Summer 2018

Computational Models Exploring the Role of Flexibility in Binding Tat Peptide to TAR RNA

Thanh Quoc Le
San Jose State University

COMPUTATIONAL MODELS EXPLORING THE ROLE OF FLEXIBILITY IN BINDING TAT PEPTIDE TO TAR RNA

A Thesis

Presented to

The Faculty of the Department of Chemistry

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Thanh Le

August 2018

The Designated Thesis Committee Approves the Thesis Titled

COMPUTATIONAL MODELS EXPLORING THE ROLE OF FLEXIBILITY IN BINDING TAT PEPTIDE TO TAR RNA

by

Thanh Le

APPROVED FOR THE DEPARTMENT OF CHEMISTRY

SAN JOSÉ STATE UNIVERSITY

August 2018

Brooke Lustig, Ph.D.	Department of Chemistry
Elaine Collins, Ph.D.	Associate Dean, College of Science
Laura Miller Conrad, Ph.D.	Department of Chemistry

ABSTRACT

COMPUTATIONTAL MODELS EXPLORING THE ROLE OF FLEXIBLITY IN BINDING TAT PEPTIDE TO TAR RNA

By Thanh Le

Viral-encoded regulatory proteins interacting with RNA target sequences control the gene expression of lentiviruses, notably the human immunodeficiency virus (HIV) and bovine immunodeficiency virus (BIV). The latter provides a simpler interaction model between the viral trans-activator protein (Tat) and trans-activation response RNA element (TAR), including Tat peptides binding to TAR RNA fragments. This model may offer insights into clinical approaches for treatment, initially through theoretical consideration of the role of peptide flexibility in binding as evidenced by literature-based binding assays and NMR. It has more recently been suggested that DNA-protein binding may also be enhanced by increases in conformational entropy. Here, the previously identified hinge region of the BIV TAR-Tat complex has key residues where K75 and R78 are available for substitution by a more local-flexible glycine; these substitutions allow for alternative RNA-peptide interactions. Initially, we generated 294 possible TAR RNA structures that bind Tat peptides. Then, molecular modeling by UCSF DOCK and GROMACS indicated, for single and double substituted K75G and R78G 14-mer peptides, conformations partially excluded from the major groove of the RNA. Interestingly, the binding energies indicate the mutants are more stable than the native peptide. Future studies should include a broader exploration of initial structures and longer simulation time.

ACKNOWLEDGMENTS

First and foremost, I would like to offer my deepest gratitude to my research advisor Prof. Brooke Lustig for giving me the opportunity to pursue this research. Without a doubt in my mind, his ever encouraging and never-ending support, countless meetings, and guidance are the only reasons I was able to finish this work. I am forever grateful to him for completely having faith in me and my work, and most importantly, for bestowing me with his wisdom. It has been a wonderful three years working with him. I am also thankful to my committee members, Prof. Elaine Collins and Prof. Laura Miller Conrad for their invaluable advice and comments.

A special thanks to undergraduate students, Truc Le, Toan Nguyen, Summer Joyce Batasin, Alex Fong, and Arjit Misra for assisting me throughout this project. I would also like to thank former and current students in Dr. Lustig's lab, Takayuki Kimura, Artem Soshnikov, Curtis Ma, Jonathan Oribello, Saira Montermoso and Radman Ahmadi for providing me with useful advice. And last but not the least, for my parents' never-ending support and care, I offer my heartfelt gratitude.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Human Immunodeficiency Virus vs. Bovine Immunodeficiency Virus

Recognized by the World Health Organization as a major global health issue, the human immunodeficiency virus (HIV) is estimated to have infected 37 million people worldwide by the end of 2016, with an additional 1.8 million infections contracted in 2016 alone.[1] In 2017, approximately 30% of the growing population of HIV-infected individuals were reported to have been unaware of their condition. More than one million deaths were associated with acquired deficiency syndrome (AIDS), a consequence following infection by HIV.[2] Furthermore, sharp declines in the overall effectiveness of antiviral treatments were attributed to the rapid rates of mutation.[3] Combined with rising costs of antiviral therapies across the globe, it is only a matter of time before HIV leads to another global epidemic. In the face of a potential global outbreak, research concerning new and effective treatments for HIV is being met with increasing interest.

It is very common to see bovine immunodeficiency virus (BIV) being used as a model in experiments to study HIV.[4-6] Many similarities ranging from structure to function can be observed in both transactivation response element (TAR) trans-activator of transcription (Tat) systems. The homologous TAR RNA structures shared between HIV and BIV possess traits such as a double helical shape, bulges on 5' ends, and a hairpin loop structure on the second stem.[4] The Tat peptides containing 86 (HIV) and 103 (BIV) residues are divided into six main sections, including N-terminal, cysteine-rich, core, TAR-binding region, glutamine-rich, and C-terminal (Figure 1.1).[4] In both systems, the arginine-rich domain plays a crucial role in interacting with TAR RNA. Before

binding, the peptides generally do not form specific secondary structures.[6,7] As soon as binding occurs in the major groove of RNA, the HIV peptide quickly forms an alpha helical conformation, whereas the protein of BIV exhibits characteristics of beta sheet structure.[7] However, one important difference is that BIV Tat protein does not require the hairpin loop for binding unlike in HIV.

