

CHAPTER 2

Methodology

2.1 Homology Modeling

2.1.1 Data Searching and Sequences Alignment

The target sequence of human tyrosinase (accession number AAA61242) was retrieve from the National Center for Biotechnology Information (NCBI, <http://www.ncbi.nlm.nih.gov/>) [1] protein sequence database. The BLAST (Basic Local Alignment Search Tool) [2] search was use to identified a homolog protein template of the query protein human tyrosinase, these tool is probably the most widely used tool/algorithm in bioinformatics. BLAST is a sequence alignment algorithm; it designed to test one target sequence against a database of sequences in an attempt to find common regions. In addition, the regions of similarity between target sequence being test and homolog protein sequence from the database can be identified by finding High Scoring Segment Pair (HSSP). Each set of HSSP are assigned a score by summing the similar value from an appropriate substitution matrix [3-4].

Substitution matrices are a key element of a pairwise sequence alignment for evaluating the quality that assigns a score for aligning any possible pair of residues. The scoring system made use of a simple match/mismatch scheme, but we can increase sensitivity to weak alignments through the use of a substitution matrix in the comparing of proteins. It is well known that certain amino acids can substitute easily for one another in related proteins, presumably because of their similar physic-chemical properties. For examples, the conservative substitutions include isoleucine for valine (both small and hydrophobic) and serine for threonine (both polar). When calculating alignment scores, identical amino acids should be given greater value than substitutions, but conservative substitutions should also be greater than non-conservative changes. The default matrices currently used by many pairwise alignment systems are the BLOSUM series; it produced from the alignment of a series of MobaXterm protein

blocks. The PAM series were the original substitution scoring matrices and were produced from the alignments of more closely related sequences than those used for BLOSUM. For general database searches, the BLOSUM matrices are often more sensitive [4-6].

BLOSUM 62 SUBSTITUTION MATRIX																				
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Figure 2.1 The block of amino acid substitution matrix of BLOSUM 62

PAM-250 SUBSTITUTION MATRIX																				
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	2																			
Arg	-2	6																		
Asn	0	0	2																	
Asp	0	-1	2	4																
Cys	-2	-4	-4	-5	12															
Gln	0	1	1	2	-5	4														
Glu	0	-1	1	3	-5	2	4													
Gly	1	-3	0	1	-3	-1	0	5												
His	-1	2	2	1	-3	3	1	-2	6											
Ile	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
Leu	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
Lys	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
Met	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
Phe	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
Pro	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
Ser	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
Thr	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
Trp	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Tyr	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
Val	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Figure 2.2 The block of amino acid substitution matrix of PAM-250

The protein template, tyrosinase of *B.megaterium* (3NQ1) with resolution at 2.3 Å, was selected according to the highest percent identity of 33.5 at the time. The structure of bacterial tyrosinase was determined [7]. The selected template and protein sequence were aligned to build a 3D structure by pairwise sequence alignment methods using Discovery Studio 2.5 software package [8-9]. The pairwise sequence alignment methods are used to find the best-matching piecewise (local) or global alignments of two query protein sequences. Pairwise alignments only can be used between two sequences at a time, but they are efficient to calculate and are often to use for methods that do not require extreme precision such as searching a database for sequences with high similarity to a query. The three primary methods of producing pairwise alignments are dot-matrix methods, dynamic programming, and word methods [10].

A local alignment without gaps contain simply of a pair of equal length segment, only one from each of two sequences to be compared. The high-scoring segment pairs (HSPs) are the modification of the Smith-Waterman or Sellers algorithms will find all

segment pairs which score cannot be improved by extension or trimming. For protein, the simple model was chosen the amino acid residues in a sequence independently with specific background probabilities for the various residues. Moreover, the expected score for aligning a random pair of amino acid is required to be negative. The statistics of HSP scores are characterized by two parameters, K and λ . Most simply, the expected number of HSPs with score at least S is given by the formula.

$$E = Kmn e^{-\lambda S} \quad (2.1)$$

We call this the E -value for the score S . When m is the length of the target sequence and n is the size of the database, and of course the s score. The parameters K and λ represent natural scale for the search space and the scoring system, respectively [11-15].

2.1.2 Homology model and structure refinement

The 3D structure was built by Build Homology Model protocol under the Protein Modeling protocol group in Discovery Studio 2.5 program packet. The template of 3D structure of human tyrosinase was downloaded from Protein Data Bank (PDB), unit cell and unnecessary atoms (water, ligand and ion) were removed as shown in **Fig. 2.3**.

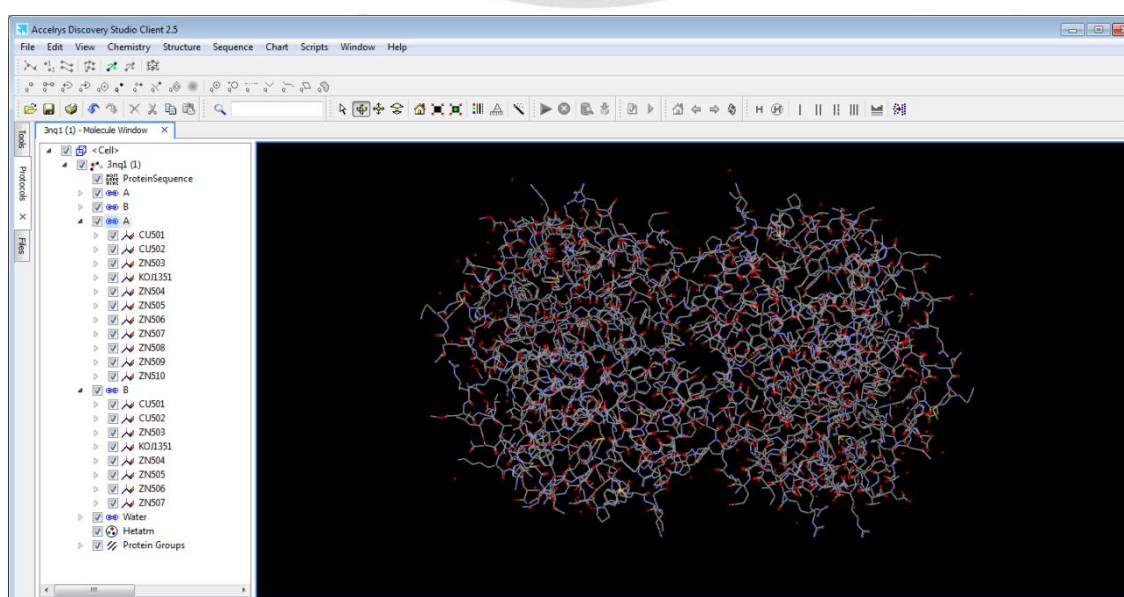


Figure 2.3 The crystal structure of template (PDB id : 3NQ1)

The target sequence was aligned with template structure from bacterial using Aling3D (Fig.2.4).

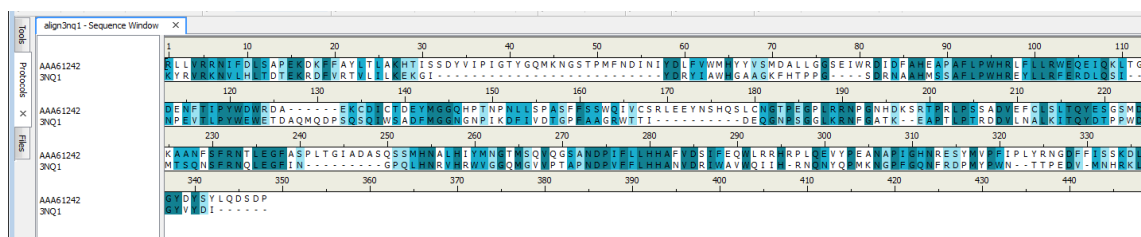


Figure 2.4 Alignment sequences of model and template

To build the 3D structure model was using Build Homology Model under the Protein Modeling protocol group. the Sequence Alignment, Model Sequence, and Template Structure were input file to built model in the Parameter Explorer, and set the Optimization Level for Model parameter to Low.

