Molecular docking to Molecular Dynamics - From the Setup to the Analysis

Luis Daniel Solis Rodriguez*

Università degli Studi di Perugia
luisdaniel.solisrodriguez@studenti.unipg.it

September 6, 2024

I. Introduction

Molecular docking and molecular dynamics are key techniques in drug discovery and molecular biology. Molecular docking predicts how a small molecule (ligand) binds to a target protein, helping to identify potential interaction sites and estimate binding affinity, crucial for drug design. There are two primary types of docking approaches:

- Flexible Docking: In this method, both the ligand (e.g., a drug) and the receptor (e.g., a protein) can undergo conformational changes during the docking process. This approach is more accurate as it considers the dynamic nature of biological molecules, allowing for adjustments in their shapes and positions to better fit together. Flexible docking is especially useful when dealing with highly dynamic or flexible molecules.

- Semiflexible Docking: This approach allows for some flexibility but is more constrained compared to fully flexible docking. Typically, the ligand is allowed to flex, but the receptor is kept rigid. This method is less computationally intensive than flexible docking and is often used when the receptor's structure is well-known and its flexibility is less critical.

Molecular dynamics simulations provide a dynamic view of these interactions over time, revealing the stability and conformational changes of the protein-ligand complex in a realistic environment.

Together, these methods are vital for understanding molecular interactions and guiding the design of more effective and selective drugs.

The $5-HT_{2A}$ receptor is a serotonin receptor (5-hydroxytryptamine receptor) primarily involved in various neurological processes. It is known to bind with several ligands, including serotonin (5-HT), its natural neurotransmitter. In addition to serotonin, the 5-HT2A receptor is also a key target for a variety of psychoactive compounds such as LSD, psilocybin, and N,N-DMT (dimethyltryptamine). These interactions are significant in regulating mood, cognition, and perception, and are implicated in the effects of psychedelic substances.

IsoDMT is a structural analog of the psychedelic compound N,N-DMT, which is known for its potent hallucinogenic effects. Unlike N,N-DMT, isoDMT exhibits a different binding profile to serotonin receptors, particularly the $5-HT_{2A}$ receptor, which is responsible for the hallucinogenic effects of many psychedelics. Early studies suggest that isoDMT may have therapeutic potential

*

similar to N,N-DMT, such as in treating mood disorders, depression, or anxiety, but with a reduced risk of hallucinations. This makes isoDMT a promising candidate for development as a novel therapeutic drug, offering the therapeutic benefits of psychedelics with fewer psychoactive side effects.

II. Setup

Before beginning, it's important to organize your files and folders for consistency throughout the tutorial, ensuring you can follow along smoothly. You can download the necessary molecules for this tutorial from the following github where both the protein and ligand files are stored in the "molecules" folder.

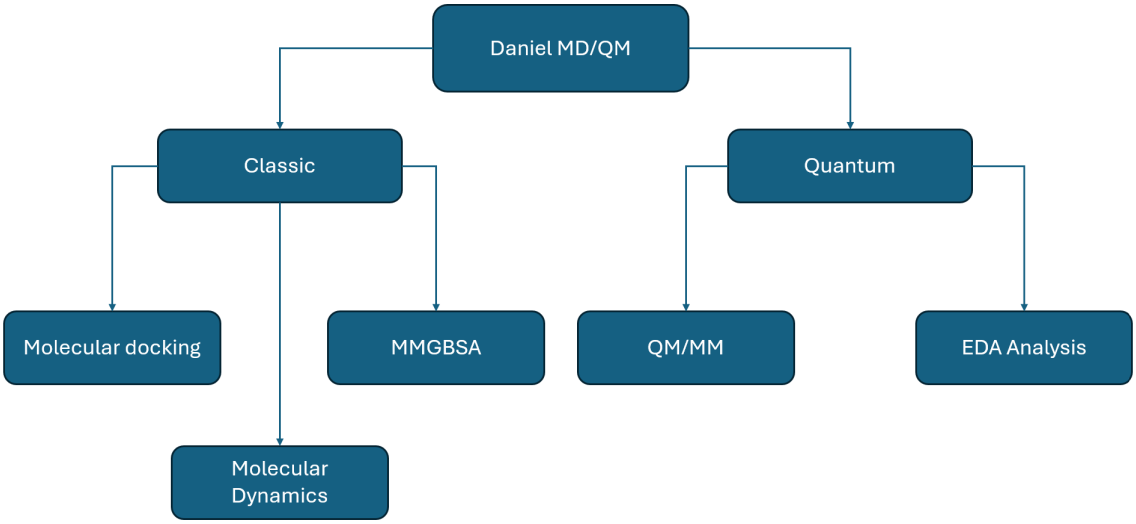


Figure 1

Additionally, the required software, AmberTools, should be installed. AmberTools isn't officially supported on Windows, but you can download it through Ambertools and extract the package to get started.

- tar xvf AmberTools23.tar.bz2

After that you enter the source directory with

- cd amber22_src

And finally compile the package with

- make install

And to make easier to manipulate the program you add to your /.bashrc thse lines:

- export AMBERHOME=$HOME/amber22
- export PATH=$AMBERHOME/bin:$PATH
- source /.bashrc

Lastly, verify your AmberTools installation by running "amber.version" in the terminal.

Next, download AutoDock Tools from the official Autodock Vina. Follow the installation instructions according to your operating system (Windows, macOS, or Linux). The site also provides access to source code and examples for molecular docking. If the installation was successful, you should see the following AutoDock Vina icon on your system:

Figure 2

After installation, launch AutoDock Tools by either double-clicking the AutoDock icon or entering adt in your terminal (if the PATH is correctly set). Once the program is launched, you will be greeted with the main window interface of AutoDock Tools Figure 3, which provides the workspace for preparing molecular docking simulations. From this interface, you can load your protein and ligand files, prepare them for docking, and configure the grid box to define the docking region.

