

# AODI Molecular Dynamics workshop

Ross Amory

School of Chemistry  
University of Nottingham

# 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](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](https://github.com/pcyra2/MD_workshop/blob/main/Slides/main.pdf)

# Table of Contents

Tasks	Docking
Introduction	Molecular Dynamics
Me	Parameterisation
Workshop	Molecular Dynamics
Background	Analysis
Tasks	Free Energy
Structures	Free Energy
Docking	Conclusion

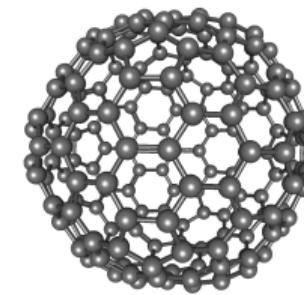
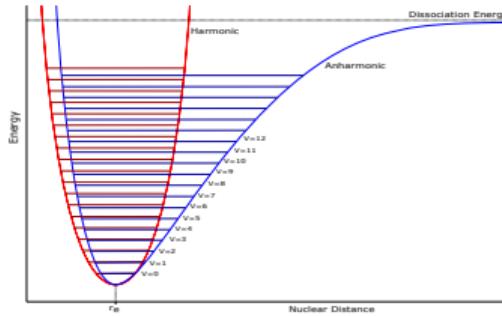
# Introduction - Me

Who am I?

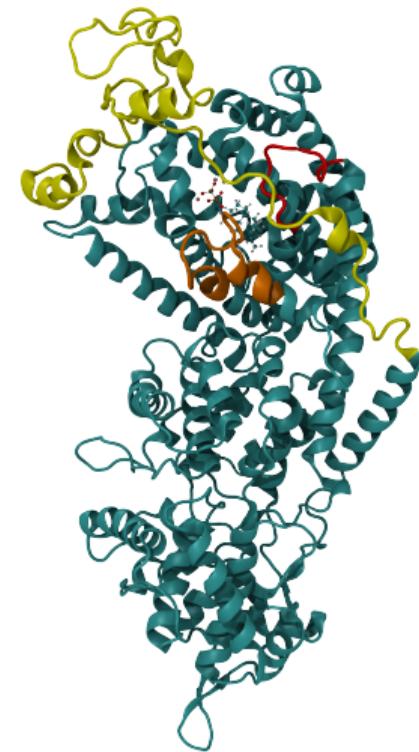
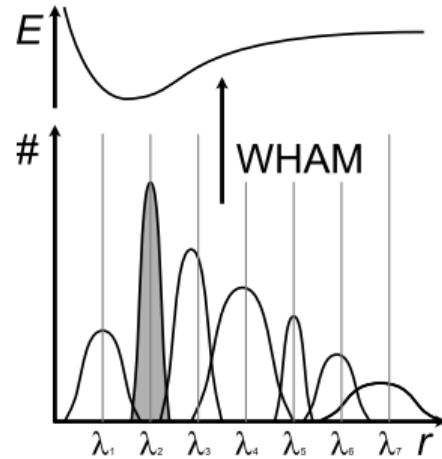
- Ross Amory
- Integrated Masters in Chemistry at the University of Nottingham
- Masters with Prof. Nicholas Besley
- PhD Student at the University of Nottingham
- Supervisors are
  - ▶ Prof. Chris Hayes,
  - ▶ Prof. Jonathan Hirst,
  - ▶ Dr. Christof Jäger
  - ▶ Dr. David Rogers



## Masters Project:



- Calculate an anharmonic correction for Raman Spectroscopy
- Implement 2<sup>nd</sup> 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

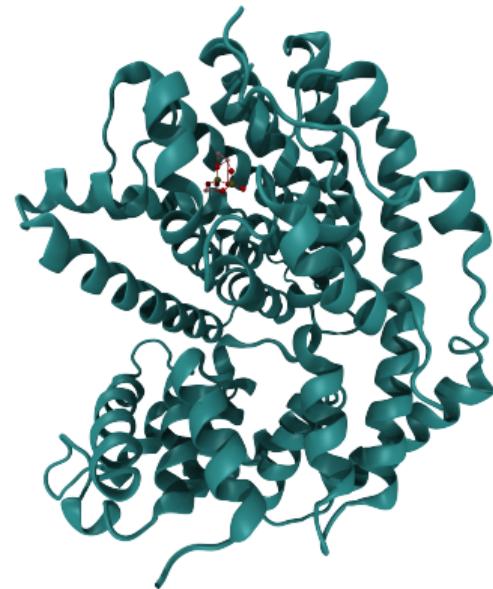
- **Self-taught** in Molecular Dynamics
- Experience of the entire workflow:
  - ▶ Homology Modelling with AlphaFold and YASARA
  - ▶ Docking using AD4 and Gnina
  - ▶ Simulations using AMBER
  - ▶ QM/MM Simulations with AMBER and QChem
  - ▶ Analysis using GBSA, Umbrella Sampling, clustering etc.