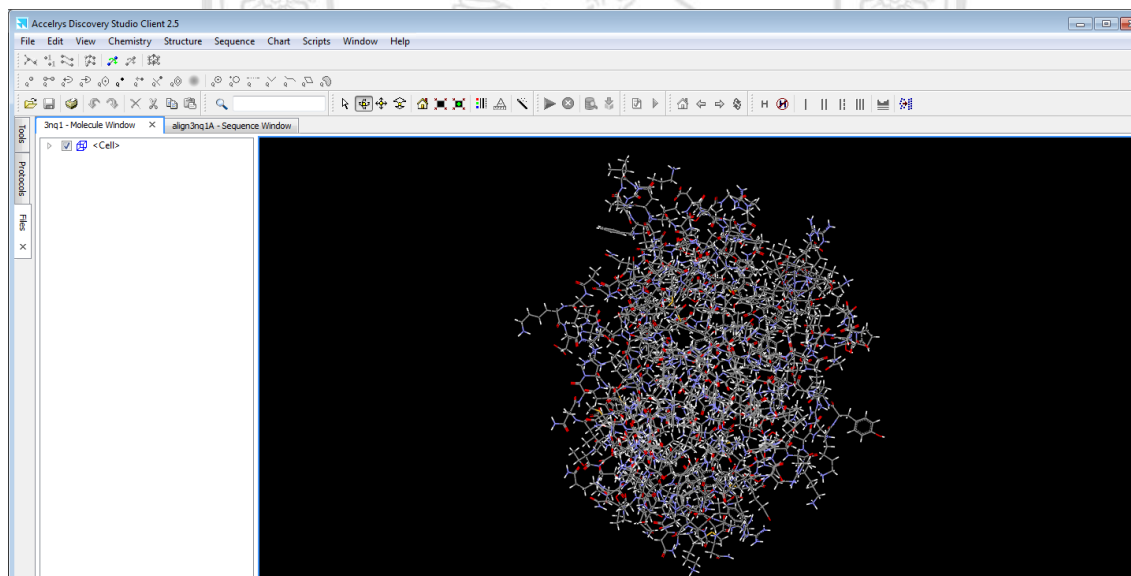


Figure 2.5 Input file in build homology model step

The homology model was refined by Loop Refinement [16] protocol base on CHARMM force-field; the conformation of a contiguous segment such as a loop of a protein structure was optimized. The Loop Refinement is based on systematic conformation sampling of the loop backbone and CHARMM energy minimization, that can be used to refine a loop structure from homology model as well as to optimize a

segment of the protein experimental structure where the structure is poorly defined. Next, the energy of the homology model was minimized by the Minimization protocol through geometry optimization using CHARMM force-field. Minimization procedure was use with the steepest descent method for 1000 step.

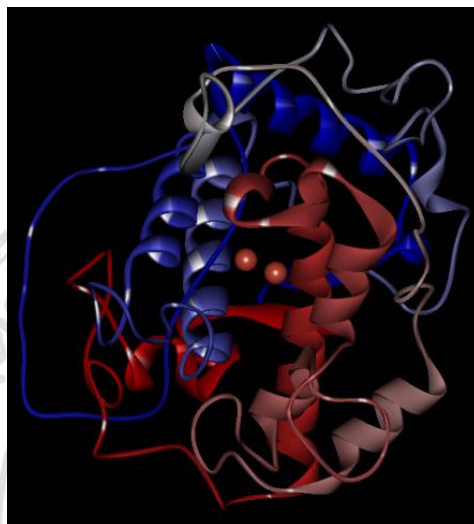


Figure 2.6 Homology model after loop refinement and minimization

Energy minimization was performed on structure prior to dynamic to relax the conformation and remove steric overlap that lead to bad contacts. In the absence of an experimental structure, a minimized ideal geometry can be used as starting point. The function optimization is a calculation that pervades much of numerical analysis. Additionally, the function to be optimize (minimize) is energy in the context of macromolecule. The energy landscape of a biomolecule possesses an enormous number of minima or conformational substrate, which the goal of energy minimization is simply to find the local energy minimum. The energy at this local minimum may be much higher than the energy of the global minimum. Moreover, energy minimization corresponds to an instantaneous freezing of the system; a static structure in which no atom receives a net force corresponds to a temperature of 0 K [17].

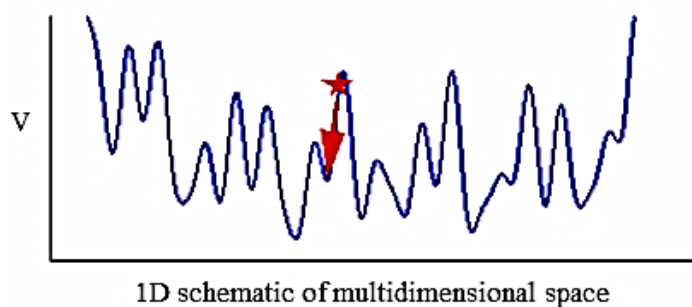


Figure 2.7 Energy minimization seeks the energy minimum nearest the starting (marked by star) conformation [18]

Steepest descent minimizes (SD) is one of the simplest and best known methods for minimizing a function. It only use first derivative information and save only the current location of the coordinates from iteration to iteration. The energy can be calculated for the initial geometry and then when one of the atoms has been moved in a small increment in one of the directions of the coordinated system. These processes will stop if the predetermined minimum condition is fulfilled. Basically, SD converges very slowly to a local minimum in a complex potential energy surface. This method is very useful for poorly refine crystallographic data or to relax graphically built model [17].

2.1.3 Evaluation of model

The structure was optimizes and checked with PROCHECK [19] and Verify3D [20-21]. Energy criteria in comparison with the potential of mean force derived from a large set of known protein structures was determined.

PROCHECK is a program based on an analysis of phi and psi angles, peptide bond planarity, bond angles, bond lengths, hydrogen bond geometry, and side chain conformation of known protein structure as a function of atomic resolution. The PROCHECK analysis was providing an idea of the stereochemical quality of all protein chains in a given PDB structure. They highlight regions of the proteins which appear to have unusual geometry and provide an overall assessment of the structure as a whole, the results was shown on ramachandran plot.

Ramachandran plot was used to check the secondary structure of protein. In a polypeptide the main chain N-Calpha and Calpha-C bonds relatively are free to rotate. These rotations are represented by the torsion angles phi and psi, respectively. The limitations impose on the primary structure of a protein by peptide bond and hydrogen bond consideration dictate the secondary structure that possible, Ramachandran and co-workers introduce backbone dihedral angles psi (ψ) against phi (ϕ) of amino acid residues in protein structure in 1963. They construct a steric map of ramachandran plot to predict the commonly allowed regions: The red regions correspond to conformations where there are no steric clashes; these are the allowed regions namely the alpha-helical and beta-sheet conformations. The yellow areas show the allowed regions if slightly shorter *van der Waals* radii are used in the calculation, the atoms are allowed to come a little closer together. This brings out an additional region which corresponds to the left-handed alpha-helix. Disallowed is the white areas, it correspond to conformations where atoms in the polypeptide come closer than the sum of their *van der Waals* radii. These regions are sterically disallowed for all amino acids except glycine which is unique in that it lacks a side chain.

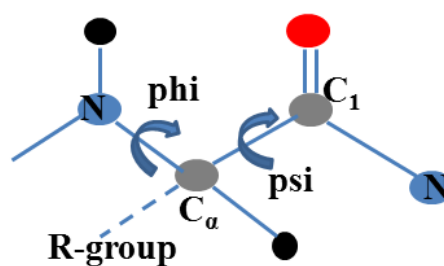
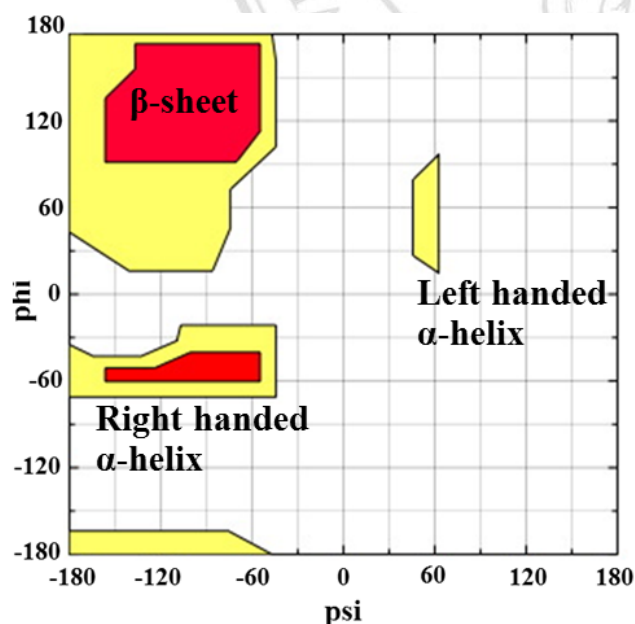


Figure 2.8 Ramachandran plot

Verify3D was used to measure the compatibility of amino acid sequence with three-dimension protein structure, this method reducing the 3D structure to simplified 1D representation called an environment string, which is then compared with the 1D amino acid sequence. The method can also be used to check the validity of a hypothetical protein structure by measuring the compatibility of that structure with the sequence of protein.