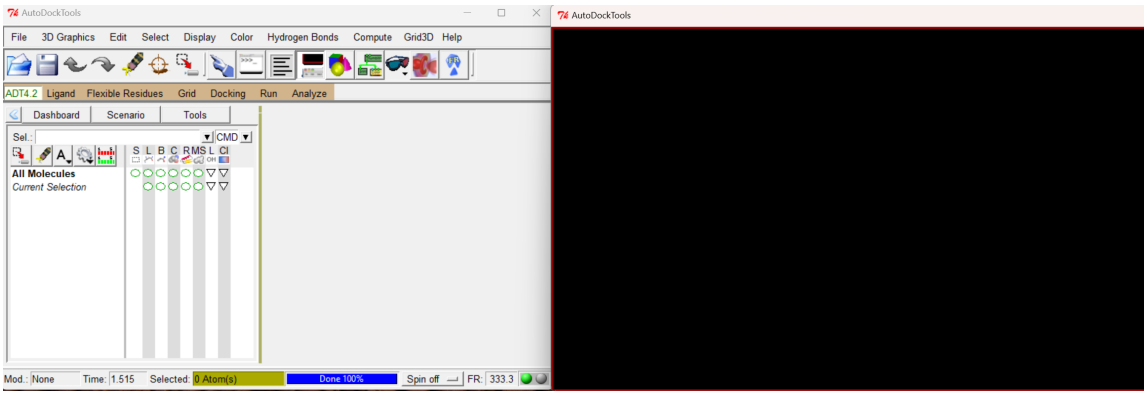


Figure 3

I. Molecular Docking. Loading of the molecules and use of the software Autodock

The RCSB Protein Data Bank (PDB) is a comprehensive archive that hosts all PDB structures obtained annually from scientific experiments. Each PDB entry is assigned a unique four-character alphanumeric ID. The website provides users with tools to visualize and download 3D structures, as well as analyze them through features such as sequence alignment, protein quality assessment, and symmetry analysis. Today, we will focus on a specific system that is currently under investigation. Therefore, it is essential to conduct an analysis of potential binding sites and perform a molecular docking study. The PDB-ID is: 6WGT [1], which is the $5-HT_{2A}$ receptor

bound to serotonin. You can download the protein $5 - HT_{2A}$ and the ligand in this case isoDMT in the next link

the first is to **load the protein** in the next way:

- Go to File > Read Molecule and select your protein structure file.
- **Remove water molecules** Water molecules can interfere with docking. Go to Edit > Hydrogens > Add Polar Only to add hydrogens only to polar atoms.
- To notice, you can choose to make some parts of the protein flexible. This is more advanced but useful for certain docking scenarios. Typically, the protein is kept rigid and ligand flexible (semiflexible docking).
- Add charges: Go to Edit > Charges > Compute Gasteiger. This adds the necessary charges to the protein.
- After this is necessary to save the protein in PDBQT format. Go to File > Save as PDBQT. This format includes atomic coordinates, charges, and AutoDock atom types.

Now, for the ligand, it is crucial to calculate the necessary charges. The recommended method is using ESP charges (Electrostatic Potential), but it can also be done with AutoDock Tools if needed. Once the charges are calculated, the next step is to **prepare the ligand**. This process involves ensuring that the ligand has the correct atom types, charges, and bond rotations, and that it is formatted properly for molecular docking. AutoDock Tools provides specific commands to guide you through this preparation process efficiently.

- First load the ligand. Go to File > Read Molecule and select your ligand file.
- After this continues to set torsions. Ligands typically have rotatable bonds that need to be defined. Go to Edit > Rotatable Bonds > Choose Torsions and you can see that the rotatable bonds are gonna be green and in red those you cannot. Autodock Tools will automatically detect and allow you to adjust rotatable bonds how it can be seen in the Figure 4.

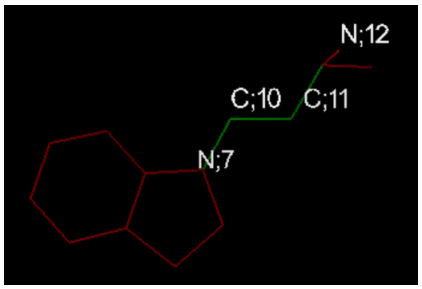


Figure 4

- Now it is important the charges of the ligand. Just like with the protein, add Gasteiger charges via Edit > Charges > Compute Gasteiger.
- Save the ligand in PDBQT format by going to File > Save as PDBQT.

Now, it is important to **set up the grid** where the possible binding site will be searched. The grid defines the volume where AutoDock Vina will look for interactions between the ligand and the receptor. Based on previous studies[2], we expect the binding site to be located inside the

protein. Therefore, it is crucial to position the grid box accurately to encompass the relevant region of the protein where the binding site is hypothesized to reside, ensuring an effective docking simulation.

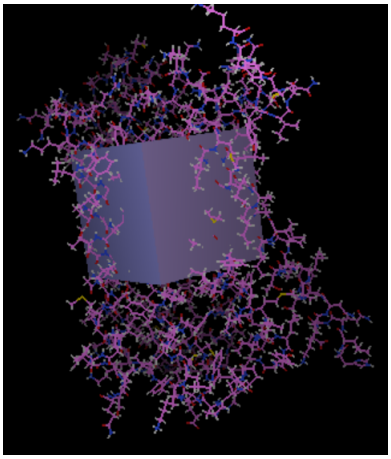


Figure 5

To do that:

- Go to Grid > Macromolecule > Choose, and select your prepared protein (PDBQT file).

