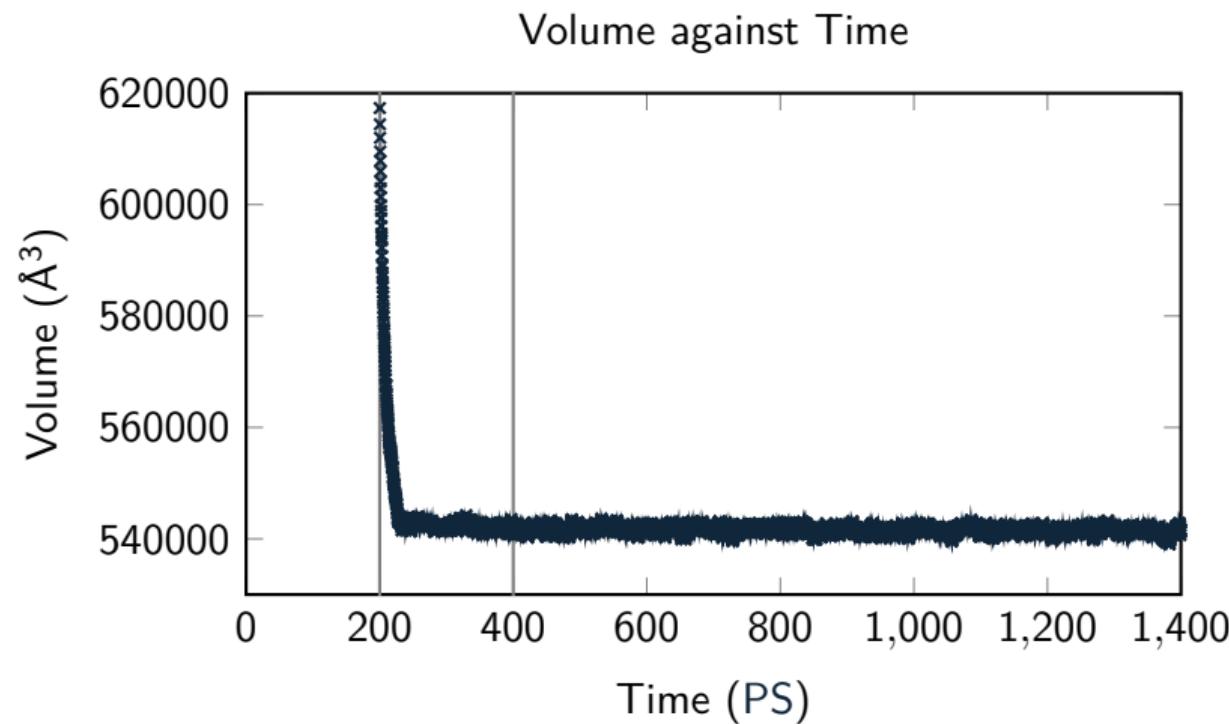
- Has the energy converged during minimisation?
- Is the temperature correct?
- Is the volume constant during NVT ensembles?
- Is the energy stable during simulations?
- Is the protein stable during simulations?

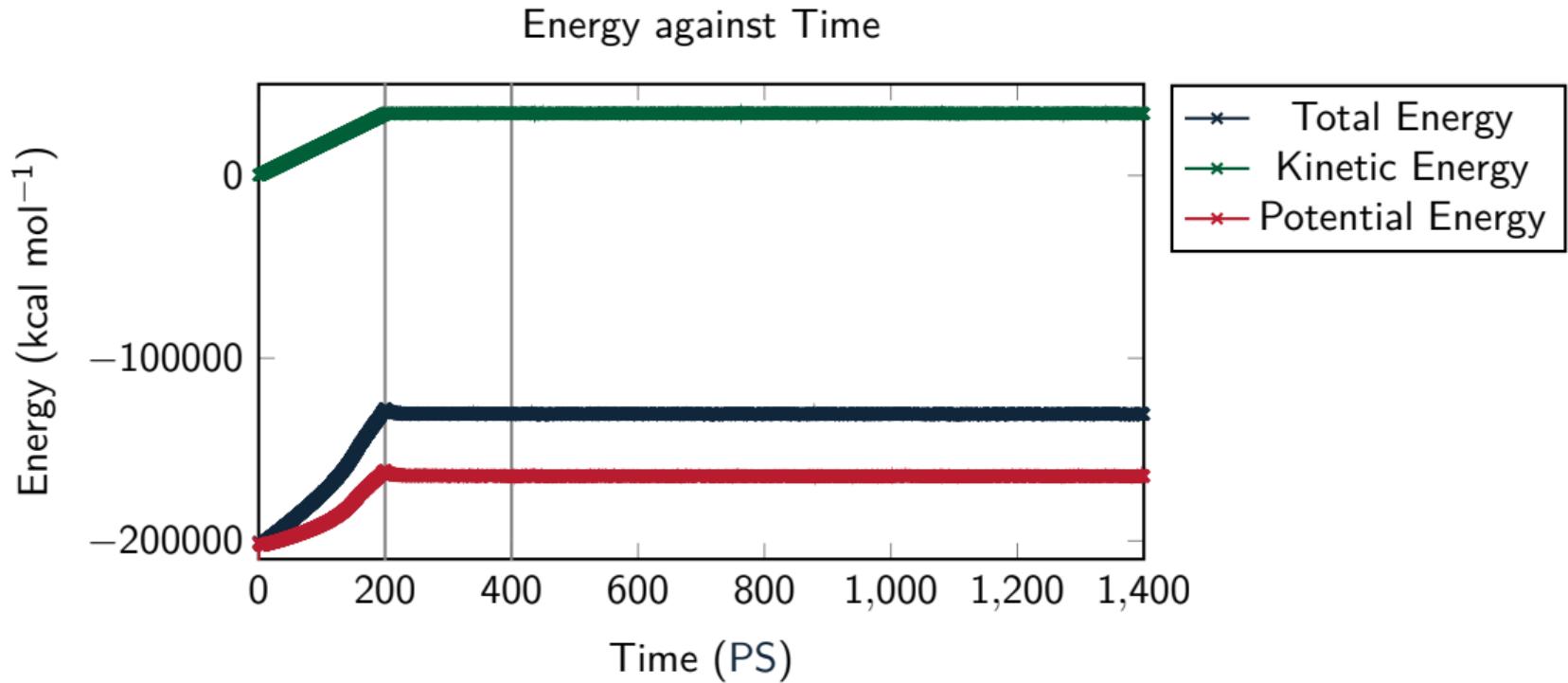


Temperature Check



Volume Check





Energy Check