The method involves three basic operations:

- i Reduction of the 3D structure to a 1D string of residue environments. From these environments are categorized according to the area of the side chain, which is buried in the protein.
- ii Generation of a position-dependent comparison matrix was known as the 3D profile. These calculated from the string of residue environments, a precalculated scoring matrix and heuristic gap penalties. The scoring matrix was calculated from the probabilities of finding each of the twenty amino acids in each of the environment classes. The gap penalties were varied with position in the profile.
- iii Alignment of a sequence with the 3D profile. The resulting alignment score is a measure of the compatibility of the sequence with the structure described by the 3D profile.

These sections explain the details of these operations and the various ways in which the results can be used [20-21].

The Ramachandran plot was calculated from PROCHECK interactive server (<http://services.mbi.ucla.edu/PROCHECK/>). The first upload PDB file of homology model on web and then fill answer in bank and press 'Run PROCHECK', wait until webpage show the result.

PROCHECK interactive server

For more information on procheck, and where to get the source files: [visit this site](#)

[SAVES](#) | [XdVal](#) | [MTZdump](#) | [Ramachandran Plot](#) | [pdbU](#) | [pdsNAFU \(Check for ADIT compliance\)](#) | [\[PROCHECK\]](#) | [Verify3D](#) | [ERRAT](#)

Upload your PDB file refine1.pdb

Please add these two numbers: 7 + 10 = [What's this?](#)

Molecular Biology Institute at the University of California, Los Angeles.

Figure 2.9 PROCHECK webpage

Verify3D was run on web server (http://services.mbi.ucla.edu/Verify_3D/) as the same with Ramachandran plot. The first upload PDB file of homology model on web and then fill answer in bank and press ‘Run Verify 3D’, wait until webpage show the result.

[SAVES](#) | [XdVal](#) | [MTZdump](#) | [Ramachandran Plot](#) | [pdbU](#) | [pdsNAFU \(Check for ADIT compliance\)](#) | [PROCHECK](#) | [Verify3D](#) | [ERRAT](#)

Determines the compatibility of an atomic model (3D) with its own amino acid sequence (1D) by assigning a structural class based on its location and environment (alpha, beta, loop, polar, nonpolar etc) and comparing the results to good structures.

References:

- [Bowie, et al., 1991](#)
- [Luethy, et al., 1992](#)

Upload a file: refine1.pdb

Please add these two numbers: 3 + 9 = [What's this?](#)

Figure 2.10 Verify3D webpage

2.2 Molecular docking

To analyze binding scaffold of substrates and inhibitors with tyrosinase, molecular docking was carried out using AutoDock4.0 software for prediction of binding structure tyrosinases with inhibitors (ascorbic acid, arbutin, kojic acid, and tropolone).

AutoDock was first published in 1990, this software was provided to be an effective tool capable of quickly and accurately predicting bound conformations and binding energies of ligands with macromolecular targets. In order to allow searching of the large conformational space available to a ligand around a protein, this software uses a grid-based method to allow rapid evaluation of the binding energy of trial conformations. The target protein is embedded in a grid in this method, and then, a probe atom is sequentially placed at each grid point, the interaction energy between the target and the probe is computed. This grid of energies may be used as a lookup table during the docking simulation [22-27].

AutoDock software employs a semiempirical force-field based on a comprehensive thermodynamic model which allows incorporation of intramolecular energies into the predicted free energy of binding and a Lamarckian genetic algorithm (LGA) for the conformational search. Lamarckian genetic algorithm is the method for conformational search, a population of trial conformations is created, and then in successive generations these individuals mutate, exchange conformational parameters, and compete in a manner analogous to biological evolution, ultimately selecting individuals with lowest binding energy. This method aspect is an added feature that allows individual conformations to search their local conformational space, finding local minima, and pass this information to later generations [22,24].

Tyrosinase from three different sources was set as a receptor. The excess water molecules were isolated and deleted from the structure. All hydrogen atoms were added to the protein structure. These changes were then saved, and converted PDB file to PDBQT file.

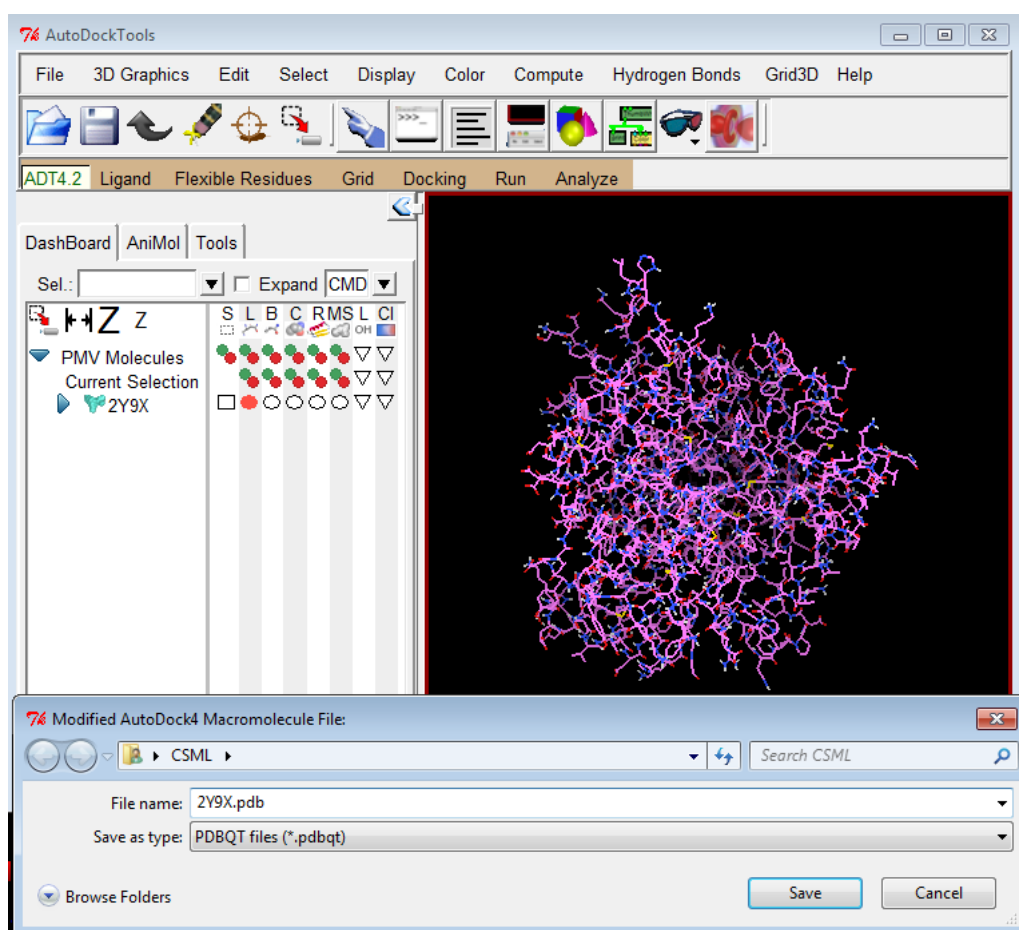


Figure 2.11 The PDB file of protein

The same as receptor file, the ligand file was opened, charges were added and all non-polar hydrogen atoms were merged. Next, bonds within the ligand were set as rotatable. After the root atom of the ligand was detected and all torsions were selected and set, the file was then saved as a PDBQT file type.

