Biotechnology and Biological Sciences Research Council

University of Nottingham
UK | CHINA | MALAYSIA

Molecular Dynamics Workshop

Ross Amory

School of Chemistry, University of Nottingham

Thursday 4th May, 2023

Contents

1 How to obtain software 1
1.a Anaconda 2
1.b AMBER 2
1.c Acpype 3
1.d gnina 3

2 Workshop outline 4
2.a Goals 4
2.b Software 4

3 Theory 6
3.a Molecular Dynamics 6
3.b Docking 7
3.c GB/PBSA 7

4 Tasks 7
4.a Initial Tasks 7
4.b Docking 9
4.c Molecular Dynamics Preparation 9
4.d Molecular Dynamics 10
4.e Free Energy calculation . . . 10

5 Cheat codes 14
5.a Initiating the file system . . 14
5.b Geting the protein structure . 14

6 Bibliography 14

1 How to obtain software

Here we will go through the steps to install all required software for this workshop. It is presumed that you are working either within a standard linux terminal, or using Windows subsystem for

linux 2 (WSL2) from within windows. If you are using any other set up, you will have to find out how to install the required software seperately.

1.a Anaconda

Install python and anaconda using the following commands. You should do this within a WSL2 terminal if you are not natively working within linux.

Listing 1.1: Commands to install anaconda and python.

bash command

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-
↪ x86_64.sh
↪ .
$ sh Anaconda3-2023.03-1-Linux-x86_64.sh
$ # Accept the license agreement and use all defaults suggested.
```

1.b AMBER

Install AmberTools22 using anaconda as this is the simplest way and suitable for this tutorial.

Listing 1.2: Commands to install AmberTools22

bash command

```
$ conda create --name AmberTools22
$ y # proceed with creating the AmberTools22 conda
↪ environment
$ conda activate AmberTools22
$ conda install -c conda-forge ambertools=22 compilers
$ y # proceed with installing
```

1.c Acpype

Install acpype using anaconda. In theory this will also install ambertools, however I find that having a separate install for both is best as you have greater control when just using amber.

Listing 1.3: Commands to install Acpype

bash command

```
$ conda create --name acpype
$     y   # proceed with creating the environment
$ conda activate acpype
$ conda install -c conda-forge acpype
$     y   # proceed with installing
```

1.d gnina

Below is the quick way of installing gnina. It simply downloads a pre-compiled binary of the software for an individual to use, however it is strongly recommended that you download the source code and compile it yourself for a more optimised experience.

Listing 1.4: Commands to install gnina.

bash command

```
$ wget                                                              ↩
   ↪ https://github.com/gnina/gnina/releases/download/v1.0.3/gnina
$ chmod +x gnina
$ # It is then recommended to add this to your PATH, .bashrc or     ↩
   ↪ modules in order to easily run this later in the workshop.
```

2 Workshop outline

2.a Goals

The main goal of this workshop is to give you a brief overview of what a standard molecular dynamics (MD) workflow would look like. We shall be covering some of the core techniques used to set-up and run a variety of different calculations including **molecular docking**, **molecular dynamics** and **free energy** calculations. We shall also briefly cover some analysis techniques available and some other resources available to further your understanding of the techniques used throughout.

The steps of the workshop are as follows:

- Locate and download the crystal structure
- Dock the ligand into the active site
- Parameterise the ligand
- Create molecular dynamics input files
- Run a simple dynamics simulation
- Perform simple trajectory analysis
- Calculate the binding energy of the ligand

2.b Software

The key software that we will be using for this workshop all free to use and should be simple to access and use. Most of the software however requires a basic understanding of how to use the command-line interface (CLI). It is possible to use all of the software from within windows, however it is recommended to use linux for the smoothest experience.

The required software for this workshop is as follows:

- anaconda
- WSL2 (windows only)
- AMBER [a]
- VMD
- gnina
- acpype [b]

Although not required for this workshop, some other software that is available to assist with the MD workflow are as follows:

- CHARMM (MD package)
- gromacs (Open source MD package)
- CGenFF (Forcefield parameterisation tool for the CHARMM forcefield)
- pyMol (graphical user interface (GUI) for molecular visualisation and manipulation)
- AutoDock4 (Open source docking software)
- YASARA (GUI for the whole MD workflow)
- Alphafold (Open source A.I. homology modelling software)

[a] It is recommended that you install the full (paid) version of amber for production dynamics due to the GPU support, however for individual machines and the purpose of this workshop, the conda binary distribution is okay.

[b] A web server version of acpype is available here for users unable to install, however installing this is highly recommended where possible.

3 Theory

In this section, a brief introduction to the theory behind the activities performed in the workshop. This will not however be a comprehensive theory section however, and further reading is strongly recommended to understand the methods.

3.a Molecular Dynamics

Due to the size of biological systems, current computational power is unable to simulate every atom within an enzyme at a true quantum mechanical (QM) level and so some approximations must be made in order to simplify the system. The traditional method for simulating large biological molecules such as enzymes is to use a classical physics (Molecular Mechanics MM) approach (Equation: 3.1), which includes using empirical force fields derived from classical physics in order to describe the system. The use of classical physics allows much simpler calculations and so increases computational efficiency dramatically, allowing the simulation of millions of atoms in a relatively short time scale.

$$\begin{aligned} E_{Total} &= E_{Bonded} + E_{Non-Bonded} \\ E_{Bonded} &= E_{Bond} + E_{Angle} + E_{Dihedral} \\ E_{Non-Bonded} &= E_{Electrostatic} + E_{vanderWaals} \end{aligned} \tag{3.1}$$

Molecular mechanics requires empirical forcefields(1) to describe the behaviours of different atoms (and more importantly biological residues) to describe the energies of the different interactions within a system and so selecting a suitable forcefield which has been parametrised for your use case is important for achieving accurate results. For enzymes, the two most commonly used forcefields are CHARMM(2) and AMBER(3) which were both parametrised for proteins and nucleic acids.

These MM methods can be combined with QM methods in order to calculate the reaction center with high accuracy QM theory whilst also including interactions with the wider environment, specifically the electrostatic, dispersion and structural effects, which can control the structure of the reaction site and the thermodynamics of the system. In QM/MM you split the system into two or three sections (usually three) with your reaction site and any atoms which are known to be directly involved being put into the QM region. Then a thin layer around this region is included as a hybrid zone which allows for communication between the two methods and used when the zone boundary intersects a covalent chemical bond. These atoms are usually calculated at both levels of theory. Finally the rest of the system is treated at the simplest level of theory (MM) (Figure: **??**).

3.b Docking

3.c GB/PBSA

4 Tasks

4.a Initial Tasks

Task 4.1: Set up the directory structure for the workshop.

Create directories for:

(a) Structure files

(b) Docking simulations

(c) MD simulations

(d) Simulation analysis

Firstly, like whenever starting any new project, a working directory needs to be created. This can be done either using the CLI in a terminal using the '`mkdir`' command, or using a GUI file manager. Commands to do this can be found in the cheat codes Listing 5.1.

Listing 4.1: Folder structure as shown by the `tree` command.

bash output