# The system

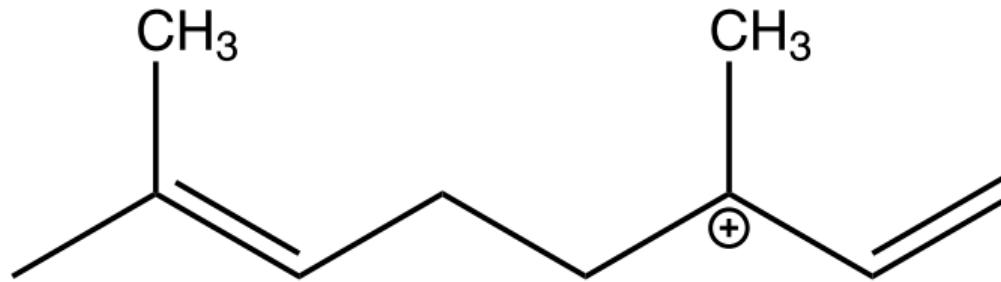
## Terpene Synthase “(+)-Bornyl Diphosphate Synthase”

- 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](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](https://github.com/pcyra2/MD_workshop/blob/main/Slides/main.pdf)

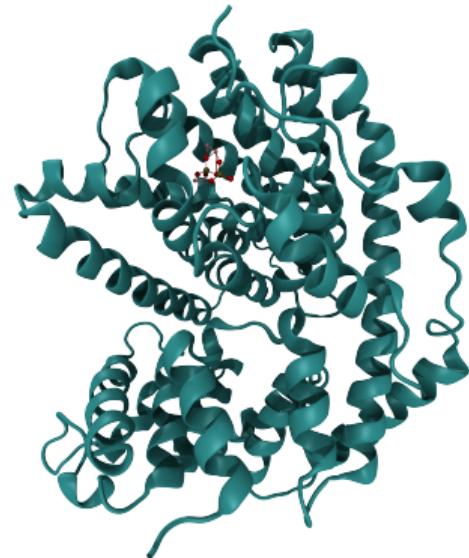
# Obtaining the Protein

There are many repositories for protein crystal structures:

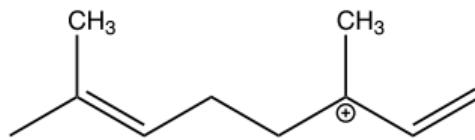
- <https://www.rcsb.org/>
- <https://www.uniprot.org/>
- <https://www.wwpdb.org/>

Our protein has the PDB code 1N23:

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



# Obtaining the Ligand

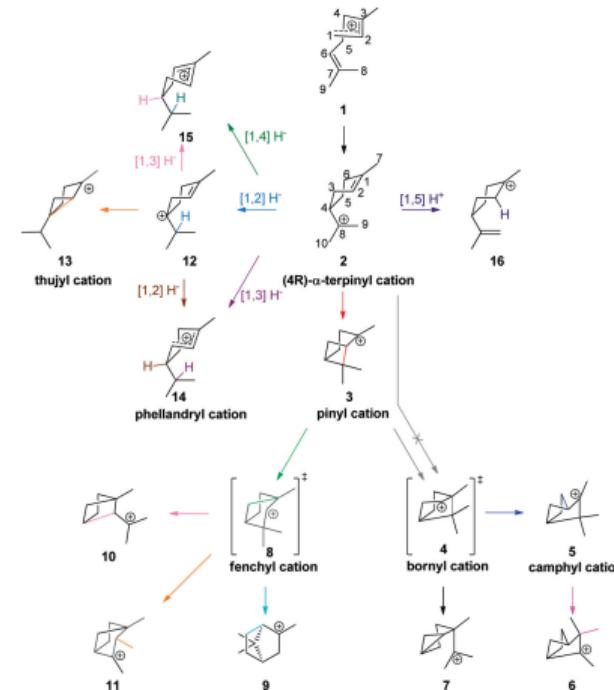


- Structure 1 from the reaction pathway
- Other intermediates from this reaction scheme should also work

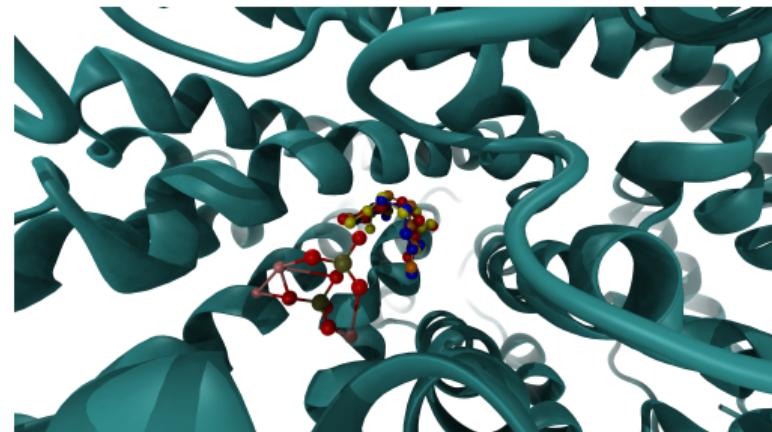
M. Weitman, D. Major, Challenges Posed to Bornyl Diphosphate Synthase:

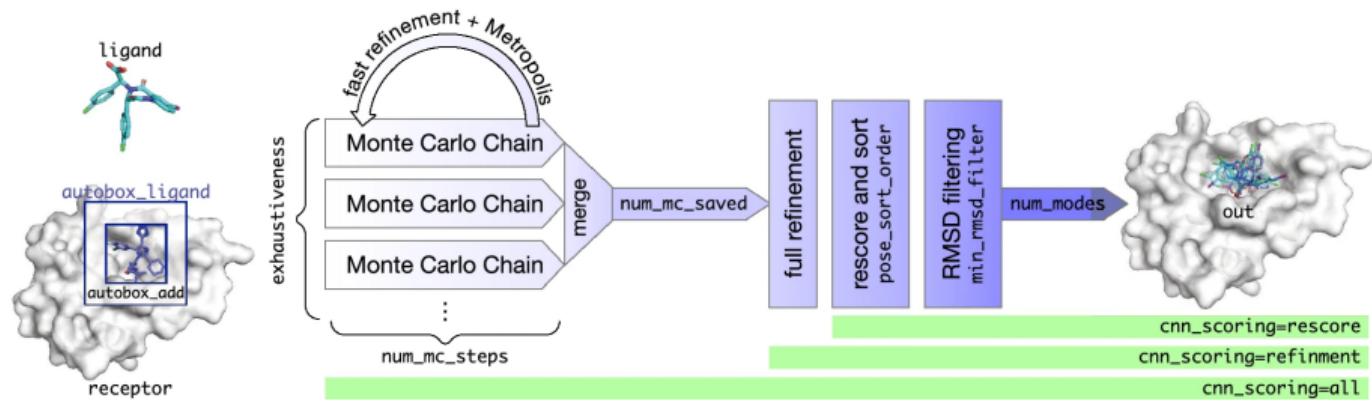
Diverging Reaction Mechanisms in Monoterpene, J. Am. Chem. Soc.,

2010, 132 (18), 6349-6360



- 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 an empirical scoring function
- Uses Convolutional Neural Networks to re-score 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)

```
"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

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

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 Newton's laws of motion and the positions are updated with respect to the time step.

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 Mg<sup>2+</sup> 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 parameters and coordinate files

# Box shape

- Solvent molecules dominate simulation time
- Reducing the 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

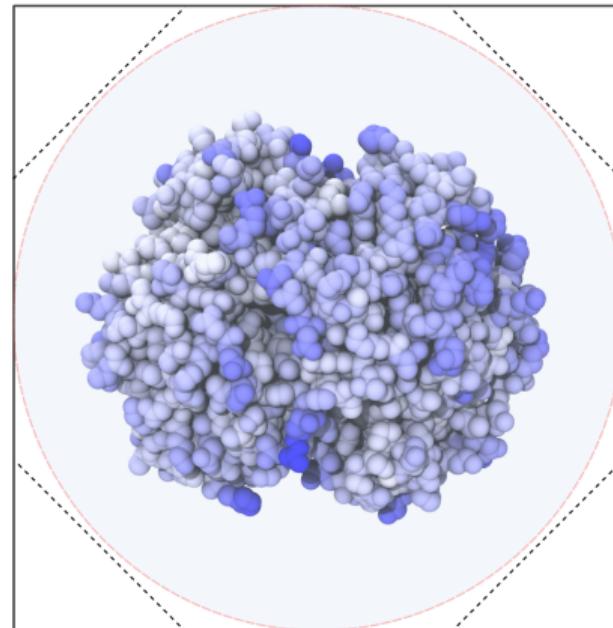
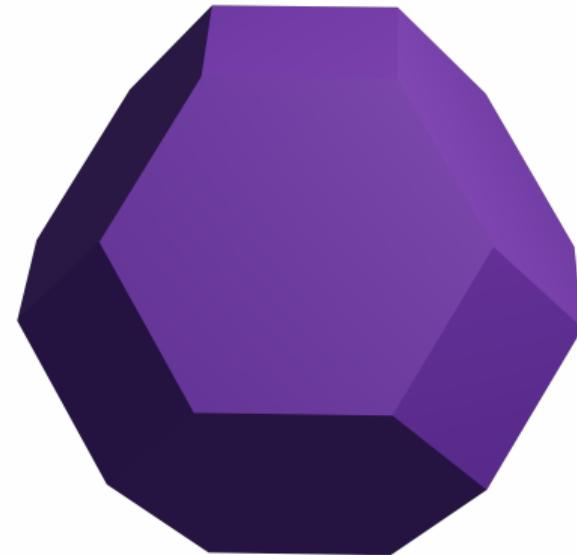


Figure: [https://computecanada.github.io/molmodsim-md-theory-lesson-novice/04-Periodic\\_Boundary/index.html](https://computecanada.github.io/molmodsim-md-theory-lesson-novice/04-Periodic_Boundary/index.html)

# Box shape

- Solvent molecules dominate simulation time
- Reducing the 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)
/
```