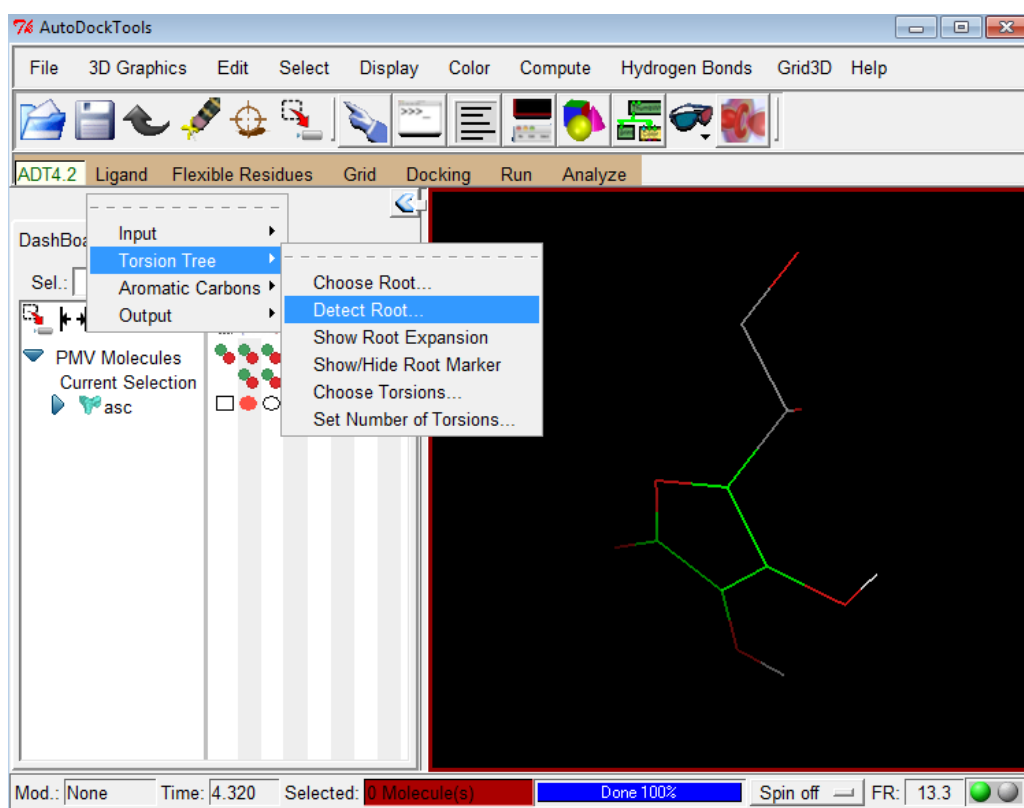


Figure 2.12 The PDB file of ligand

The grid maps was constructed using the AutoGrid function. Both receptor and ligand files were chosen for the mapping. A grid box was then used to selected where area of the protein structure to be mapped. Generally this grid box is located at the active site. The size of grid used in this work was set to be 60 Å × 60 Å × 60 Å in the x, y, and z axis, respectively (Fig. 2.11).

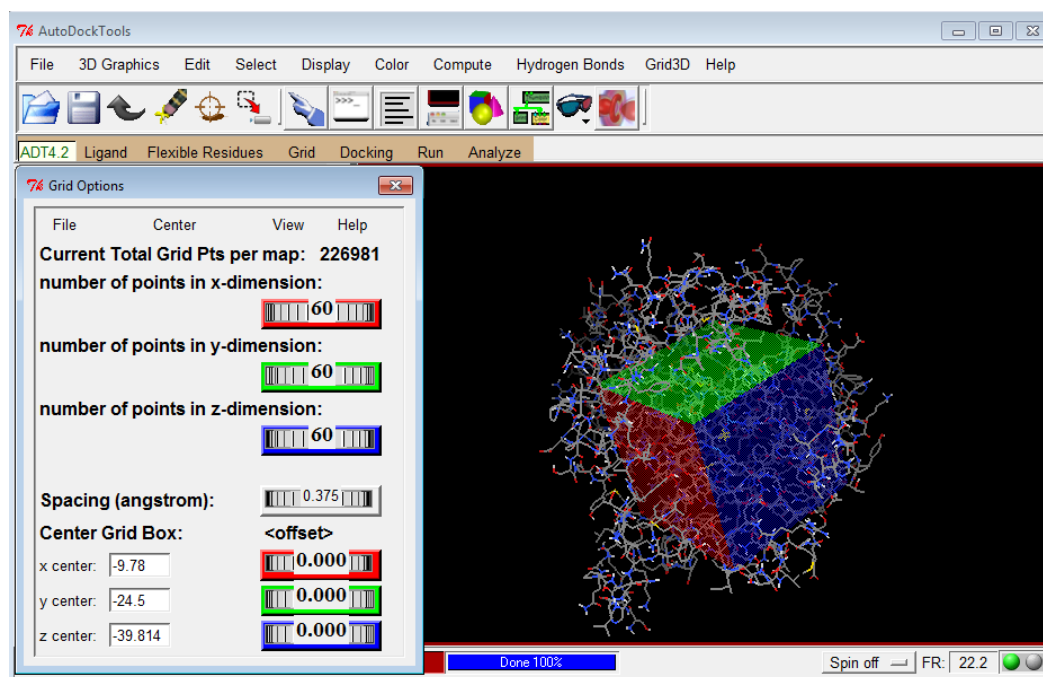


Figure 2.13 Preparation GPF file

To run the program, cywin terminal was open and command line ‘autogrid4 -p my_work.gpf -l my_work.glg &’ was used.

```

CSML@CSML1 ~
$ autogrid4 -p my_work.gpf -l my_work.glg &

```

Figure 2.14 Command lines for run autogrid4

The rigid protein and ligand files were selected. The Lamarkian genetic algorithm was used. A population size of 150 conformations and maximum number of energy evaluations of 2.5 million were applied. A docking file was created as DPF file. The resulting docking conformations were returned in the DLG file.

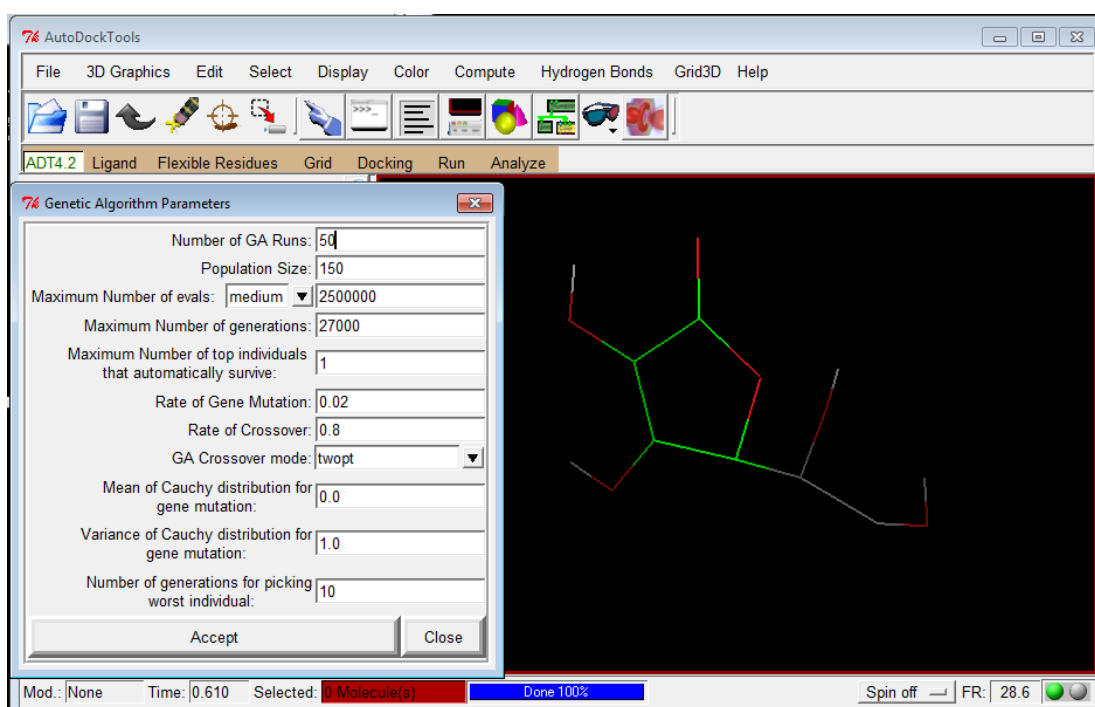


Figure 2.15 Preparation DPF file

At this step, command line 'autodock4 -p my_work.dpf -l my_work.dlg &' was used for run (Fig. 2.13-2.14).