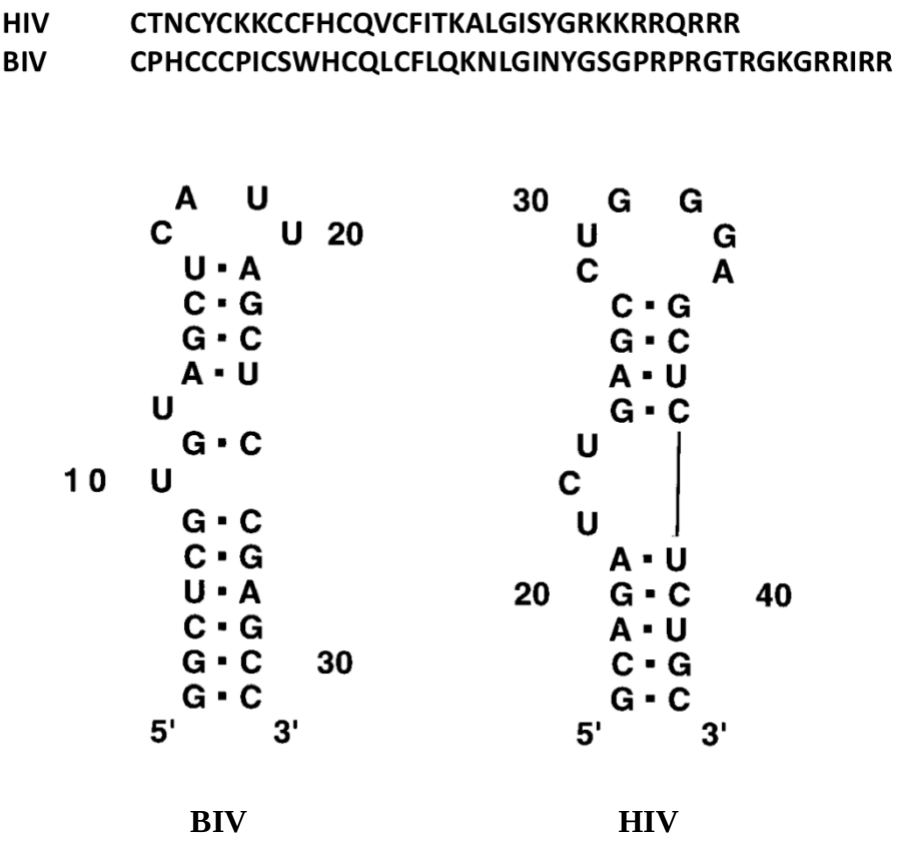

Figure 1.1 Primary sequences of (top) Tat binding domain regions and secondary structures of (bottom) TAR RNA binding domains of HIV and BIV.

The high affinity and specificity binding between the RNA-binding domain of BIV Tat and TAR are facilitated by unique characteristics of both molecules.[7,8] In the case of HIV, its large, three-nucleotide bulge actually destabilizes coaxial stacking of the double

helical stem.[5] However, the two single-nucleotide bulges, consisting of Uracil-10 and Uracil-12, enhance BIV TAR RNA coaxial stacking. Unlike U-12, located in the minor groove, the bulged U-10 is located in the major groove and part of a network of adjacent hydrogen bonds.[7,8] In addition, BIV Tat peptide binding to BIV TAR facilitates the formation a base triple, consisting of U-10, A-13, and U-24.[4] The presence of this co-planar triple significantly stabilizes the binding by reorienting a phosphate group between G-9 and U-10. This phosphate group plays in a major role in the interaction by forming a hydrogen bond with Arg-73's side chain. Due to its position in the triple nucleotide network, U-10 can easily maintain key contacts with both Ile-79 and A-13. Furthermore, structural distortions in the lower and upper helical stems caused by the G-11-C-25 pair widen the major groove, allowing for better penetration of Tat peptide. It is no coincidence that the wild-type peptide has multiple flexible glycine amino acids in its sequence. They permit the formation of a beta turn sheet, which allows the binding domain to fit better in the major groove. Finally, as mentioned previously, the presence of many positively charged arginine amino acids greatly enhances the interaction.

1.2 Flexibility

Due to various factors, including environmental interactions, structural mutations, and even post-translational modifications, proteins tend to adopt molecular and supramolecular behaviors that cannot be predicted accurately from structural data alone. Tzeng and coworkers suggested that DNA-protein interaction heavily depends on conformational entropy.[9] Because effects cannot be easily characterized from protein

mutations, the change in binding activity cannot be predicted. The entropy effect is not just confined to ground state conformations.

In an article published in 1992, Weeks and Crothers stated that kinetic stability may contribute to the binding specificity between the TAR and Tat peptide.[10] They also hypothesized that variations in the TAR RNA structure could lead to different binding kinetics and stability. A larger RNA internal loop or bulge size would provide a more suitable binding confirmation for TAR target, due to increased access to the major groove. In 1998, Lustig and coworkers used lattice-like calculations to study the effect of various RNA bulge sizes on the TAR-Tat interaction; this method utilized rigid double helical stems connected by a bulge region to simulate the behavior of HIV and BIV TAR RNA.[11] The results indicated that an increase in bulge size would allow the Tat peptide to better access the binding domain active site, further showing that binding depends on flexibility.

As mentioned previously in this chapter, secondary and tertiary structures of both TAR RNAs share many common characteristics, including bulges, a hairpin loop, and a double helical stem. Smith and coworkers explored cross-binding experiments where BIV Tat peptide was found to bind to a mutant HIV TAR with the same affinity and manner as to BIV TAR. However, it was also observed that the binding occurs weakly experimentally.[12] They were able to synthesize a hybrid TAR RNA, containing structural characteristics of both BIV and HIV, that greatly increased binding affinity of BIV Tat.[5] They found that the BIV Tat's arginine-rich domain can effectively discriminate the subtle differences in the regions surrounding the binding sites of HIV and BIV TARs. A

2-nucleotide bulge of HIV RNA mutant allows significant binding of a BIV Tat peptide. The HIV hairpin loop was also found to assist binding. The last structural feature of this hybrid RNA is the double helical stem of BIV TAR. The presence of U16-A21 base pair, located in the upper stem above the binding site, is critical for binding. It seems these amino acids act as a clamp to maintain binding site integrity. Ultimately, Tat peptide binds with high affinity to a hybrid TAR RNA, consisting of BIV TAR's double helical stem, HIV TAR's two-nucleotide bulge, and six-nucleotide loop. Clearly, both involved arginine-rich peptides bind to a two-stem RNA fragment. Interestingly, there is some binding of native BIV Tat peptide to wild-type HIV TAR.