- Then, go to Grid > Grid Box. A grid box will appear around your protein.

- Adjust the grid box size and position to cover the binding site where you expect the ligand to bind. The box size and center can be modified by entering values or using the mouse to move the box. Also a window is gonna show up where you can play with the size of the box and the position of that. In the Figure 6 you can see the the dimensions of the box that you can modified and the coordinates of the centering of the box. For simplicity of this document follow the number that you can see in Figure 6.

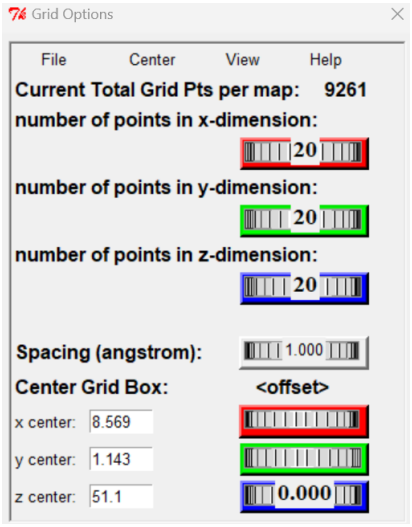


Figure 6

- After setting up the grid, save the grid parameter file by going to Grid > Output > Save.

After defined the grid box continues to configure the docking parameters.

- Go to Docking > Macromolecule > Set Rigid Filename, and select the protein PDBQT file
- Go to Docking > Ligand > Choose, and select the ligand PDBQT file.
- Set the number of runs and other parameters (optional). For this document, the default settings are sufficient.
- Save the docking parameter file by going to Docking > Output > Lamarckian GA > Save.

Finally, is just to run the autodock.

- AutoDock is typically run from the command line. Open a terminal or command prompt.
- Navigate to the directory where your PDBQT, also you can find them in the folder molecular docking.
- To run Vina is better to define a .txt file to put in there the options that you want for the simulation. The example of that document you can find it in the github. Figure 7 shows us how to put the name of the receptor and ligand, also the name and format of the output (important), as well as the size of the box and the spacing, which is necessary to define perfectly the size of the box. Also you defined there the coordinatees of the box and the exhaustiveness that is not more that the computational effort you put in to search the binding sites, for this exercise is enough with 8 in exhaustiveness.

```
receptor = receptor.pdbqt
ligand = lig.pdbqt

out = out.pdbqt

center_x = 35.3
center_y = 34.1
center_z = 55.3

size_x = 20
size_y = 20
size_z = 20

num_modes = 50

spacing = 1.0

exhaustiveness = 32

cpu = 16
```

Figure 7

- After setting up the grid, save the grid parameter file by going to Grid > Output > Save GPF.
- Run AutoDock using the command: C:\Documents\ProgramFiles(x86)\The Script Research Institute\Vina\vina.exe –config conf.txt –log log.txt

III. Results and Discussion of the molecular docking

After completing the molecular docking simulation, you can visualize the results using AutoDock Tools. To do this, go to Analyze > Dockings > Open Autodock Vina Results, and select the output .pdbqt file. This will display the nine potential binding positions. Among these, focus on the two positions with the lowest energy scores, as they are the most likely to represent spontaneous binding sites. These results, presented in Table 1, guide further analysis. By examining these conformations, you can decide which are suitable for molecular dynamics (MD) simulations, which are computationally expensive. For visualization, refer to Figure 8, where an example ligand position is displayed.

After choosing the top two binding sites, the next step involves incorporating them into a lipid bilayer for further analysis, using tools like CHARMM-GUI to simulate membrane environments.

Table 1: *Representative binding poses of the molecular docking simulation* $5 - HT_{2A}$*R-isoDMT*

Binding Site	**Energy**(kcal/mol)
1	-6.5
2	-6-5
3	-6.3
4	-6.2
5	-6.1
6	-6.1
7	-6.0
8	-6.0
9	-6.0

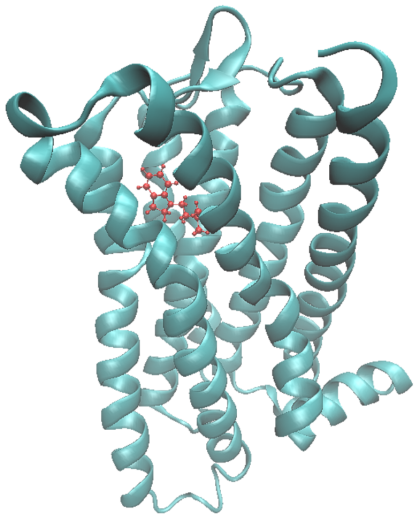


Figure 8

IV. Use of CHARMM-GUI to introduce protein into membrane

- The first is to make an account in CHARMM-GUI which is easy
- Now, in the section Input Generator you have to choose membrane builder > Bilayer builder, where you have to put the PDB generated after the docking, in this case you can use the pdb file that I give you in the folder docking for a faster use.
- After that you have to choose the protein and the ligand as can be seen in Figure 9.