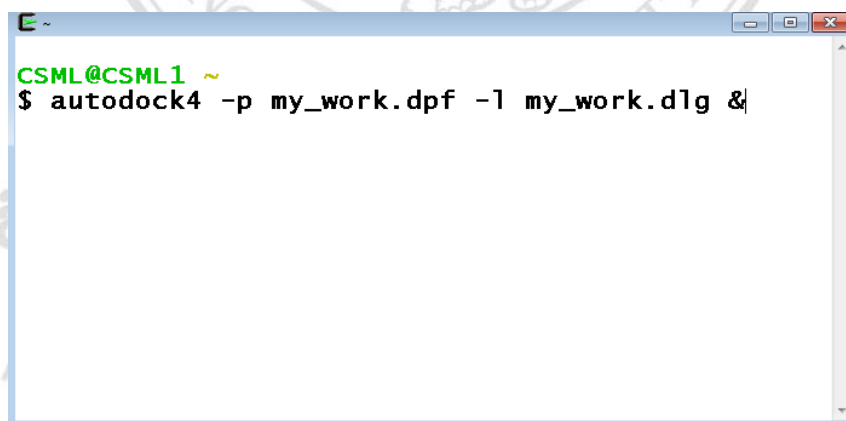


Figure 2.16 Command lines for run autodock4

The RMSD is mostly used in structural studies to determine how similar two biomolecule structures are such as proteins, small ligand molecules. One has to

differentiate between a) using the RMSD to measure the similarity of two structures as ($\text{RMSD}_{\text{Protein}}$) and b) to determine the quality of a ligand placement after docking into a binding site of protein ($\text{RMSD}_{\text{Ligand}}$). The RMSD is used as a measurement of how close the docked solution is to a given “real” ligand placement (docking accuracy) is in the latter case. The RMSD values was calculated using by Pymol [31] and was given in all-atom RMSD if not indicated otherwise (another common measurement is C-alpha or $\text{C}\alpha$ RMSD where only the $\text{C}\alpha$ atoms of the protein chain are considered for the calculations). This is between two structures X and Y containing of N atoms on each calculated as the square root of the average of the squared distances between the corresponding atoms of structure X and structure Y:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=0}^N |x_i - y_i|^2} \quad (2.2)$$

The typical values for $\text{RMSD}_{\text{Ligand}}$ that are considered as in good agreement, such as with experimental ligand configurations are RMSDs of $\leq 2^\circ\text{A}$. This boundary was used throughout to distinguish between “good” and “poor” ligand placements after docking [28].

From docking result file, the structures were clustered determined by Root Mean Square Deviation (RMSD) value of $2 \text{ \AA}$ comparing with the initial position of ligand. The conformation of the highest numbered population size cluster with the lowest energy in was selected

2.3 Molecular Dynamics Simulation

Molecular dynamics (MD) simulation was performed using AMBER 12 program [29]. The protein-ligand complex from docking was set as an initial structure. This structure was lack hydrogen atoms because they cannot to be resolved properly. Thus these have to be added to the file.

csml@csml-01: ~/Job/Bee/nofix2arb

ATOM	1	N	SER	2	-31.005	-0.748	-33.326
ATOM	2	CA	SER	2	-29.899	-1.651	-33.659
ATOM	3	CB	SER	2	-28.672	-0.682	-33.907
ATOM	4	OG	SER	2	-28.922	0.305	-35.019
ATOM	5	C	SER	2	-29.721	-2.476	-32.396
ATOM	6	O	SER	2	-29.456	-3.687	-32.571
ATOM	7	N	ASP	3	-29.933	-1.711	-31.331
ATOM	8	CA	ASP	3	-29.924	-2.148	-30.060
ATOM	9	CB	ASP	3	-31.410	-2.245	-29.721
ATOM	10	CG	ASP	3	-31.776	-2.391	-28.331
ATOM	11	OD1	ASP	3	-30.934	-2.334	-27.418
ATOM	12	OD2	ASP	3	-33.127	-2.466	-28.071
ATOM	13	C	ASP	3	-28.965	-3.334	-29.840
ATOM	14	O	ASP	3	-27.928	-3.442	-30.541
ATOM	15	N	LYS	4	-29.045	-4.157	-28.799
ATOM	16	CA	LYS	4	-28.036	-5.166	-28.574
ATOM	17	CB	LYS	4	-28.394	-6.007	-27.254
ATOM	18	CG	LYS	4	-28.323	-5.103	-25.976
ATOM	19	CD	LYS	4	-28.319	-5.947	-24.651
ATOM	20	CE	LYS	4	-27.030	-6.806	-24.610
ATOM	21	NZ	LYS	4	-26.921	-7.499	-23.296
ATOM	22	C	LYS	4	-27.908	-6.073	-29.780
ATOM	23	O	LYS	4	-28.664	-7.054	-29.981
ATOM	24	N	LYS	5	-26.815	-6.149	-30.505
ATOM	25	CA	LYS	5	-26.673	-7.203	-31.437
ATOM	26	CB	LYS	5	-25.764	-6.794	-32.581
ATOM	27	CG	LYS	5	-26.313	-5.583	-33.440
ATOM	28	CD	LYS	5	-27.510	-6.000	-34.350
ATOM	29	CE	LYS	5	-27.854	-4.884	-35.345
ATOM	30	NZ	LYS	5	-28.839	-5.443	-36.297
ATOM	31	C	LYS	5	-26.228	-8.464	-30.673
ATOM	32	O	LYS	5	-27.131	-9.152	-30.085
ATOM	33	N	SER	6	-25.028	-9.158	-30.852
ATOM	34	CA	SER	6	-24.633	-10.543	-30.320
ATOM	35	CB	SER	6	-23.028	-10.354	-30.480
ATOM	36	OG	SER	6	-22.088	-11.479	-30.257
ATOM	37	C	SER	6	-25.062	-10.669	-28.826
ATOM	38	O	SER	6	-24.801	-9.687	-28.195
ATOM	39	N	LEU	7	-25.717	-11.684	-28.167

Figure 2.17 PDB file for prepare topology file

The input file to prepare of the topology file of protein and ligand was contain pdb file, prepin file and frcmod file can generate in Antechamber by using command line ‘antechamber -i xxx.pdb -fi pdb -o xxx.prepin -fo prepi -at amber -c gas -nc 0 &’ to prepare prepin file and ‘parmchk -i xxx.prepin -f prepi -o xxx.frcmod’ to prepare frcmod file (xxx instead file name).


```

csml@csml-01: ~/Job/Bee/nofix2arb
csml@csml-01:~/Job/Bee/nofix2arb$
csml@csml-01:~/Job/Bee/nofix2arb$ tleap
-I: Adding /home/amber12/dat/leap/leap/leaprc.ff12SB
-I: Adding /home/amber12/dat/leap/lib to search path.
-I: Adding /home/amber12/dat/leap/parm to search path.
-I: Adding /home/amber12/dat/leap/cmd to search path.

Welcome to LEaP!
(no leaprc in search path)
> source leaprc.ff12SB
----- Source: /home/amber12/dat/leap/cmd/leaprc.ff12SB
----- Source of /home/amber12/dat/leap/cmd/leaprc.ff12SB done
Log file: ./leap.log
Loading parameters: /home/amber12/dat/leap/parm/parm10.dat
Reading title:
PARM99 + frcmod.ff99SB + frcmod.parmbsc0 + OL3 for RNA
Loading parameters: /home/amber12/dat/leap/parm/frcmod.ff12SB
Reading force field modification type file (frcmod)
Reading title:
ff12SB protein backbone and sidechain parameters
Loading library: /home/amber12/dat/leap/lib/aminol2.lib
Loading library: /home/amber12/dat/leap/lib/aminoc12.lib
Loading library: /home/amber12/dat/leap/lib/aminont12.lib
Loading library: /home/amber12/dat/leap/lib/nucleic12.lib
Loading library: /home/amber12/dat/leap/lib/ions08.lib
Loading library: /home/amber12/dat/leap/lib/solvents.lib
> loadamberparams frcmod.ionsjc_tip3p
Loading parameters: /home/amber12/dat/leap/parm/frcmod.ionsjc_tip3p
Reading force field modification type file (frcmod)
Reading title:
Monovalent ion parameters for Ewald and TIP3P water from Joung & Cheatham JPCB (2008)
> mods=loadAmberParams arb.frcmod
Loading parameters: ./arb.frcmod
Reading force field modification type file (frcmod)
Reading title:
remark goes here
> loadAmberPrep arb.prepin
Loading Prep file: ./arb.prepin
> █

```

Figure 2.19 Tleap command line

All protein–ligand complexes were solvated in cubic box of the CHARMM version of TIP3P model [30]. And water extending at least 10 Å in each direction from the solute, while the cut-off distance was kept at 12 Å in order to compute the nonbonded interactions with Cl^- or Na^+ ions added for electrical neutralization.