Lustig and coworkers constructed a lattice model to study how local residue flexibility can affect the binding of BIV TAR-Tat.[8] It was found that by introducing flexible amino acids, such as glycine, to the hinge regions at positions 75 and 78 (predicted by an NMR ensemble), an increase in flexibility could be observed (Figure 1.2). This predicted increase in flexibility suggests relevant amino acids, including Arg-70, Arg-73, and Ile-79, to better access to existing or alternative RNA contacts flanking and in the bulge region, allowing possibly better binding.

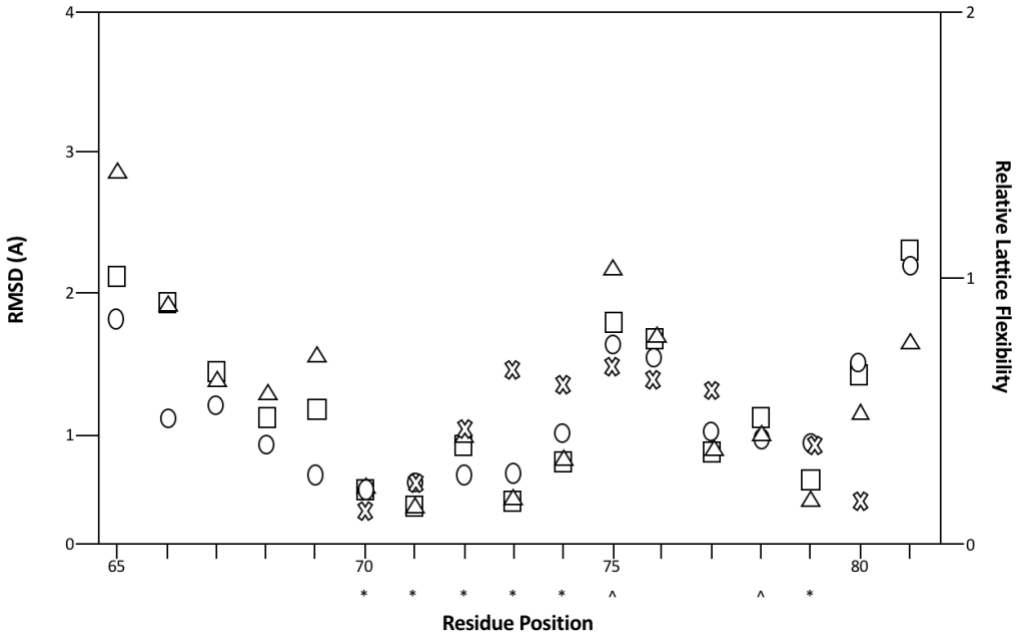

Figure 1.2 Flexibility of bound BIV Tat peptide residues. Root mean square deviation (RMSD) for each residue was obtained from an ensemble of five NMR structures. Amino acids that make contacts with RNA are labeled with *. ^ symbol indicates chosen possible candidates, Lysine 75 and Arginine 78, for glycine substitution.[8]

Another coarse-grained study was conducted by the Lustig group to study how mutations at the peptide's hinge regions can affect the interaction.[13] This study also generated over 12 million lattice structures to model 2-residues per move of a BIV TAR RNA fragment and 11-mer wild-type, K75G, R78G, and K75G-R78G mutant peptides. Here, virtual bonds are indicated spanning two adjacent amino acids based on the C-alpha positions. Consideration of virtual bonds between adjacent C-alpha's (i.e. one-residue per move) allows significantly larger number of conformational states, not easily stored in memory. It was found that systems with peptides having glycine mutation at position 78 or at both positions 75 and 78 experience a clear binding destabilization effect. The

K75G mutant system experiences less destabilization than the others, due to more flexibility.[13]

1.3 Computational Simulation Methods

1.3.1 Coarse-grained Modeling

Coarse-grained modeling is a method of computational modeling that aims to simulate the behaviors of a complex system by generating a simplified (coarse-grained) representation. Proteins tend to experience many structural changes when performing biological functions or interacting with other macromolecules. Because of this, computational modeling of these structural behaviors is useful to the understanding of protein function, providing possible binding and other modalities of interest. The construction of coarse-grained models is usually less computationally expensive than models using an all-atom approach, allowing for simulations of longer time-scales or larger systems. A well-designed coarse-grained model should allow for the proper analysis of large protein systems without the computational work or time investment generally associated with the construction of a classical all-atom model.

1.3.2 All-atom Modeling

Although coarse-grained modeling is more efficient in terms of reducing the time and computational work required for a simulation, it does not capture the behavior of every atom within a system during a simulation. In coarse-grained modeling, the model may assume varying levels of reduced polypeptide representation (main chain represented by all heavy atoms; groups of atoms represented by a united atom). However, when constructing an all-atom model, every atom is represented as a separate part of the

system. All-atom models tend to produce depictions of systems in much higher resolution than coarse-grained models, but at the cost of time and computational expense. When considering the type of simulation method needed for a particular experiment, the aforementioned tradeoff is an important detail that must be considered prior to running a simulation. Often, the two methods are combined, first with coarse-grained, in what is referred to as a hierarchical approach.