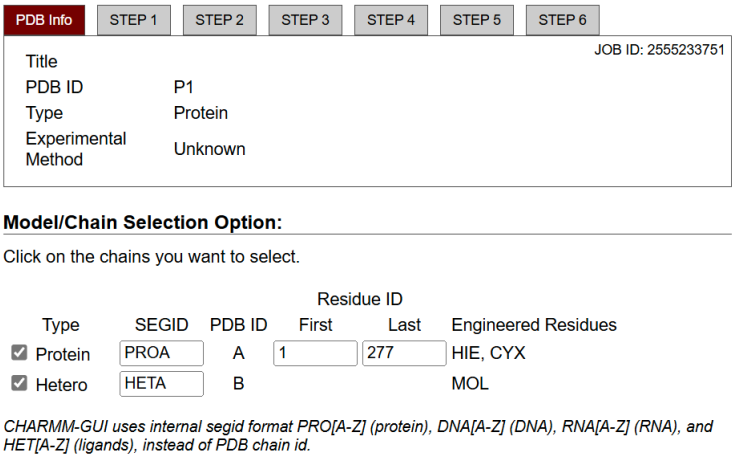


Figure 9

- To proceed with the next step, you need to include the file lig.mol2, which can be generated using the ANTECHAMBER module of AmberTools. For the purpose of this tutorial, the file is provided in the "docking" folder. This lig.mol2 file is crucial for CHARMM-GUI, as it

helps in assigning the appropriate atom types. After uploading the file, you must choose the correct terminal group patching for the system; in this case, you will select ACE and CT2. Once completed, you can move forward with the next configuration steps.

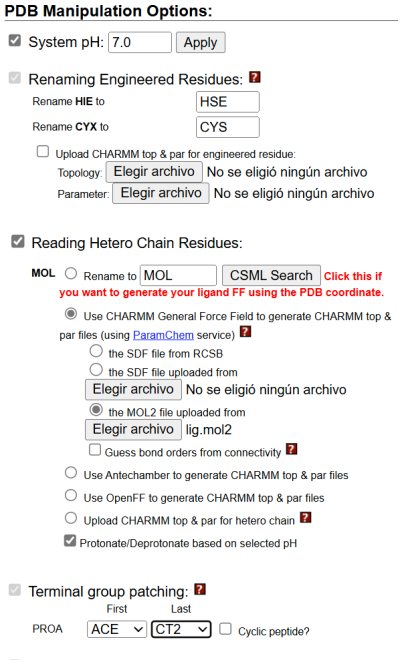


Figure 10

- After of this you have to choose PPM 2.0 for the orientation of the protein

- In this step, the protein-ligand complex is placed inside a defined space with a specific orientation, which can also be visualized for confirmation. Next, you'll select the type of membrane to incorporate into the system; in this case, the POPC membrane is chosen. As shown in Figure 13, the hydration level is set to 40, and the membrane composition consists of a 50/50 ratio of POPC. This configuration allows for proper membrane-protein interaction during the subsequent molecular dynamics simulations.

System Size Determination Options:

○ **Homogeneous Lipid** "Homogeneous Lipid" option is no longer supported. You can use "Heterogeneous Lipid" option even for homogeneous lipid bilayer building.
◉ **Heterogeneous Lipid**

1. Box Type: Rectangular (Currently, only CHARMM, NAMD, and GROMACS support the hexagonal box)

2. Length of Z based on:
○ Water thickness 22.5 (Minimum water height on top and bottom of the system)
◉ Hydration number 40 (Number of water molecules per one lipid molecule)

3. Length of XY based on:
○ Ratios of lipid components
◉ Numbers of lipid components

XY Dimension Ratio: 1
(The system size along the X and Y must be the same)

Show the system info click this once you fill the following table:

Lipid Type	Charge [e]	Tail Info. [sn1/sn2]	Images	# of Lipid on Upperleaflet	# of Lipid on Lowerleaflet	Surface Area
▸ **Sterols**						
▸ **PA (phosphatidic acid) Lipids**						
▾ **PC (phosphatidylcholine) Lipids**						
DDPC	0	10:0 / 10:0	[Image]	0	0	64.1
DCPC	0	11:0 / 11:0	[Image]	0	0	64.1
DLPC	0	12:0 / 12:0	[Image]	0	0	64.1
DMPC	0	14:0 / 14:0	[Image]	0	0	64.1
DPPC	0	16:0 / 16:0	[Image]	0	0	63.0
DSPC	0	18:0 / 18:0	[Image]	0	0	65.6
PSPC	0	16:0 / 18:0	[Image]	0	0	68.3
PYPC	0	16:0 / 16:1	[Image]	0	0	64.1
POPC	0	16:0 / 18:1	[Image]	50	50	68.3
PLPC	0	16:0 / 18:2	[Image]	0	0	65.2
PiLPC	0	16:0 / 18:2	[Image]	0	0	60.0
PEPC	0	16:0 / 22:1	[Image]	0	0	66.5
SOPC	0	18:0 / 18:1	[Image]	0	0	66.2
SLPC	0	18:0 / 18:2	[Image]	0	0	65.2
DRPC	0	14:1 / 14:1	[Image]	0	0	67.0
DYPC	0	16:1 / 16:1	[Image]	0	0	67.0
YOPC	0	16:1 / 18:1	[Image]	0	0	67.0
DOPC	0	18:1 / 18:1	[Image]	0	0	69.7
OEPC	0	18:1 / 22:1	[Image]	0	0	66.5
DUPC	0	18:2 / 18:2	[Image]	0	0	70.0
DGPC	0	20:1 / 20:1	[Image]	0	0	66.7

Figure 11

- Also in this section you can calculate the area of the lipids that cover the protein and how many of them there are.

Calculated XY System Size:

	Upperleaflet	Lowerleaflet
Protein Area	1194.48632	1194.30971
Lipid Area	3415	3415
# of Lipids	50	50
Total Area	4609.48632	4609.30971
Protein X Extent	28.76	
Protein Y Extent	25.07	
Average Area	4609.40	
A	67.89	
B	67.89	

Figure 12

- After that continues to put the concentration of the salt, which is NaCl at a 0.15 ML^{-1} concentration. You can calculate the number of ions to put and in this case is 7 and 15 for Na^{+} and Cl^{-} respectively. And finally you have to choose the type of force field to your system, that in this case are gonna the predetermined by Amber. You have to wait around

20 minutes because this is the heavier calculation.