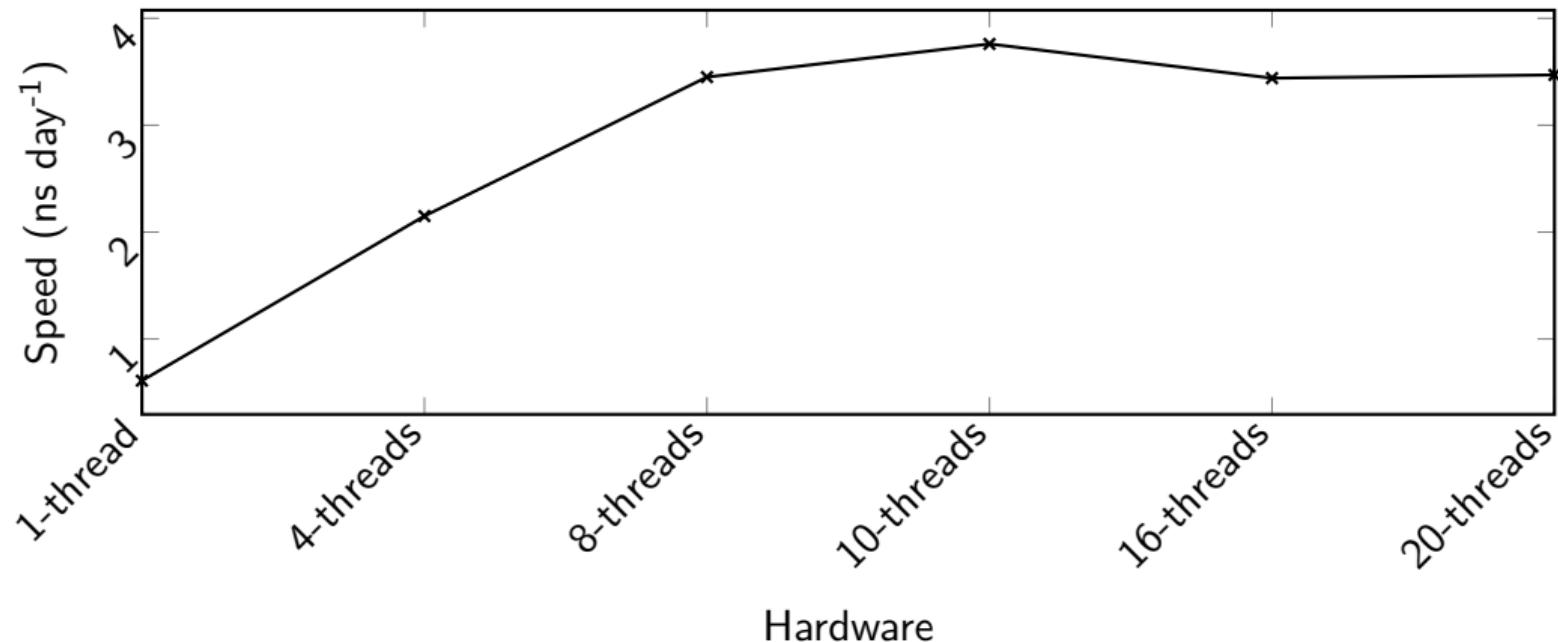


## bash command

```
$ # 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.
```

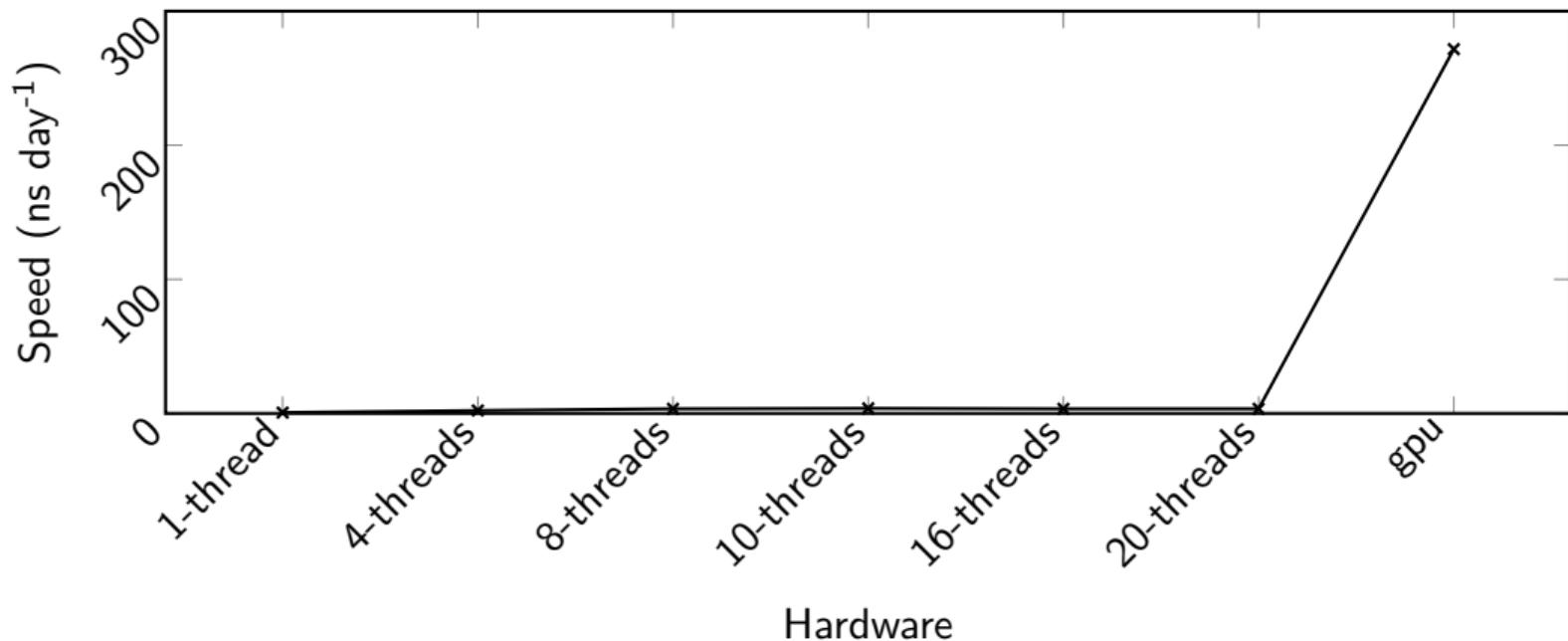
Molecular dynamics scales incredibly well with increasing computational power.