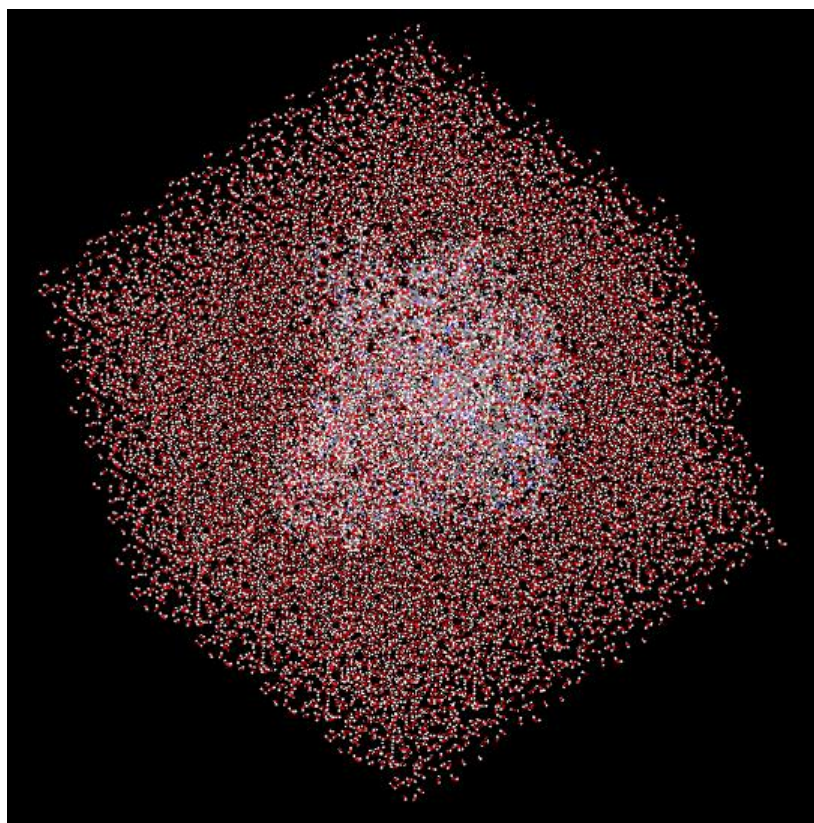


Figure 2.20 The 3D structure of protein–ligand complexes after add water and ions

All simulations were performed under periodic boundary conditions [31], and long-range electrostatics force was treated by using the particle-mesh-Ewald method [32-33]. Bond lengths involving those to hydrogen atoms were constrained using SHAKE, this algorithm was used to satisfy hydrogen bond geometry constraints. Prior to MD simulations, the systems were relaxed by a series of the steepest descent (SD) and conjugated gradient (CG). Periodic boundary conditions and long-range interactions were the computational time and cost required for a MD simulation, increases approximately with the square of the number of particles of the system. Therefore, they are necessary to adopt an approximation in order to model a system with an acceptable number of particles. In addition, it is possible to deduce that a significant number of molecules will be close to the edges of the box when simulating a box. If the normal boundary condition was used to run in simulation, surface forces will be predominant over the bulk ones. Hence, the system properties will be different to the ones observed

within a macroscopic container. One way to interfere these simulation artefacts is to adopt periodic boundary condition.

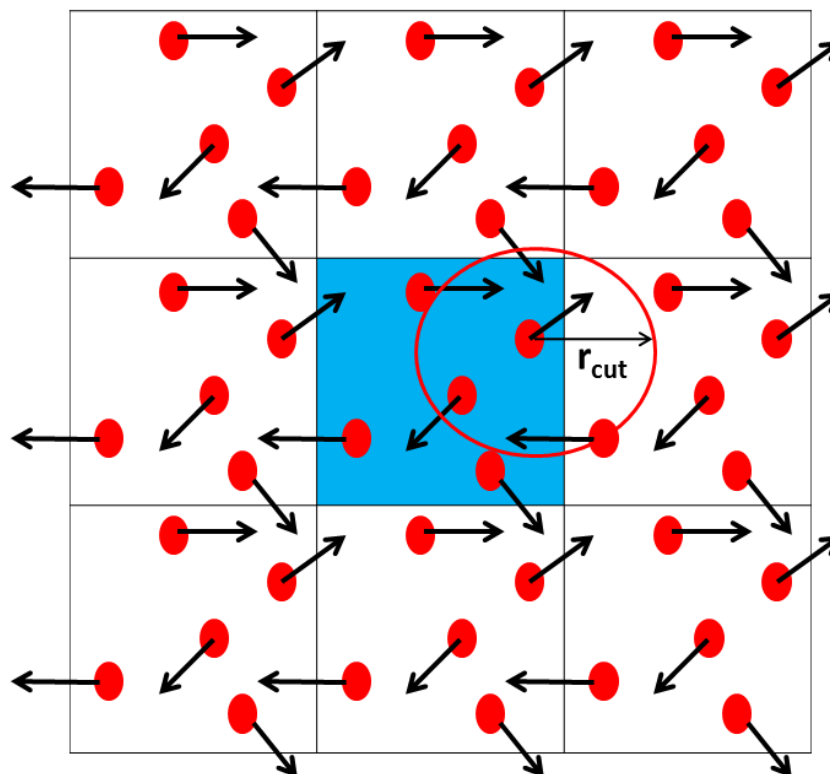


Figure 2.21 Periodic boundary conditions. The center box (blue) was shown along with its first periodic images. The circle was radius, while the box encompasses the nearest periodic image of each of the other molecules. r_{cut} is the cut-off radius applied when calculating non-bonded interactions between particles.

The replicas were surrounded simulation box (called unit cell) in order to build a periodic crystal lattice. So, when one particle leaves the box in one direction, the identical particle coming from the opposite direction with the exact same velocity will be replaced. Therefore, the simulation box is kept the number of atoms in the constant and particles are not experiencing surface forces. To decrease the computational cost of this method, only the non-bonded interactions between particles registered within a certain distance are evaluated (r_{cut} in **Figure 2.19**). In order to interfere interaction between a given particle and multiple images of another particle, the minimum image

convention was used, these states that the cut-off radius must be smaller than half the width of the cell. The combination of periodic boundary conditions and long range electrostatic interactions is evaluated using the Ewald summation [34], and in particular it's derived method well known particle-mesh Ewald [32].

For our work, the CHARMM force-field determining interactions with a cutoff distance of 12 Å using for computing *van der Waals* forces, short-range electrostatic interactions, and the long-range electrostatic interactions were applied [35-36]. The CHARMM force-field has functioned contains terms for both internal and external interactions were following equation [37-38]:

$$U_{\text{CHARMM}} = U_{\text{bonded}} + U_{\text{non-bonded}} \quad (2.3)$$

Where U_{bonded} consists of the following terms,

$$U_{\text{bonded}} = U_{\text{bond}} + U_{\text{angle}} + U_{\text{UB}} + U_{\text{dihedral}} + U_{\text{improper}} \quad (2.4)$$

Where $U_{\text{non-bonded}}$ consists of two terms,

$$U_{\text{non-bonded}} = U_{\text{LJ}} + U_{\text{elec}} \quad (2.5)$$

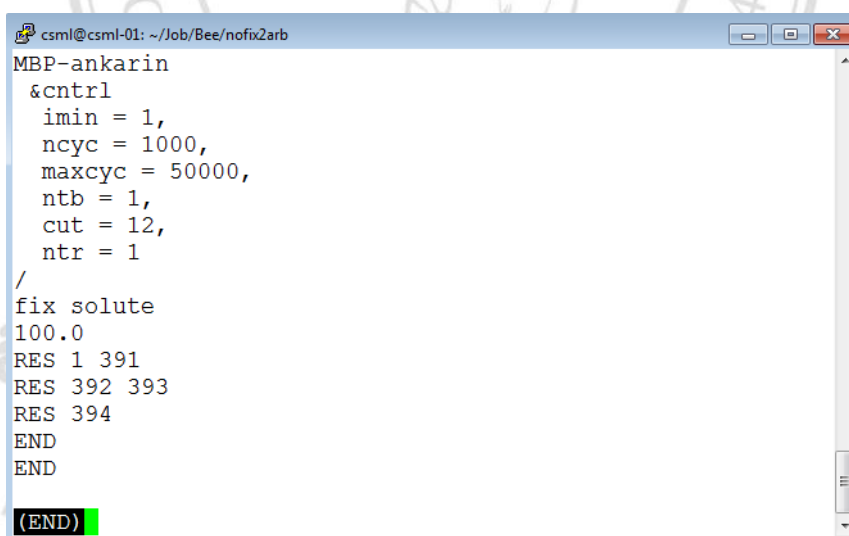
,which bonded terms relating to atoms linked by covalent bonds and non-bonded terms describing the long-range electrostatic and *van der Waals* interactions.