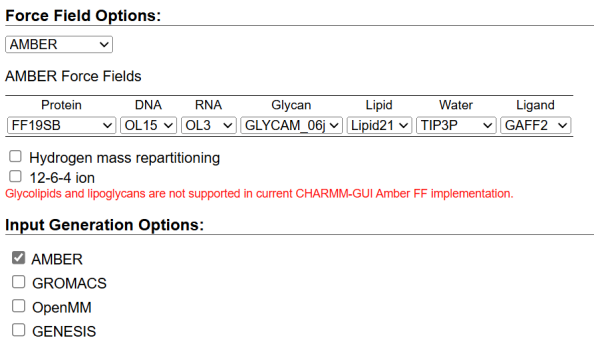


Figure 13

With all this you have put the system protein-ligand into the lipid bilayer membrane. and your system suppose to see in the next way:

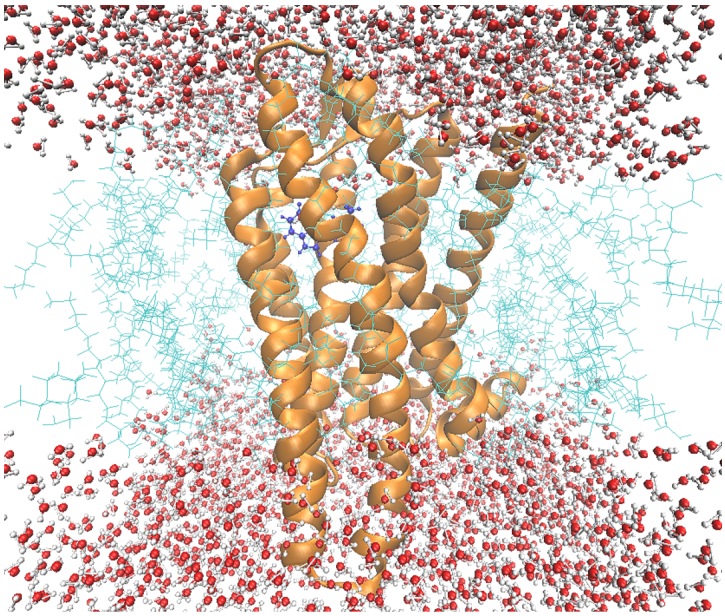


Figure 14

where the cyan are the membrane, in cpk the molecules of water, in orange the protein and in purple the ligand.

V. Molecular dynamics

Since AmberTools is already installed on your system and you have prepared your protein-ligand system, the steps for conducting the molecular dynamics (MD) simulation include minimization, heating, and the production run. To begin, navigate to the "min" folder where you will find the min.in file.

```
Minimization input file in explicit solvent
 &cntrl
    ! Minimization options
    imin=1,         ! Turn on minimization
    maxcyc=10000,   ! Maximum number of minimization cycles
    ncyc=5000,     ! 100 steepest-descent steps, better for strained systems

    ! Potential energy function options
    cut=12.0,       ! nonbonded cutoff, in angstroms

    ! Control how often information is printed to the output file
    ntpr=100,      ! Print energies every 100 steps
    ntxo=2,        ! Write NetCDF format

    ! Restraint options
    ntr=1,
    restraint_wt=10.0,
    restraintmask=':1-580',

    ! Set water atom/residue names for SETTLE recognition
    watnam='WAT',  ! Water residues are named WAT
    owtnm='O',     ! Water oxygens are named O
 /
```

Figure 15

The purpose of minimization is to locate a local energy minimum of the starting structure to prevent the MD simulation from becoming unstable. In this process, the maxcyc parameter specifies the total number of minimization steps (1000), with the algorithm switching from steepest descent to conjugate gradient during the final 500 steps. To perform the simulation, use the sander package (Simulated Annealing with NMR-Derived Energy Restraints) from Amber, which facilitates both energy minimization and molecular dynamics. Execute the simulation as follows:

```
sander -O -i min.in -p system.parm7 -c system.rst7 -o min.out -r min.rst7 -inf min.inf
```

After the calculation is complete, review the min.out file to verify if the total energy has converged. If it has, you can proceed to the heating phase of the simulation. Although this stage is termed heating, it is often confused with equilibration; the terminology is less important than understanding its purpose. The objective is to conduct a simulation in a specific thermodynamic ensemble (such as NVE or NPT) at a defined state point (target energy, temperature, and pressure) and gather data once these conditions are met. Even when velocities are assigned according to the appropriate distribution, a thermostat may be necessary to adjust the system's temperature to achieve the correct balance between kinetic and potential energies. Therefore, it is recommended to perform a thermostatted simulation before the production run, even if the final simulation will be in the NPT ensemble. In essence, the heating phase is designed to adjust the system to the desired temperature. Please navigate to the "heat" folder and open the heat.in file.

```
A NVT simulation for common production-level simulations
 &cntrl
    imin=0,         ! No minimization
    irest=0,        ! This IS a restart of an old MD simulation
    ntx=1,          ! So our inpcrd file has velocities

    ! Temperature control
    ntt=3,          ! Langevin dynamics
    gamma_ln=1.0,   ! Friction coefficient (ps^-1)
    tempi=303.15,    ! Initial temp -- give it some small random velocities
    temp0=303.15,    ! Target temperature

    ! Potential energy control
    cut=12.0,        ! nonbonded cutoff, in angstroms

    ! SHAKE
    ntc=2,          ! Constrain bonds containing hydrogen
    ntf=2,          ! Do not calculate forces of bonds containing hydrogen

    ! MD settings
    nstlim=125000, ! 125K steps, 125 ps total
    dt=0.001,       ! time step (ps)

    ntr=1,          ! Positional restraints for proteins, sugars, ligands, and lipid head groups
    restraint_wt=10.0,
    restraintmask=':1-580',


    ! Control how often information is printed
    ntpr=1000,      ! Print energies every 1000 steps
    ntwx=5000,      ! Print coordinates every 5000 steps to the trajectory
    ntwr=10000,     ! Print a restart file every 10K steps (can be less frequent)
!   ntwv=-1,        ! Uncomment to also print velocities to trajectory
!   ntwf=-1,        ! Uncomment to also print forces to trajectory
    ntxo=2,         ! Write NetCDF format
    ioutfm=1,       ! Write NetCDF format (always do this!)

    ! Wrap coordinates when printing them to the same unit cell
    iwrap=0,

    ! Set water atom/residue names for SETTLE recognition
    watnam='WAT',  ! Water residues are named WAT
    owtnm='O',     ! Water oxygens are named O
 /
```