## Hardware scaling of MD simulations



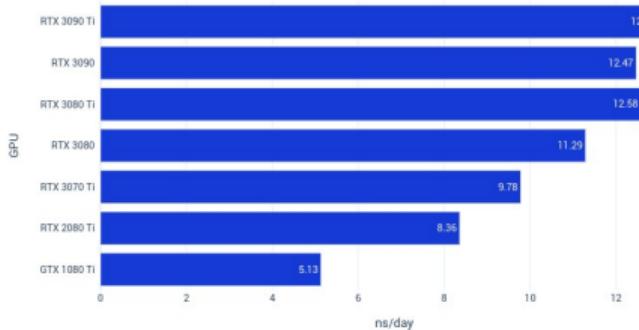
Molecular dynamics scales incredibly well with increasing computational power.

## Hardware scaling of MD simulations



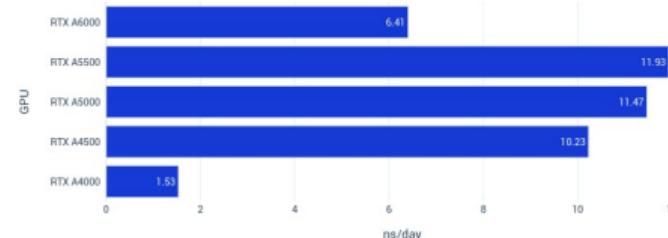
# GPU Benchmark

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

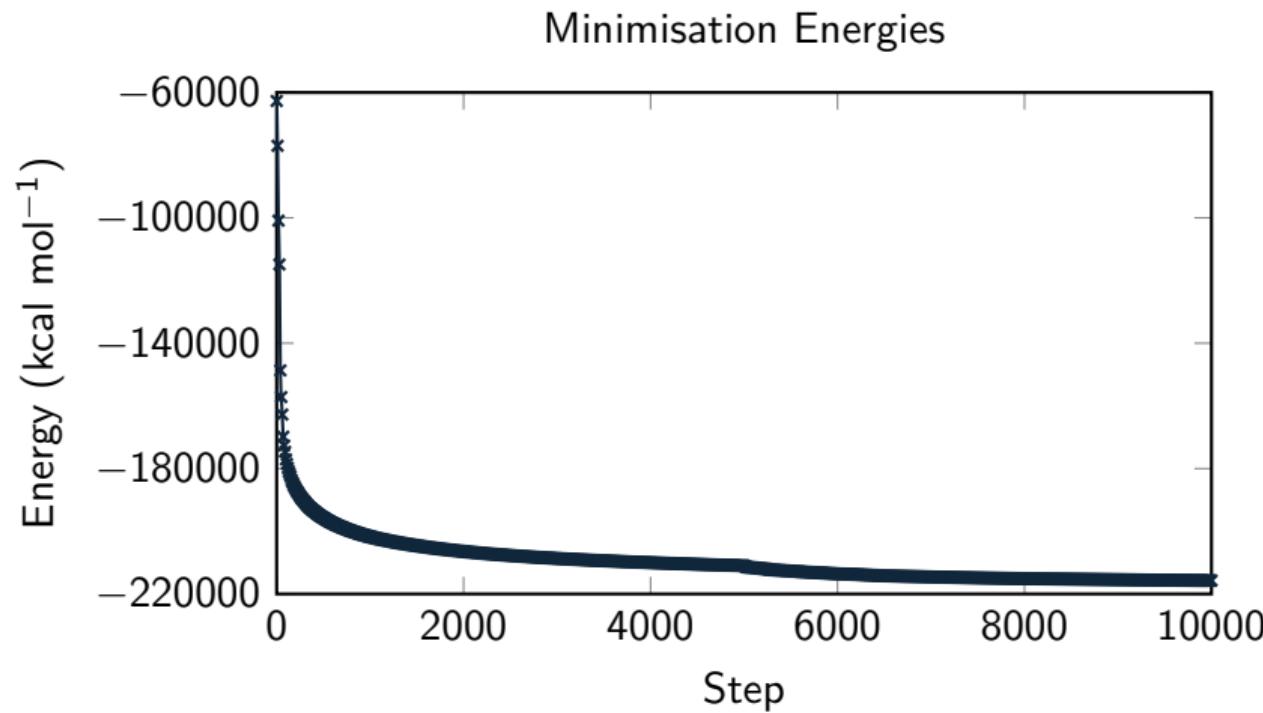
Table: Software

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