1.3.3 Multiple Bennett Acceptance Ratio

The Multiple Bennet Acceptance Ratio (MBAR) method utilizes free energy calculations that embrace an alchemical approach, usually applied in Molecular Dynamics (MD).[14] Alchemical free energy methods invoke the path independent nature of free energy, thus allowing the construction of thermodynamic pathways that include non-physical states as intermediates. The inclusion of non-physical states allows for the efficient calculation of free energy differences; this method is not limited by the natural binding processes and thermodynamic pathways for systems. Therefore, MBAR can bypass unreasonable simulation time scales and streamline the calculation process. Additionally, the inclusion of intermediate states improves the phase space overlap between states. So, the cycle states of complex in water and peptide in water are not by themselves physically realistic. They allow calculation of the relevant thermodynamic state properties. The hallmark of efficient free energy calculations is the presence of significant phase space overlap. Rather than sampling only the end states of the binding interactions, often resulting in little to no phase space overlap, the measuring of phase space overlap between intermediate states across the entire reaction produces a favorable

estimate of the binding free energy. Here, using this and a more coarse-grained, but exhaustive approach, we explore changes in the network of interaction for glycine mutants.

2 METHODS

2.1 Research Overview

2.1.1 General Approach

Studying RNA-protein/peptide interactions computationally has always been difficult due to the unique characteristics of RNA. Since RNA is typically comprised of a single strand, its structure usually displays a high degree of flexibility, which can be difficult to account for in current simulation software. This research explores how certain mutations to the amino acid sequence can give rise to changes in peptide flexibility, and ultimately, binding specificity and affinity. This project is designed to utilize and build upon the data generated from previous studies conducted by the Lustig group. Approaches here include five components: (1) ligand mutation, (2) generation of RNA tertiary structures, (3) docking of complexes, (4) all-atom molecular dynamics simulations, and (5) absolute binding free energy calculations (Figure 2.1). In-house scripts were written to efficiently combine and automate third-party programs including UCSF DOCK[15,16] and GROMACS[17] (see Appendices A and B) (Table 2.1). The work discussed here also emphasizes the robustness of rigid and flexible docking protocols, especially the similarities in results generated by coarse-grained and all-atom modeling.

Table 2.1 List of associated computational methods.

Programs	Functions
SCWRL4[18]	Amino acid mutation
UCSF Chimera[19]	Complex preparation and viewing
Visual Molecular Dynamics (VMD)[20]	Complex analysis and viewing
MC-Fold/Sym[21]	RNA structures generation
UCSF DOCK	Rigid and flexible docking
SETTER[22-25]	RNA RMSD calculation
Open Babel[26]	Chemical file formats converter
GROMACS	All-atom modeling simulation
MBAR	Absolute free energy calculation

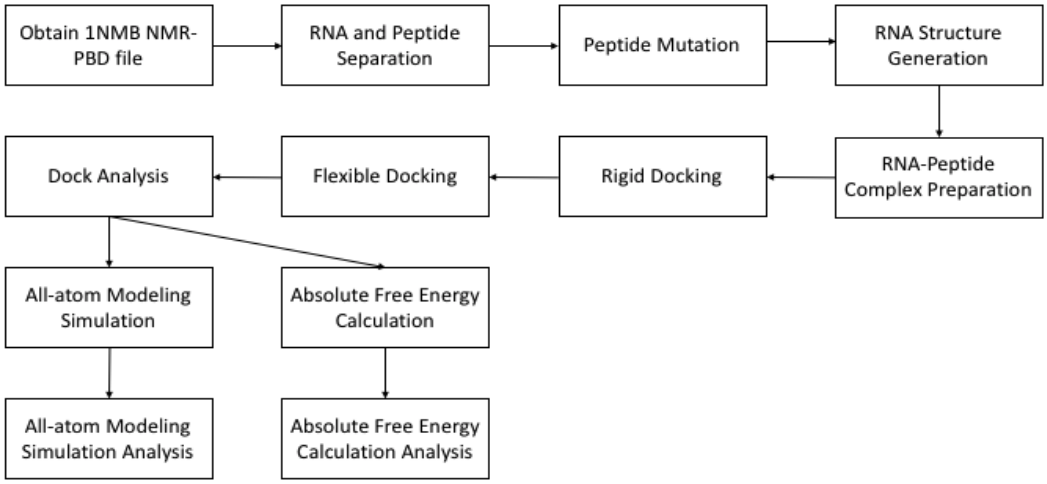

Figure 2.1 Flowchart of steps taken in this project.

2.1.2 Root Mean Square Deviation

Root mean squared deviation (RMSD) compares and numerically measures the difference between two structures (Equation 1). The structures generally used to calculate RMSD are C-alpha atoms or backbone atoms (backbone atoms are used for this experiment's RMSD calculations). Applications of RMSD are diverse and include analyzing structural changes in protein folding simulations, comparing different model

structures constructed through experimentation, and illustrating high-resolution polymer shapes.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N_{atoms}}[r_i(t_1) - r_i(t_2)]^2}{N_{atoms}}} \qquad \textbf{Equation 1}$$

- N_{atoms} is the number of atoms whose positions are being compared.
- $r_i(t)$ is the position of atom i at time t.

2.1.3 Root Mean Square Fluctuation

Root mean square fluctuation (RMSF) calculations of individual atoms are used for monitoring the variability in the conformation of trajectories. Unlike RMSD, RMSF is calculated over time. RMSF is defined as a root mean squared average distance between an atom and its average position in a given set of structures (Equation 2). In essence, the RMSF reveals the fluctuations of an atom about its average position over the course of simulation and can provide insight regarding the flexibility of regions within a protein.

$$\text{RMSF}_\text{I} = \sqrt{\frac{1}{\text{T}}\sum_{\text{j}=1}^{\text{T}}[\text{r}_\text{i}(\text{t}_\text{j}) - \text{r}_\text{i}^{\text{ref}})]^2} \qquad \textbf{Equation 2}$$

- T is the trajectory time over which RMSF is calculated.
- r_i^{ref} is the reference position of residue i.

2.1.4 Radius of Gyration

The radius of gyration of an object, by definition, is the radial distance of a point on the object to the object's given axis of rotation. Mathematically, it can also be defined as the root mean squared distance of points on the object to the object's center of

mass (Equation 3). In the context of this experiment, radius of gyration calculations is used for monitoring the changes in the compactness of a structure. Its values reveal how stable a molecule folds over the course of a simulation.

$$R_g = \left(\frac{\sum_i |r_i|^2 m_i}{\sum_i m_i}\right)^2$$ **Equation 3**

- R_g is radius of gyration with unit in nanometer.
- m_i is the atom i's mass.
- r_i is the position of atom i.