Figure 16

To perform the heating simulation, you will need both the topology file (system.parm7) and the coordinates of the minimized structure (min.rst7). Please ensure these two files are copied into the "heat" folder. To initiate the simulation, execute the following command:

```
sander -O -i heat.in -p system.parm7 -c min.rst7 -o heat.out -x heat.crd -r heat.rst7 -inf heat.info
```

If the previous steps have been completed successfully, you can move on to the final stage of the MD simulation: the production run. At this point, you will transition from an NVT ensemble to an NPT ensemble and begin collecting data once the system has reached equilibrium. Please navigate to the "prod" folder and open the prod.in file:

```
A NPT simulation for common production-level simulations
 &cntrl
    imin=0,         ! No minimization
    irest=1,        ! This IS a restart of an old MD simulation
    ntx=5,          ! So our inpcrd file has velocities

    ! Temperature control
    ntt=3,          ! Langevin dynamics
    gamma_ln=1.0,   ! Friction coefficient (ps^-1)
    temp0=303.15,    ! Target temperature

    ! Potential energy control
    cut=12.0,        ! nonbonded cutoff, in Angstroms

    ! SHAKE
    ntc=2,          ! Constrain bonds containing hydrogen
    ntf=2,          ! Do not calculate forces of bonds containing hydrogen

    ! MD settings
    nstlim=50000000, ! 100 ns total
    dt=0.002,     ! time step (ps)

    ! Control how often information is printed
    ntpr=10000,      ! Print energies every 1000 steps
    ntwx=25000,     ! Print coordinates every 25000 steps to the trajectory
    ntwr=25000,     ! Print a restart file every 10K steps (can be less frequent)
!   ntwv=-1,        ! Uncomment to also print velocities to trajectory
!   ntwf=-1,        ! Uncomment to also print forces to trajectory
    ntxo=2,         ! Write NetCDF format
    ioutfm=1,       ! Write NetCDF format (always do this!)

    ! Wrap coordinates when printing them to the same unit cell
    iwrap=1,

    ! Constant pressure control.
    barostat=1,     ! Berendsen barostat... change to 2 for MC
    ntp=3,          ! 1=isotropic, 2=anisotropic, 3=semi-isotropic w/ surften
    pres0=1.0,      ! Target external pressure, in bar
    taup=1.0,

    ! Constant surface tension (needed for semi-isotropic scaling). Uncomment
    ! for this feature. csurften must be nonzero if ntp=3 above
    csurften=3,     ! Interfaces in 1=yz plane, 2=xz plane, 3=xy plane
    gamma_ten=0.0, ! Surface tension (dyne/cm). 0 gives pure semi-iso scaling
    ninterface=2,  ! Number of interfaces (2 for bilayer)

    ! Set water atom/residue names for SETTLE recognition
    watnam='WAT',   ! Water residues are named WAT
    owtnm='O',      ! Water oxygens are named O
```

Figure 17

Again, copy the topology (system.parm7) and the coordinate file of the last frame of the heating run (heat.rst7) to the "prod" folder. Please execute it as:

```
sander -O -i prod.in -p system.parm7 -c heat.rst7 -o prod.out -r prod.rst7 -x prod.crd -inf heat.info
```

If the simulation proceeds as expected, you will have completed your MD simulation. In structure-based drug design, the objective is to identify a new pharmaceutical compound that effectively binds to a macromolecular receptor. The strength of this binding is quantified by the binding free energy, ΔG_{bind}. Traditionally, this process is conducted experimentally, which is both time-consuming and expensive. The Molecular Mechanics Generalized Born Surface Area (MMGBSA) method is a widely used endpoint approach for this purpose. This method estimates ΔG_{bind} based on the free energies of the reactants and products in the following equation:

$$\Delta G_{bind} = \langle G_{RL} - G_R - G_L \rangle \tag{1}$$

The free energy G of the complex, receptor, and ligand is estimated by:

$$G = G_{bond} + G_{elec} + G_{vdW} + G_{pol} + G_{np} - TS \tag{2}$$

In this equation, the first three terms represent the standard MM force energy components: G_{bond}(which includes bond, angle, and dihedral terms), G_{elec}(electrostatic interactions), and G_{vdw}(van der Waals interactions). G_{pol} denotes the polar contribution to the solvation free energy, while G_{np} represents the non-polar contribution to the solvation free energy. The final term is the product of the absolute temperature T and the entropy S. Although entropy can be estimated through normal mode analysis of vibrational frequencies from conformational snapshots obtained during the MD simulations, this term is often neglected due to its significant computational cost. It is generally assumed that binding to different pockets and the relative binding free energy of similar ligands exhibit comparable entropy variations. In the 1A-MMGBSA approach, G_{bond} is set to zero for a single-trajectory calculation. The equation simplifies to:

$$G = G_{elec} + G_{vdW} + G_{pol} + G_{np} - TS \tag{3}$$

The electrostatic term, G_{elec}, is calculated using Coulomb's law with atomic charges derived from the MM force field. Research has demonstrated that using electrostatic potential (ESP) charges from advanced quantum mechanical calculations does not significantly enhance the results. However, it is essential that the ligand's functional groups are well-parametrized by the force field to ensure accurate atomic charges. The polar solvation term, G_{pol}, accounts for the electrostatic interactions between the solute and the continuum solvent, and is computed using the Generalized Born implicit solvent model. The non-polar solvation energy, G_{np}, encompasses cavitation, dispersion, and repulsion energies between the solute and solvent that are not captured by G_{pol}. This term is estimated using a linear relationship with the solvent-accessible surface area (SASA). The explicit solvent from the MD simulation is replaced with the GBSA implicit solvent model to estimate the solvation energy.

Please exit the "prod/" folder and enter the "mmgbsa/" folder. Copy the topology file (system.parm7) and the newly generated trajectory file (100f.nc) into this folder. The MMGBSA program requires the generation of topology files for the complex, receptor, and ligand. Fortunately, a Python script is available to automate this process, and it is included in the ante-mmgbsa.in file.

```
#!/bin/bash

ante-MMPBSA.py -p system.parm7 -c complex.parm7 -r receptor.parm7 -l lig.parm7 -s :WAT,Cl-,Na+ -n :MOL --radii mbondi2
```

Execute the provided script to proceed. A key feature of MMGBSA is its capability to decompose the binding free energy into pairwise residue-ligand binding energies. To achieve this, we must first identify the residues surrounding the ligand and use this information as input. Utilizing our cpptraj skills, we can apply the nativecontacts function to list residues within a specified radius of the ligand. Please open the contact.in file to continue.

```
parm system.parm7
trajin new.crd
nativecontacts :MOL&!@H= @1-8486&!@H= byresidue resout residue.out distance 7.0 savenonnative
go
exit
```

You execute it using cpptraj -i contact.in. The output file we're interested in is the residue.out which shows us the RESID of all the residues in contact with out ligand 280. Now let's open the mmgbsa.in file, which will be the input file for our free energy calculation.

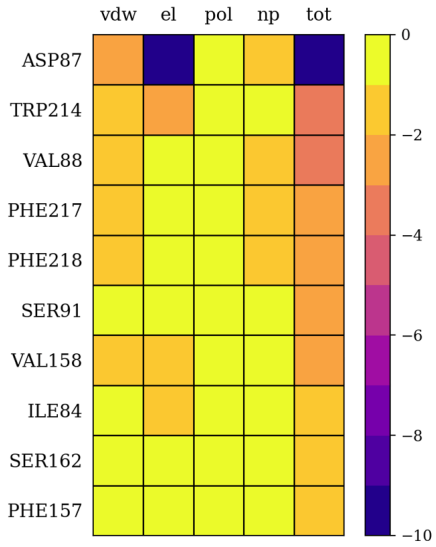
```
Input file for running PB and GB
&general
  startframe = 1,
  interval = 200,
  endframe = 10100,
  verbose = 1,
  ligand_mask = "280",
/
&gb
  igb = 5,
  saltcon = 0.150 ,
/
&decomp
  idecomp=3,
print_res=
  dec_verbose=0,
/
```

In the general section, you need to define the initial and final frames, the interval for reading frames, and the RESID of the ligand. The gb section specifies the solvation model to be used, while the decomp section is where you list the residues for energy decomposition. You'll notice that the "print_ res" field is currently empty; this is where you should input the relevant residues. You can either add them manually or use the Python script provided in this tutorial for automation. Using the command python res.py. If you open again the mmgbsa.in file, the script has printed in a line the first 10 residues taken from the residue.out file, plus at the end also the RESID of our ligand, 280. With all the ingredients ready, we can first open the MMGBSA.in file,

```
#!/bin/bash

MMPBSA.py -O -i mmgbsa.in -o FINAL.dat -do FINAL_DECOMP.dat -sp system.parm7 -cp complex.parm7 -rp receptor.parm7 -lp lig.parm7 -y new.crd &
```

After running the script, the results will be saved in two files: FINAL.dat, which contains information on the total binding free energy, and FINAL_ DECOMP.dat, which provides the residue-ligand energy decomposition. This calculation typically takes around 6-7 minutes, making it a good time for any questions. Once completed, at the end of the FINAL.dat file, you'll find the total binding free energy (ΔG_{bind}) decomposed into (ΔG_{vdw}), (ΔG_{elec}), (ΔG_{pol}) and (ΔG_{np}). Since the FINAL_ DECOMP.dat output may appear cluttered, use the provided lig.sh script to extract only the relevant lines. This small bash script will generate a file called lig.dat, that contains the ligand-residue energy decomposition of the terms previously described. Use this Python script to generate a heat map of the interaction, using **python get-matrix_new.py lig.dat**. You get the heat map:



VI. Quantum part

Following this step, the remaining task is to perform the Energy Decomposition Analysis (EDA). At this point, the focus shifts to the application of two advanced techniques: QM/MM and EDA.