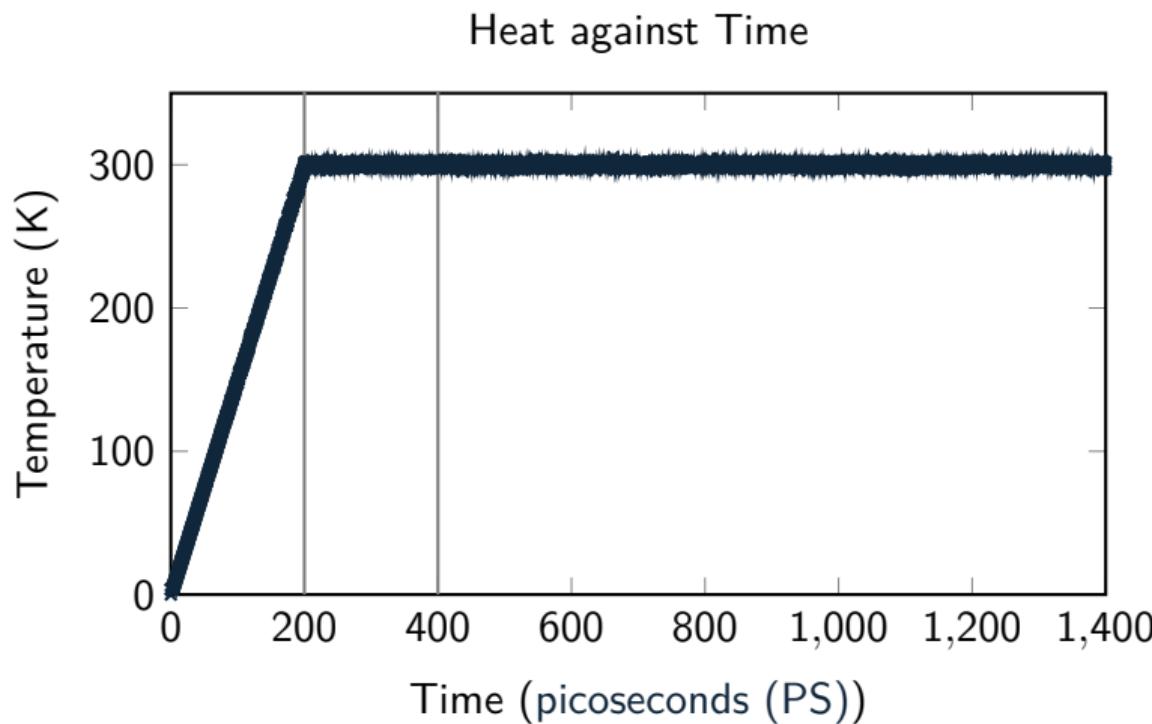
We need to perform analysis to ensure our simulations have run correctly.

You should always check:

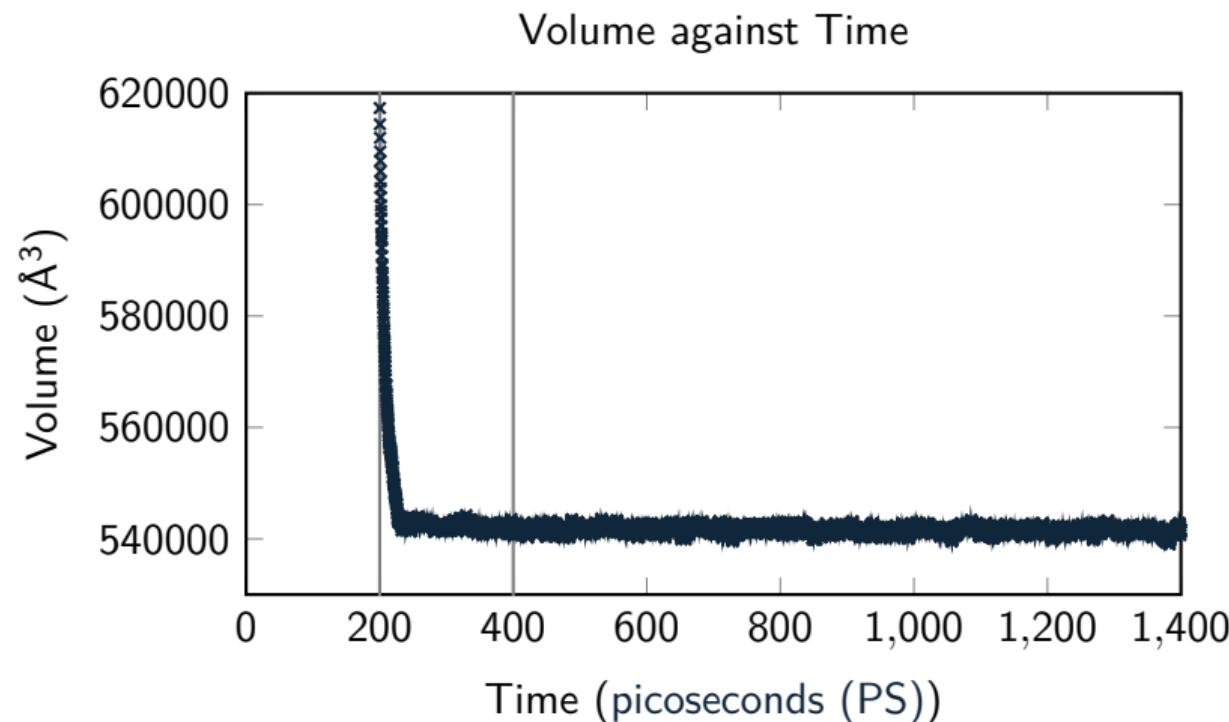
- Has the energy converged during minimisation?
- Is the temperature correct?
- Is the volume constant during NVT ensembles?
- Is the energy stable during simulations?
- Is the protein stable during simulations?