2.2 Preparation

2.2.1 Peptide Mutation

The BIV TAR-Tat complex was obtained by downloading the 1MNB NMR-PDB file from the Protein Data Bank. It contains a 14 amino acid peptide (residues 68-81) bound to a 28-nucleotides RNA receptor target. The two molecules were separated and saved in their different files using UCSF Chimera. Then SCWRL4 was used to substitute lysine (Lys) and arginine (Arg) amino acids, at positions 75 and 78 respectively, with a smaller and more flexible glycine (Gly) which generated K75G, R78G, and K75G-R78G mutant peptides. Unlike ab initio, where molecules are built from scratch, SCWRL4 is designed for structural homology modeling which, in this case, performs side chain predictions based on a template (backbone position for wild-type BIV Tat peptide). This program takes into account the changes in backbone dihedral angles, van der Waals interactions between atoms, and hydrogen bonds.

2.2.2 RNA Tertiary Structures Generation

In UCSF DOCK docking methods, it is assumed that the target is conformationally rigid, which requires only the ligand's conformational, translational, and rotational

degrees of freedom to be sampled during complex formation. Therefore, it is necessary to include all structural changes involving mutant peptides by generating all possible tertiary RNA structures. Utilizing the primary structure of the wild-type BIV TAR RNA and the secondary structure generated by MC-FOLD, 294 structures were constructed by MC-SYM when exploring the conformational space (Figure 2.2). The structures were then divided into four main clusters with different RMSDs using SETTER.

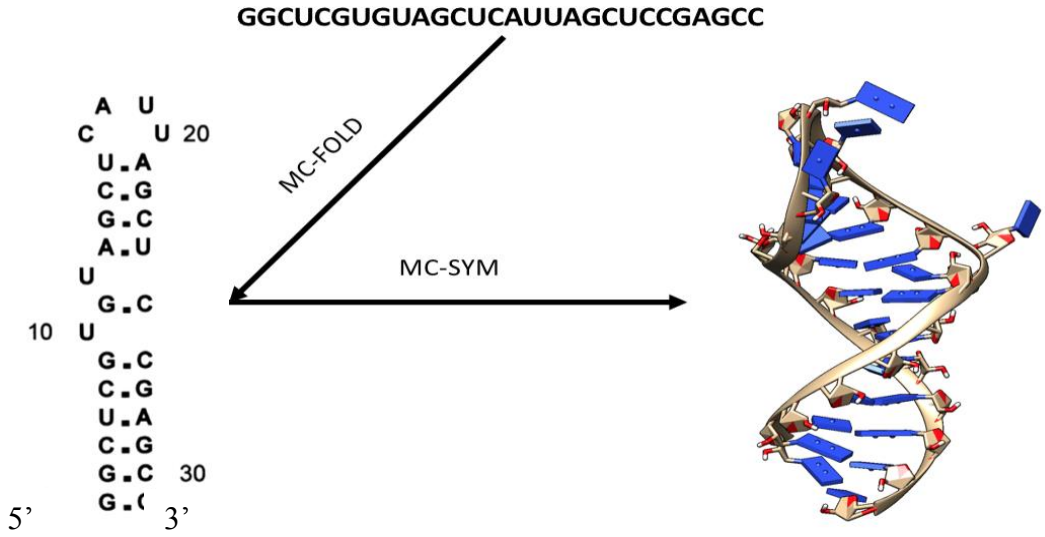

Figure 2.2 Tertiary structure generation pathway of MC-SYM. Determined from primary structure (i.e. sequence) of wild-type BIV TAR RNA, secondary structure generated by MC-FOLD, then MC-SYM generated 294 TAR RNA structures from the exploration of conformational space.

2.3 Rigid and Flexible Docking Methods

295 RNA targets, including 294 generated by MC-SYM and one wild-type (1mnb.pdb), were docked with each of four peptides, K75G, R78G, K75G-R78G, and the wild type. The lowest energy structures generated by flexible docking were further analyzed through molecular dynamics simulations. For each target-ligand system, the system was prepared for docking with UCSF Chimera wherein non-standard residues

and hydrogen atoms were removed and charges were added using a standard AMBER-14SB forcefield.

UCSF DOCK employs what is known as an anchor and grow protocol, an incremental algorithm wherein part of the ligand is used as an anchor to bind to the target. The flexible layers of the ligand, connected by single bonds allowing for free rotation, are then added on the best ranked anchor-portion orientations. Thus, the ligand grows. Rigid docking is relatively less computationally intensive given that only anchoring and orienting are required.

Next, surface spheres were generated using the DOCK suite, with the radius range set to 1.4 – 4 Å. The appropriateness of the parameters was then validated through manual inspection with UCSF Chimera. The spheres within 8 Å of the target RNA were then pruned.

Given the pruned surface spheres, a cubic box (side length: 8 Å) was generated to enable grid calculations. Grid calculations estimate the potential energy in the vicinity of the molecule, where docking occurs (see Equation 4). By using grid estimates, computational time for docking is reduced drastically. Contact scoring that approximates some all-atom features was disregarded (default option) in order to minimize bias in the calculations. Lennard-Jones exponents for attraction and repulsion were set to 6 and 9 respectively, adopting a soft-function approach in lieu of a hard-function (r^{-12}) for repulsion.

$$E = \sum_{i=1}^{lig} \sum_{j=1}^{rec} \left(\frac{A_{ij}}{r_{ij}^{a}} - \frac{B_{ij}}{r_{ij}^{b}} + 332 \frac{q_i q_j}{D r_{ij}}\right)$$ **Equation 4**

- E is the interaction energy with unit in kcal/mol.
- i is the ligand atoms.
- j is the target atoms.
- a is the van der Waals repulsive exponent.
- b is the van der Waals attractive exponent.
- A_{ij} is the van der Waals repulsion parameters.
- B_{ij} is the van der Waals attraction parameters.
- r_{ij} is the distance between i and j atoms.
- q_i is the point charge of ligand atoms.
- q_j is the point charge of target atoms.
- D is the dielectric function.
- 332 is used to convert electrostatic energy to kcal/mol.