QM/MM is a hybrid computational method that combines Quantum Mechanics (QM) for accurate modeling of a small, crucial region of the system, such as a reacting molecule, with Molecular Mechanics (MM) for efficiently simulating the larger surrounding environment, such as solvent molecules or protein residues. The key aspects of this method are as follows:

- QM Region: Provides high accuracy and is used for regions where electronic effects, such as bond formation or breaking, are critical.
- MM Region: Uses a simplified classical model and is applied to the environment surrounding the QM region, such as solvent molecules or protein structures.

Energy Decomposition Analysis (EDA) is employed to dissect the total interaction energy between molecular fragments into distinct components. This approach aids in understanding the individual forces that contribute to bonding and interactions within the system. Typically, the interaction energy ΔE_{int} is divided into the following components:

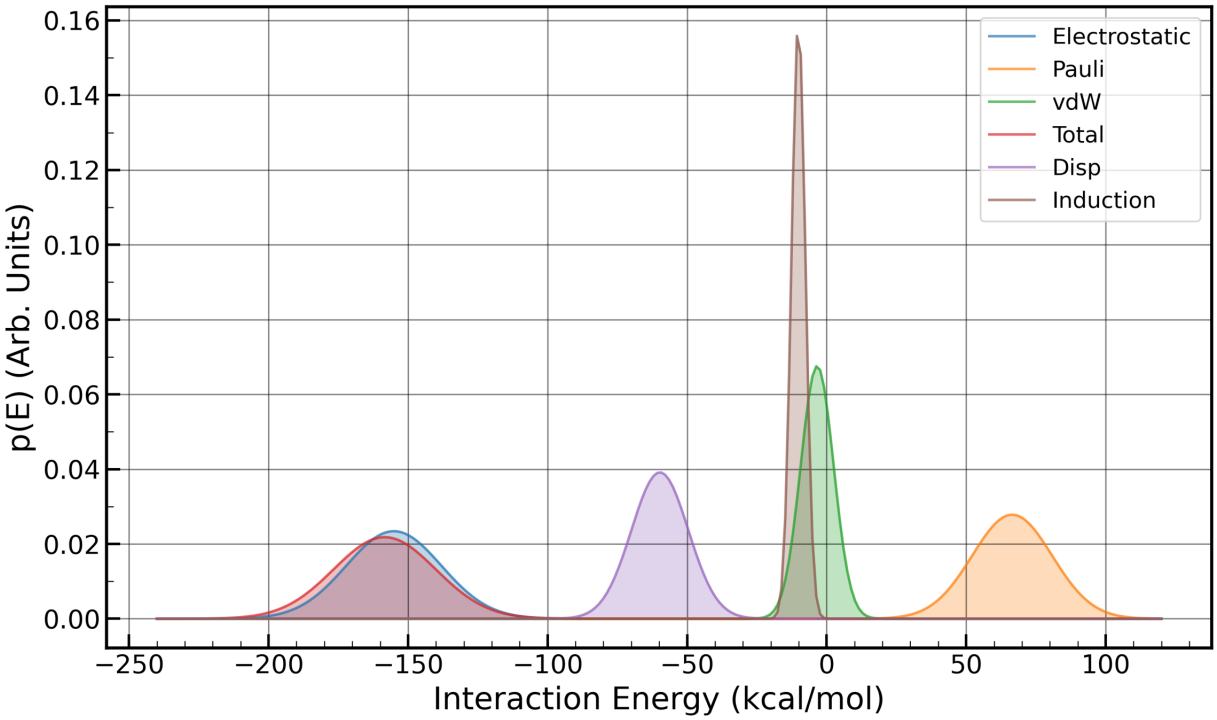
- Electrostatic Energy: Describes the attraction or repulsion between charge distributions.
- Pauli Repulsion: Represents the energy penalty associated with overlapping electron densities, as dictated by the Pauli exclusion principle.
- Orbital (or Covalent) Interaction: Accounts for stabilization through the mixing of molecular orbitals, including charge transfer and polarization effects.
- Dispersion: Refers to Van der Waals forces arising from correlated electron motions.

EDA provides valuable insights into molecular interactions, reaction mechanisms, and bonding in a rigorous and physically meaningful manner.

In the context of using QM/MM techniques, defining the QM region is critical. The QM region is the area where the chemical reaction occurs, and it is typically based on the residues interacting with the ligand. Using the qmmask in the main input file, you can select specific residues for this region. In this case, the QM region is determined by results from an earlier MMGBSA analysis, focusing on three key residues.

```
&main
tpl = gaussian
top = system.parm7
traj = 100f.nc
qmmask = @4481-4513, 1405-1410, 1472-1476
geoms = 1 100 1
&end
```

For simplicity, the trajectory file from the previous MD simulation is selected, and the EDA.sh script is run to conduct the energy decomposition analysis. Finally, the getEDA_energies.py and yesvdw.py scripts are executed to extract and analyze the data, generating the decomposition analysis.



References

[1] Kuglae Kim, Tao Che, Ouliana Panova, Jeffrey F DiBerto, Jiankun Lyu, Brian E Krumm, Daniel Wacker, Michael J Robertson, Alpay B Seven, David E Nichols, et al. Structure of a hallucinogen-activated gq-coupled 5-ht2a serotonin receptor. *Cell*, 182(6):1574–1588, 2020.

[2] Vito F Palmisano, Carlos Gómez-Rodellar, Hannah Pollak, Gustavo Cárdenas, Ben Corry, Shirin Faraji, and Juan J Nogueira. Binding of azobenzene and p-diaminoazobenzene to the human

voltage-gated sodium channel na v 1.4. *Physical Chemistry Chemical Physics*, 23(5):3552–3564, 2021.