```
MD_Workshop
├── Structures
├── Docking
├── Dynamics
├── Analysis
└── Free_Energy
```

Task 4.2: Obtaining the structure files.

(a) Obtain the protein crystal structure

(i) Visit the protein database website

(ii) Find the protein using code 1AZX

(iii) Download the crystal structure

(b) Save the ligand coordinate files located in the workshop files.

There are a few places where you can find and store crystal structure coordinates, however the main two that are used are rscb and uniprot. These websites are powerful tools, and often group multiple versions of the same protein. Most proteins are published on the rscb database however for unresolved crystal structures, uniprot sometimes also contains the alphafold() homology model structure.

Although we will not be covering the theory or methods of homology modelling in this workshop, being aware of the technique could potentially be useful when working with an un-resolved protein. The method allows for an approximate structure to be generated using similar proteins as templates. A homology modelling program with growing popularity is the alphafold() program which uses artificial intelligence to estimate a crystal structure from a FASTA sequence.

To start this workshop, you will need to visit the rscb website and download the 'pdb' file for the antithrombin/pentasaccharide protien complex which can be found using the code 1AZX. If you google the protein code, it usually comes up as one of the first options also. If you have any problems with this then use the commands found in the cheat codes Listing 5.2.

4.b Docking

4.c Molecular Dynamics Preparation

4.c.i Parameterisation

To run a molecular dynamics simulation, you first need to parameterise all of the atoms in the system. There are multiple different molecular dynamics force fields that contain standard parameters for common amino acids and some other small or simple molecules. For bigger or more complicated molecules however, you will need to create your own custom parameters.

Although there is a rigorous method that can be undertaken to correctly parameterise complex ligands, including comparing the parameters with ab-initio calculations, we shall be using an automated method for speed and simplicity. [a]

We shall be using the `acpype()` automated script to parameterise the ligands in this workshop, which combines the parameterisation tools offered by amber into a simple to use script. We shall be using `bcc` charges and the `gaff` atom typing so that it will be compatible with the 'FF14SB' forcefield that we shall be using for the simulation.

Listing 4.2: How to use acpype.

[a] You can learn more about the rigorous process of parameterisation here.

bash command