Using the sphere and grid inputs, rigid docking was performed. A maximum of 1000 iterations was specified, and energy minimization of the ligand was employed. Rigid poses were scored using the grid-based energy calculations only; no secondary scoring function was used nor were contact scores considered. For each of the 1180 systems (295 targets X 4 ligands), several hundred rigid poses (estimated mean: ~300s) were generated; this method sampled over 300,000 complexes.

Flexible docking was performed on every single rigid pose. The protocol and parameters were almost identical to those of rigid docking. An average of 10 flexible poses per rigid pose were generated, totaling to over 3 million flexible poses. Flexible poses were filtered by their thermodynamic stability, which were estimated by the docking program's grid energy score.

2.4 All-atom Molecular Dynamics Simulation

Groningen Machine for Chemical Simulations (GROMACS) is a molecular dynamics (MD) program that simulates the behavior of molecules under user-provided constraints.

When running GROMACS simulations, two important assumptions regarding the complexes' atoms were drawn: 1) the coordinate and mass are at the atom center; 2) the behavior of electrons is modeled by covalent bonds and atomic charges. Furthermore, the transfer of charges and formation or breakage of bonds do not occur in regular simulation.

Since GROMACS calculations are extremely resource intensive in comparison to other methods, simulating all 3 million poses generated by flexible docking was determined to be problematic. Therefore, the best pose from each mutant complex was chosen based on grid score energy and location of binding to undergo all-atom modeling.

Each system was prepared with UCSF Chimera wherein non-standard residues and hydrogen atoms were stripped. All runs were performed using GROMACS package versions 5.0.7 with AMBER14sb_parmsc1 force field parameter set.[27] Each starting complex was placed in a dodecahedron box placed at a distance 1.0 nm from the box boundary. Transferable intermolecular potential 3P (TIP3P) water molecules were utilized to solvate the box. By adding an ion concentration of 150 nM containing sodium (Na^+) and chlorine (Cl^-), each system's net charge became neutral. The final systems contained 23030 (wild-type system), 16665 (K75G mutant system), 21530 (R78G mutant system), and 19076 atoms (K75G-R78G mutant system).

A long-range method was used for calculating electrostatic interactions, as opposed to a truncation method, which ignores all interactions beyond a certain cutoff. Long range electrostatic interactions were calculated (explicitly taking into account all interactions) using Particle Mesh Ewald (PME) summation algorithm with 4^{th} order interpolation and

0.16 FF grid spacing for the reciprocal sum space. The PME method was designed by Darden to improve the performance of the reciprocal sum (hence, the application of PME in calculating electrostatic interactions).[28] Cutoff of non-bonded interactions, including short-range electrostatic and van der Waals, were set at 1.4 nm. A Linear Constraint Solver (LINCS) algorithm was utilized to constrain all water molecules and bond lengths.[29] Then each system underwent a 1000 steps of steepest descent energy minimization procedure prior to equilibration and MD run. It is extremely important to carry out these simulations with parameter values as close to *in-vivo* conditions as possible. Hence, to maintain a constant temperature of 300 K, the bound molecules and associated atoms were slowly heated up together from 0 K over a period of 2 nanoseconds using V-rescale thermostat and a time step of 2 femtoseconds (fs). Immediately following this step was an NPT equilibration. During the equilibration, the pressure (1 bar) was isotropically maintained by using the Parrinello-Rahman barostat with a coupling constant of 2 picoseconds.[30] Finally, at constant pressure (1 bar) and temperature (300 Kelvin), each MD simulation ran for 100 nanoseconds (ns) with an integration step set at 2 fs. At the end of the simulations, each run yielded 50,000 conformations that would be used for further analysis.

2.5 Absolute Binding Free Energy Calculation

GROMACS package version 2016.3 was used to carry out absolute binding free energy calculation for each chosen flexible docked pose using the non-physical thermodynamic cycle illustrated in Figure 2.3. Unlike the normal MD simulations

described previously, this calculation required the peptide and target-ligand complex to be simulated separately in physical and non-physical intermediate states (alchemical).

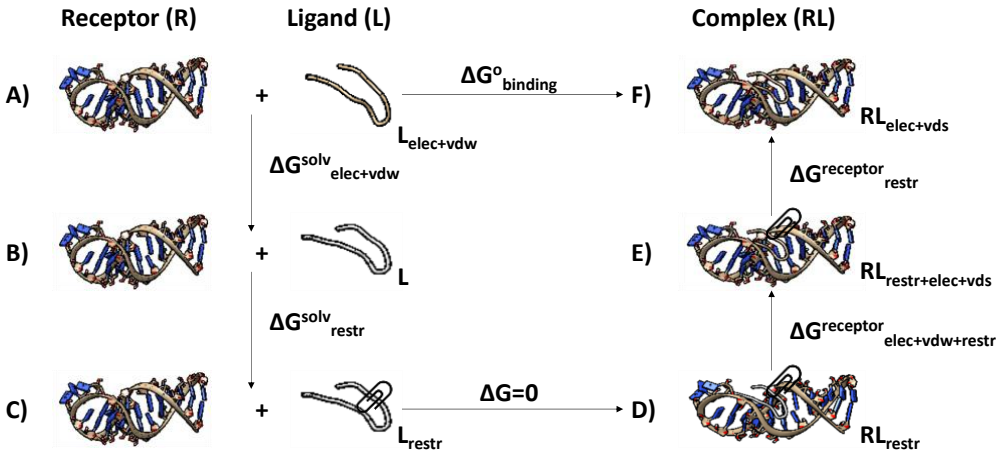

Figure 2.3 Non-physical thermodynamic cycle of BIV TAR-Tat complex. In this cycle, the peptide and bound complex are simulated in the presence of explicit water and restraints (paperclip). A decoupled ligand represented in white color indicates it is not interacting with the surrounding solvent. States D, E, and F involve the simulations of target-ligand complex, whereas A, B, and C involve the simulations of the peptide in solution.