The energy function has the form following equation:

$$\begin{aligned} U(r) = & \sum_{\text{bonds}} K_b (b - b_0)^2 + \sum_{\text{UB}} K_{\text{UB}} (S - S_0)^2 + \sum_{\text{angle}} K_{\theta} (\theta - \theta_0)^2 \\ & + \sum_{\text{dihedrals}} K_{\chi} (1 + \cos(n\chi - \delta)) + \sum_{\text{improvers}} K_{\text{imp}} (\varphi - \varphi_0)^2 \\ & + \sum_{\text{nonbond}} \epsilon \left[\left(\frac{R_{\text{min}_{ij}}}{r} \right)^{12} - \left(\frac{R_{\text{min}_{ij}}}{r} \right)^6 \right] + \frac{q_i q_j}{\epsilon_1 r_{ij}} \end{aligned} \quad (2.6)$$

,where K_b , K_{UB} , K_θ , K_χ , and K_{imp} represent bonds, Urey-Bradley, angle, dihedral angle, and improper dihedral angle force constants, respectively; b , S , θ , χ , and φ represent bonds length, Urey-Bradley 1,3-distance, bond angle, dihedral angle, and improper torsion angle, respectively. The last two terms represent non-bonded interactions; ϵ and R_{min} is the Lennard-Jones well depth and the distance at the Lennard-Jones minimum while q_i and ϵ_1 represent is the partial atomic charge and the effective dielectric constant, respectively.

Minimizations the MD simulations were performed based on each of the minimized systems. The simulations were performed using PMEMD.CUDA from AMBER 12. The particle-mesh Ewald Molecular Dynamic (PMEMD) method was used for computing the long-range electrostatic interactions. The first step, which to optimize the positions of hydrogen atoms and water molecules while imposing position restraints on the protein and ligand. These steps were NVT dynamics (conserved number of particles, volume and temperature).



```

csml@csml-01: ~/Job/Bee/nofix2arb
MBP-ankarin
&cntrl
  imin = 1,
  ncy = 1000,
  maxcyc = 50000,
  ntb = 1,
  cut = 12,
  ntr = 1
/
fix solute
100.0
RES 1 391
RES 392 393
RES 394
END
END
(END)

```

Figure 2.22 Input file for run relaxing step

The minimization step, These steps were NVT dynamics (conserved number of particles, volume and temperature) the same as step before but no longer imposing position restraints on the protein and ligand anymore.

```

csml@csml-01: ~/Job/Bee/nofix2arb
MBP-ankarin
&cntrl
  imin = 1,
  ncyc = 1000,
  maxcyc = 40000,
  ntb = 1,
  cut = 20,
  ntr = 0
/
(END)

```

Figure 2.23 Input file for run minimizing step

This step was performed using by gradually heating over 60 ps from 0 to 310.15 K of NVT dynamics (conserved number of particles, volume and temperature).

```

csml@csml-01: ~/Job/Bee/nofix2arb
MBP-ankarin
&cntrl
  imin = 0,
  irest = 0, ntx = 1,
  ntb = 1, ntp = 0,
  ntc = 2, ntf = 2,
  ntt = 3, gamma_ln = 1.0,
  tempi = 0.0,
  temp0 = 310.15,
  nstlim = 25000, dt = 0.002
  cut = 20.0,
  ntr = 1,
  nrespa = 1,
  ntp = 100, ntwx = 500, ntwr = 100
/
Keep indigo fixed with weak restraints
10.0
RES 1-391
RES 392 393
RES 394
END
END
:

```

Figure 2.24 Input file for run heating step

Next, the energy minimization was carried out on all atoms in the system. In the following step, 1 ns was MD equilibration was carried out employing time step of 2 fs and use NVT dynamics the same as before step.

```
csml@csml-01: ~/Job/Bee/nofix2arb
MBP-ankarin
&cntrl
  imin   = 0,
  irest  = 1, ntx = 7,
  ntb    = 2,
  pres0  = 1.0, ntp = 1, taup = 2.0,
  ntc    = 2, ntf = 2,
  ntt    = 3, gamma_ln = 1.0,
  tempi   = 310.15,
  temp0  = 310.15,
  nstlim = 25000, dt = 0.002
  cut    = 12.0,
  ntr    = 1,
  nrespa = 1,
  ntptr  = 100, ntwx = 100, ntwr = 100
/
fixed with weak restraints
50.0
RES 1 394
END
END
:█
```

Figure 2.25 Input file for run equilibrium step

Finally, 10 ns MD production were conducted for each fully flexible system in the NPT (conserved number of particles, pressure and temperature) at a constant temperature of 310.15 K.

```
csml@csml-01: ~/Job/Bee/nofix2arb
MBP-ankarin (no wrap,run for 1 ns)
&cntrl
  imin   = 0,
  irest  = 1, ntx = 7,
  ntb    = 2,
  pres0  = 1.0, ntp = 1, taup = 2.0,
  ntc    = 2, ntf = 2,
  ntt    = 3, gamma_ln = 1.0,
  tempi   = 310.15,
  temp0  = 310.15,
  nstlim = 500000, dt = 0.002
  cut    = 12.0,
  ntr    = 1,
  nrespa = 1,
  ntptr  = 100, ntwx = 100, ntwr = 100
/
fixed with weak restraints
100.0
RES 60
RES 84
RES 93
RES 258
RES 262
RES 295
RES 392 393
END
END
:█
```

Figure 2.26 Input file for run production step

The command line on each step was shown in Fig. 2.25.

```
pmemd.cuda -O -i 1-relax.in -o 1-relax.out -p t.top -c t.crd -r 1-relax.rst -ref t.crd &
wait
pmemd.cuda -O -i 2-mini.in -o 2-mini.out -p t.top -c 1-relax.rst -r 2-mini.rst -ref 1-
relax.rst &
wait
pmemd.cuda -O -i 3-heat.in -o 3-heat.out -p t.top -c 2-mini.rst -r 3-heat.rst -ref 2-
mini.rst -x 3-heat.mdcrd &
wait
pmemd.cuda -O -i 4-eq-1.in -o 4-eq-1.out -p t.top -c 3-heat.rst -r 4-eq-1.rst -ref 3-
heat.rst -x 4-eq-1.mdcrd &
wait
pmemd.cuda -O -i 4-eq-2.in -o 4-eq-2.out -p t.top -c 4-eq-1.rst -r 4-eq-2.rst -ref 4-
eq-1.rst -x 4-eq-2.mdcrd &
wait
pmemd.cuda -O -i 5-production.in -o 5-production-1ns.out -p t.top -c 4-eq-2.rst -r
2y9xarb-1ns.rst -ref 4-eq-2.rst -x 2y9xarb-1ns.mdcrd &
```

Figure 2.27 The command line on each step of MD simulation

Total energy, kinetic energy and potential energy were analyzed by using command line ‘./process_mdout.perl md1.out md2.out md3.out’

```

csml@csml-01: ~/Job/Bee/nofix2arb
#!/usr/bin/perl

if ($#ARGV < 0) {
    print " Incorrect usage...\n";
    exit;
}

foreach $i ( 0..$#ARGV ) {
    $filein = $ARGV[$i];
    $checkfile = $filein;
    $checkfile =~ s/\./Z//;
    if ( $filein ne $checkfile ) {
        open(INPUT, "zcat $filein |") ||
            die "Cannot open compressed $filein -- $!\n";
    } else {
        open(INPUT, $filein) || die "Cannot open $filein -- $!\n";
    }
    print "Processing sander output file ($filein)...\n";
    &process_input;
    close(INPUT);
}

print "Starting output...\n";
@sortedkeys = sort by_number keys(%TIME);
@sortedavgkeys = sort by_number keys(%AVG_TIME);

foreach $i ( TEMP, TSOLUTE, TSOLVENT, PRES, EKCMT, ETOT, EKTOT, EPTOT, DENSITY, VOLUME, ESCF ) {
    print "Outputting summary.$i\n";
    open(OUTPUT, "> summary.$i");
    %outarray = eval "\%$i";
    foreach $j ( @sortedkeys ) {
        print OUTPUT "$j ", $outarray{$j}, "\n";
    }
    close (OUTPUT);

    print "Outputting summary_avg.$i\n";
    open(OUTPUT, "> summary_avg.$i");
process_mdout.perl