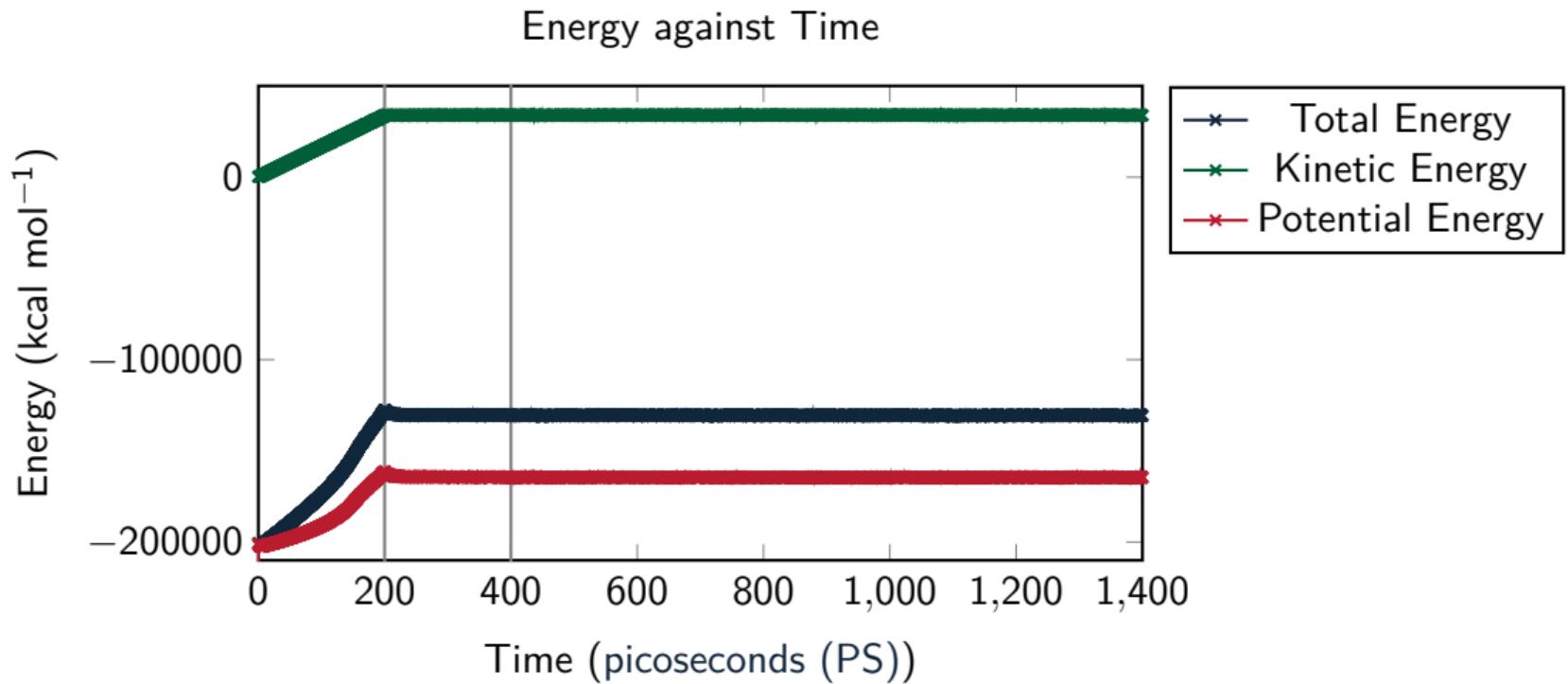


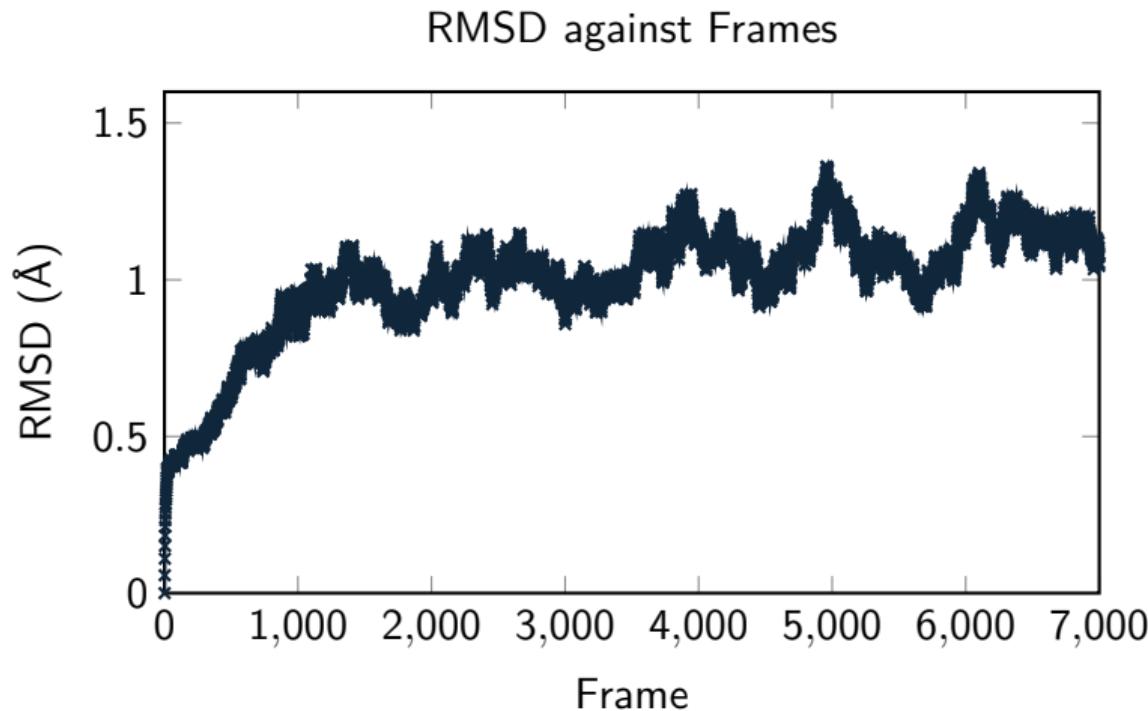
# Temperature Check



# Volume Check



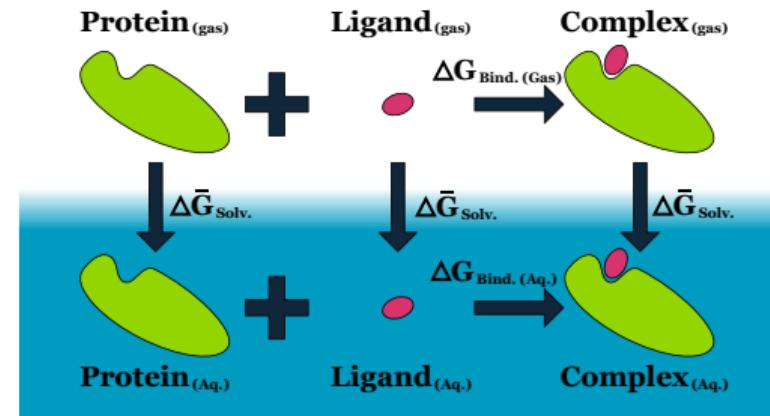




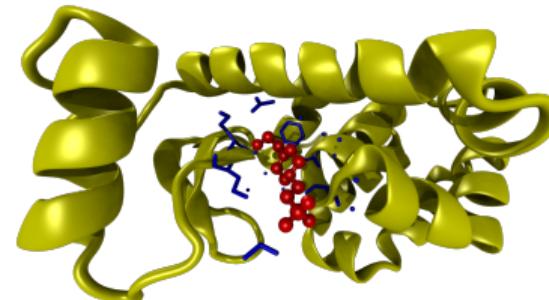
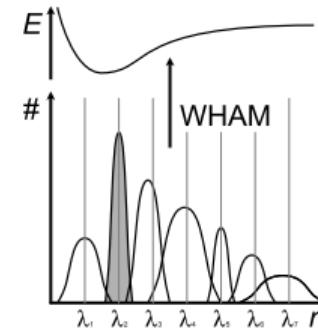


- Solvent-solvent interactions heavily dominate Dynamics energies
- Impossible to directly extract binding energies
- Use thermodynamics cycles to approximate relative energies

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



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



# Free energy calculation

To use the “MMPBSA.py” code, you first need to split the parameter file into:

- Complex (no solvent)
- Receptor (ligand, Mg<sup>2+</sup> and diphosphate ions)
- Ligand

This can be done using “ante-MMPBSA.py”

Further reading: <https://ambermd.org/tutorials/advanced/tutorial3/>



## bash command

```
$ mpirun -np NUM_PROCESSORS MMPBSA.py.MPI -O -i INPUT_FILE.in -o
  ↵ gb_results.dat -sp SOLVATED_COMPLEX.parm7 -cp complex.parm7
  ↵ -rp receptor.parm7 -lp ligand.parm7 -y
  ↵ PRODUCTION_TRAJECTORY.nc
```



- We have introduced a basic workflow for obtaining free energies of binding
- We have taken many short-cuts in the interest of time
- The principles in this workshop can be transferred to other software packages
- Relative free energies require a second calculation with a different ligand to be useful!

# Acknowledgements

Although this workshop is my own work, It would not have been possible without the help of...

## My supervisory team:

- Prof. Chris Hayes
- Prof. Jonathan Hirst
- Dr. Christof Jäger
- Dr. David Rogers

## My funding group:

- BBSRC DTP
- The University of Nottingham

## My colleagues:

- Dr. Bang Huynh
- Aiden Cranney
- Comp. Chem. Nottingham!