By going through a series of alchemical intermediate states (B, C, D, and E as illustrated in Figure 2.3), the thermodynamic cycle could go from state A, representing the physical unbound stage (ligand) to state F, representing the physical bound stage (TAR-Tat complex). Moving from states A to B, the peptide was decoupled from solution, generating $\Delta G^{solv}_{elec+vdw}$ energy. Since the peptide has stopped interacting with the surrounding environment, a set of restraints was introduced to keep its position and orientation as close to that of the bound ligand. The restraint energy, ΔG^{solv}_{restr} was calculated using the method described by Boresch.[31] The free energy between states C and D remained 0 as the ligand still did not interact with the environment. However, as

soon as coulombic and van der Waals interactions became active, $\Delta G^{receptor}_{elec+vdw}$ could be calculated; the ligand still remained restrained. As soon as the restraints were removed from state E, $\Delta G^{receptor}_{restr}$ was obtained. Finally, with all the free energies across the thermodynamics cycle obtained, the absolute binding free energy, ΔG^{o}_{b}, of the peptide was recovered using Equation 5.

$$\Delta G_b^o = -\Delta G_{elec+vdw+restr}^{receptor} + \Delta G_{elec+vdw}^{solv} + \Delta G_{restr}^{solv} \quad \textbf{Equation 5}$$

- ΔG_b^o is absolute binding free energy.
- $-\Delta G_{elec+vdw+restr}^{receptor}$ is the energy from decoupling ligand from complex.
- $\Delta G_{elec+vdw}^{solv}$ is the energy from decoupling ligand from solution.
- ΔG_{restr}^{solv} is the energy from restraints.
- elec is electrostatics energy.
- vdw is van der Waals energy.
- restr is restraints.

Using a linear alchemical pathway with $\Delta\lambda = 0.05$ for van der Waals and $\Delta\lambda = 0.25$ for coulombic transformations, it was possible to neutralize charges and decouple van der Waals interactions of the ligand. Note that in the complex windows, the bonded interaction is also defined. A total of 30 windows for the complex MD simulations and 20 windows for the ligand MD simulations were applied.

Each window was relaxed by 10,000 steps of steepest descent energy minimization procedure prior to equilibration and MD run. A temperature of 300 K was maintained using Langevin dynamics with a time step of 2 fs. Then, a 4 ns NPT equilibration was followed wherein the pressure of the system was kept at 1 bar using the weak Berendsen coupling algorithm. Furthermore, the bound ligand was restrained with respect to the RNA target by means of three dihedral harmonic potentials, two angles, and one distance. The transformation of van der Waals interactions were modeled using a soft-core

potential. PME summation algorithm with 6^{th} order interpolation, 0.10 FF grid spacing, real space cut-off of 1 nm, and a relative tolerance was used for electrostatic interactions. Then LINCS constraint algorithm was utilized only on hydrogen bonds. Finally, at constant pressure of 1 bar and temperature of 300 K, each window ran for 100 ns with an integration step set as 2 ns.

3 RESULTS

3.1 Ligand Mutation

This project tries to understand how specific mutations in BIV Tat peptide structure can alter binding affinity and specificity. Mutated amino acids at positions 75 and 78 at the hinge regions were shown to have the potential to not only induce unusual structural changes to TAR RNA and, possibly, different flexibility level, but also might interact with TAR at different sites.[8]

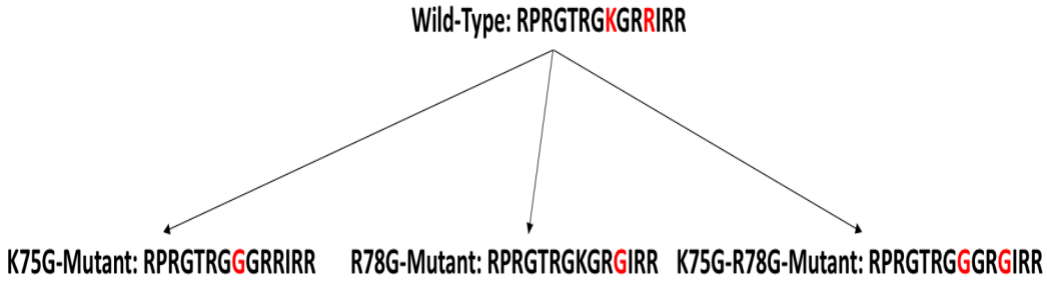

Figure 3.1 Mutation flow chart of peptides. Red colored letters indicate amino acids of interest. In the wild-type primary sequence, red colored K and R stand for lysine and arginine at positions 75 and 78, respectively. In the case of mutated peptides, red colored G's stand for mutated glycine amino acid at positions 75 and 78, depending on mutants.

Three new mutated peptides with different primary sequences were generated by SCWRL4 using the wild type 1MNB Tat as a template (Figure 3.1). As mentioned previously, SCWRL4 does structural homology modeling rather than build the desired molecules from scratch. This method saves a lot of simulation time and possibly provides more accurate structures. Since this project only studies single and double mutation, it is safe to assume the structure and function of the peptide do not deviate much from the original.[32] Hence, the new mutant ligands are expected to share similar tertiary structure as the wild type.