```

Figure 2.28 Input file for calculated energy

Command line for analysis of bond distance was 'ptraj topology_file.top
<rmsd.in> rmsd.out'

Copyright© by Chiang Mai University
All rights reserved

```

csml@csml-01: ~/Job/Bee/nofix2arb/RMSD
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-1ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-2ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-3ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-4ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-5ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-6ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-7ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-8ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-9ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-10ns.mdcrd
reference ../t.crd
rms reference out rmsd_Compstructure2.plot :1-394@CA,C,N,O
rms reference out rmsd_allAtomcompstuc2.plot :1-394
rms reference out rmsd_Comp2.plot :60 84 93 258 262 295 392 393 394@CA,C,N,O
rms reference out rmsd_allAtomcom2.plot :60 84 93 258 262 295 392 393 394
rms reference out rmsd_Compbinding2.plot :60 243 247 256 258 259 262 263 279 280 281 282 285 291 394@CA,C,N,O
rms reference out rmsd_allAtombindingcom2.plot :60 243 247 256 258 259 262 263 279 280 281 282 285 291 394
rms reference out rmsd_protein2.plot :60 84 93 258 262 295 392 393@CA,C,N,O
rms reference out rmsd_allAtompt2.plot :60 84 93 258 262 295 392 393
rms reference out rmsd_allAtomlg2.plot :394
(END)

```

Figure 2.29 Input file for calculated RMSD

The measurement bond distance analysis can be calculated by using command line for analysis of bond distance was ‘ptraj topology_file.top <distance.in> distance.out’

```

csml@csml-01: ~/Job/Bee/nofix2arb/Dist
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-1ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-2ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-3ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-4ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-5ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-6ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-7ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-8ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-9ns.mdcrd
trajin /media/data02/Bee/nofix2y9xarb/2y9xarb-10ns.mdcrd

distance d1 :259@HD22 :394@O6 out d1.out
distance d2 :394@H30 :259@OD1 out d2.out
distance d3 :394@H29 :280@O out d3.out
distance d4 :282@H :394@O5 out d4.out

go
(END)

```

Figure 2.30 Input file for calculated bond distance

REFERENCES

- [1.] D.L. Wheeler, T. Barrett, D.A. Benson, S.H. Bryant, K. Canese, V. Chetvernin, Nucleic Acids Res., 35(2007) 5-12.
- [2.] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, Nucleic Acids Res., 25(1997) 3389-3402.
- [3.] M. Moorhouse, P. Barry, Bioinformatics Biocomputing and Perl. Willy, UK. 2004.
- [4.] R. Durbin, S. Eddy, A. Krogh, G. Mitshison, Biological sequence analysis Probabilistic models of protein and nucleic acid, Cambridge University Press, 1998.
- [5.] M.A. Lrkin, Clustal W and Clustal X version 2.0., Bioinformatics, 23(2007) 2947-2948.
- [6.] S.F. Altschul, J.Mol.Biol., 219(1991) 555-565.
- [7.] M. Sendovski, M. Kanteev, V.S. Ben Yosef, N. Adir, A. Fishman, J. Mol. Biol., 405(2011) 227-237.
- [8.] H. McWilliam, W. Li, M. Uludag, S. Squizzato, Y.M. Park, N. Buso, A.P. Cowley, R. Lopez, Nucleic Acids Res., 41(2013) 597-600.
- [9.] Accelrys Software Inc., Discovery Studio Modeling Environment, Release 2.5, San Diego: Accelrys Software Inc., 2013.
- [10.] D.M. Mount, Bioinformatics: Sequence and Genome Analysis, 2nd ed., Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, 2004.
- [11.] T.F. Smith, M.S. Waterman, J. Mol. Biol., 147(1981) 195-197.
- [12.] P.H. Sellers, Bull. Math. Biol., 46(1984) 501-514.

- [13.] E. J. Gumbel, Statistics of extremes, 2nd ed., Columbia University Press, New York, 1958.
- [14.] S. Karlin, S.F. Altschul, Proc. Natl. Acad. Sci., 87(1990) 2264-2268.
- [15.] A. Dembo, S. Karlin, O. Zeitouni, Ann. Prob., 22(1994) 2022-2039.
- [16.] V.Z. Spassov, P.K. Flook, L. Yan, Protein Eng. Des. Sel., 21(2008) 91-100.
- [17.] H.D. Höltje, W. Sippl, D. Rognan, Molecular Modelling : basic principles and applications, 3rd ed., WILEY-VCH, Weinheim, Germany, 2008.
- [18.] Steinbach, Energy minimization,
http://cmm.cit.nih.gov/intro_simulation/node22.html, (accessed September 2016).
- [19.] R.A. Laskowski, M.W. MacArthur, D.S. Moss, J.M. Thornton, J. Appl. Cryst., 26(1993) 283-291.
- [20.] J.U. Bowie, R. Lüthy, D. Eisenberg, Science., 253(1991) 164-170.
- [21.] R. Lüthy, J.U. Bowie, D. Eisenberg, Nature, 356(1992) 83-85.
- [22.] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, A.J. Olson, J Comp Chem., 19(1998) 1639–1662.
- [23.] D.S. Goodsell, A.J. Olson, Struct. Funct. Genet., 8(1990) 195–202.
- [24.] R. Huey, G.M. Morris, A.J. Olson, D.S. Goodsell, J. Comput. Chem., 28(2006) 1145–1152.
- [25.] F. Osterberg, G.M. Morris, M.F. Sanner, A.J. Olson, D.S. Goodsell, Struct. Funct. Genet., 46(2002) 34-40.
- [26.] G.M. Morris, D.S. Goodsell, R. Huey, A.J. Olson, J. Comput. Aided. Mol. Des., 10(1996) 293–304.
- [27.] D.S. Goodsell, G.M. Morris, A.J. Olson, J. Molec. Recognition., 9(1996) 1–5.
- [28.] R.J. Hyndman, A.B. Koehler, Int. J. Forecast., 22(2006) 679–688.

- [29.] D.A. Case, T.E. Cheatham III, T.A. Darden, H. Gohlke, R. Luo R, K.M. Merz Jr., A. Onufriev, C. Simmerling, B. Wang, R.J. Woods, *J. Comput. Chem.*, 26(2005) 1668-1688.
- [30.] D. Beglov, B. Roux, *J. Chem. Phys.*, 100(1994) 9050–9063.
- [31.] W. Weber, P. Hünenberger, J. McCammon, *J. Phys. Chem. B.*, 104(2000) 3668-4575.
- [32.] T. Darden, D. York, L. Pedersen, *J. Chem. Phys.*, 98(1993) 10089-10092.
- [33.] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, L.A. Pedersen, *J. Chem. Phys.*, 103(1995) 8577-8592.
- [34.] M.P. Allen, D.J. Tildesley, *Computer simulation of liquids*, Oxford University Press, New York, 1989.
- [35.] D.J. Hardy, Z. Wu, J.C. Phillips, J.E. Stone, R.D. Skeel, K. Schulten, *J. Chem. Theory Comput.*, 11(2015) 766-779.
- [36.] O. Ichikawa, K. Fujimoto, A. Yamada, S. Okazaki, K. Yamazaki, *PLoS One*, 11(2016) 1-19.
- [37.] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.*, 4(1983) 187-217.
- [38.] A.D. MacKerell, D. Bashford, M. Bellott, R.L. Dunbrack, J.D. Evanseck, M.J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F.T. Lau, C. Mattos, S. Michnick, T. Ngo, D.T. Nguyen, B. Prodhom, W.E. Reiher, B. Roux, M. Schlenkrich, J.C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, M. Karplus, *J. Phys. Chem. B*, 102(1998), 3586– 3616.