```
$ conda activate acpype
$ acpype -i LIG.pdb -c bcc -n NET_CHARGE -a gaff
```

4.d Molecular Dynamics

4.e Free Energy calculation

There are many different ways of calculating the Gibbs free energy of binding for a ligand into a protein, however the simplest and cheapest way is to use either the Molecular Mechanics/Generalized Born Surface Area (MM/GBSA) or Molecular Mechanics/Poisson–Boltzmann Surface Area (MM/PBSA) method. The scientific theory behind these methods can be found in Section 3.c.

There are two ways to run a MM/GBSA or MM/PBSA calculation. There are two ways of performing these types of calculations, as you can either 3 independent simulations of the ligand, protein and complex in order to calculate the $\Delta G_{\text{Solv, Lig.}}$, $\Delta G_{\text{Solv, Rec.}}$ and $\Delta G_{\text{Solv, Complex}}$ directly, and the $\Delta G_{\text{Vaccum, Bind.}}$ is calculated as the interactions between the ligand and protein from the complex simulation. The other way of performing this calculation however, is to perform a single simulation of the complexed system. It is then possible to separate each of the energies out post-hoc. Advantages of this method are that it only requires a single simulation, meaning it is much less computationally intensive so you can potentially run a longer simulation. This type of calculation also has been shown to increase accuracy. Due to the ease and accuracy of the single simulation method, we are going to use this for calculating the binding energy of the ligand.

4.e.i Setup

Firstly, you need to copy the production-dynamics trajectory file from the previous task to the 'Free_Energy's directory, along with the parameter file for the solvated complex. You should then rename the parameter file to indicate that it is solvated, as you are about to remove the

solvent in the next step. You can then use the `ante-MMPBSA.py` script to prepare the stripped parameter files for the complex, receptor and ligand respectively.

Listing 4.3: Commands to setup the GBSA calculation.

bash command

```
$ ante-MMPBSA.py -p SOLVATED_COMPLEX.parm7 -c complex.parm7 -r
  ↪ receptor.parm7 -l ligand.parm7 -s :WAT,Na+ -n :LIGAND_NAME
  ↪ --radii=mbondi2
```

The input variables for the `ante-MMPBSA.py` are as follows:

Option	Value	Type
-p	Solvated complex parameter file	Input
-c	Stripped complex parameter file	Output
-r	Stripped receptor/protein parameter file	Output
-l	Stripped ligand parameter file	Output
-s	Strip mask for the solvent and counter ions.	Variable
-radii	Atomic Radii parameter set	Variable

4.e.ii Calculation

Once the stripped parameter files are generated, one final input file is required in order to run the MM/GBSA or MM/PBSA calculation. This file has a similar format to the input files used by `sander` and other AMBER programs.

File 4.1: Input file for required for the MM/GBSA program.

MMPBSA.py file

```
GBSA input file
&general
        startframe=1, endframe=100000, interval=10, \\ This allows
        ↪ you to speed up the calculation by only calculating every
        ↪ n bins.
        strip_mask=:WAT:Na+, \\ The AMBER mask for the solvent and
        ↪ counter ions stripped in the previous step.
        keep_files=0, \\ Delete temporary files after the calculation
        ↪ completes.
/
&gb
        igb=2, \\ Specifies the generalised Born method to use.
        saltcon=0.100, \\ Salt concentration for the implicit solvent (M)
/
```

Once all the files are in place, the MM/GBSA calculation can begin. Use the following command with your file names to run the calculation. Make sure you change the variables:

- `NUM_PROCESSORS` to the number of available threads on your machine
- `INPUT_FILE` to the name of your newly created MM/GBSA input file
- `SOLVATED_COMPLEX` to the name of your original solvated complex parameter file
- `PRODUCTION_TRAJECTORY` to the name of the production trajectory file

Listing 4.4: Commands to run the GBSA calculation.

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
```

4.e.iii Analysis

The data from the MM/GBSA calculation is outputted to the 'gb_results.dat' file.

5 Cheat codes

If you have any problems with any of the tasks within this workshop, here is a list of commands which should help you to complete the task. We recommend using this as a last resort however, as simply compying and pasting commands doesnt allow you to learn the methodology.

5.a Initiating the file system

Listing 5.1: Linux commands to initiate your local file structure.

bash command

```
$ mkdir MD_Workshop
$ cd MD_Workshop
$ mkdir Structures
$ mkdir Docking
$ mkdir Dynamics
$ mkdir Analysis
$ mkdir Free_Energy
```

5.b Geting the protein structure

Listing 5.2: Linux command to download the crystal structure.

bash command

```
$ wget https://files.rcsb.org/download/1AZX.pdb .
```

6 Bibliography

References

1. J. W. Ponder, D. A. Case, in, pp. 27–85, DOI 10.1016/S0065-3233(03)66002-X.

2. K. Vanommeslaeghe *et al.*, CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.*, NA–NA, DOI 10.1002/jcc.21367 (2009).

3. J. A. Maier *et al.*, ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theo. Comput.* **11**, 3696–3713, DOI 10.1021/acs.jctc.5b00255 (Aug. 2015).

Acronyms

CLI command-line interface

MM/GBSA Molecular Mechanics/Generalized Born Surface Area

GUI graphical user interface

MD molecular dynamics

MM/PBSA Molecular Mechanics/Poisson–Boltzmann Surface Area

WSL2 Windows subsystem for linux 2