In the case of K75G mutant, the program did a single point mutation, changing lysine amino acid at position 75 to a smaller and more flexible glycine. It is evident that a single hydrogen atom has replaced the long side chain of lysine, resulting in a less crowded area at position 75 (Figure 3.2). A similar situation happened to R78G mutant; instead of replacing lysine, arginine amino acid at position 78 underwent mutation to become glycine (Figure 3.3). This action freed up a significant amount of space at this location and, possibly, eliminated steric hindrance. In the case of K75G-R78G mutant, a double-point-mutation was carried out, replacing both lysine and arginine at positions 75 and 78, respectively, with glycine (Figure 3.4).

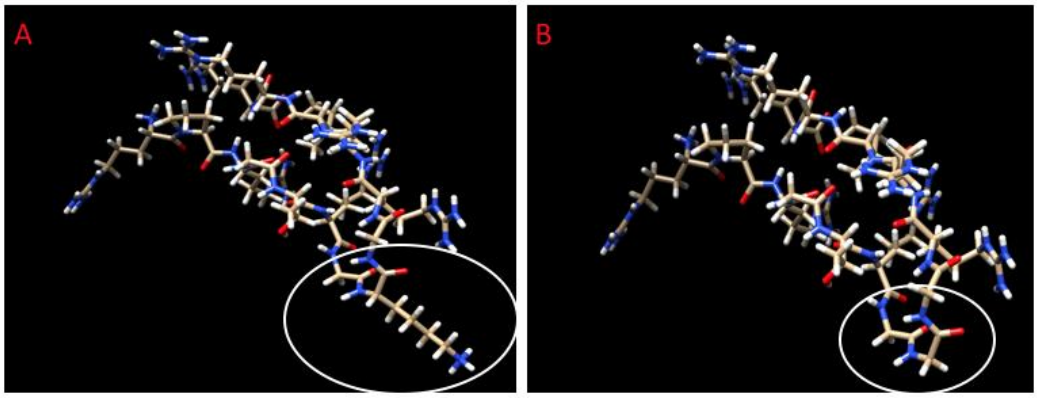

Figure 3.2 Structural comparison between wild-type (A) and K75G mutant (B) peptides. Circled amino acids signify the mutation at position 75 from lysine in wild-type to glycine in K75G mutant.

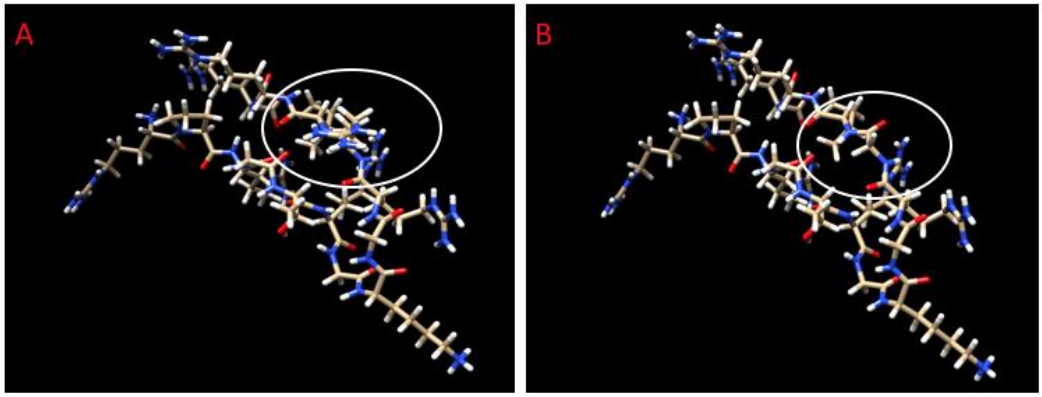

Figure 3.3 Structural comparison between wild-type (A) and R78G mutant (B) peptides. Circled amino acids signify the mutation at position 78 from arginine in wild-type to glycine in R78G mutant.

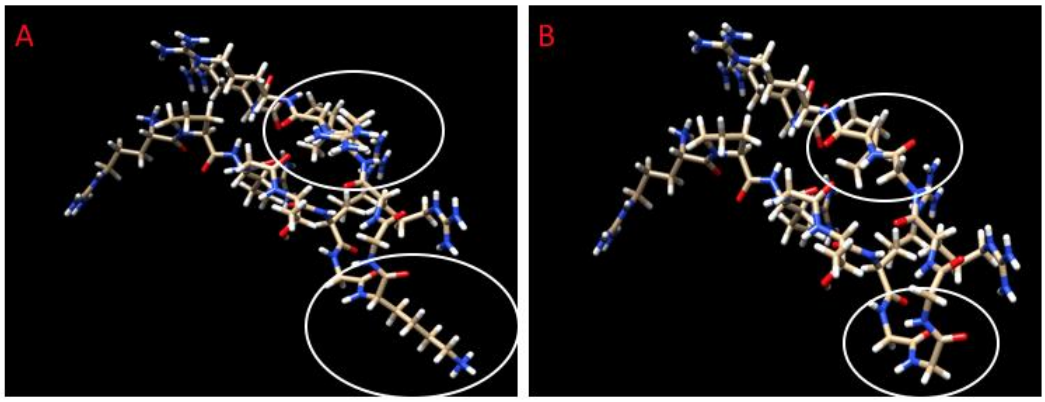

Figure 3.4 Structural comparison between wild-type (A) and K75G-R78G mutant (B) peptides. Circled amino acids signify the mutation from lysine and arginine in wild-type to glycine in K75G-R78G mutant.

3.2 RNA Tertiary Structures Generation

One of the major drawbacks of using a coarse-grained modeling program like UCSF DOCK comes from its algorithm. As mentioned previously, the program assumes the target RNA would remain conformationally rigid during simulation. However, in the case of a protein target, UCSF DOCK does allow a certain degree of flexibility for the protein target.[15] This situation completely ignores structural changes in the RNA target induced

by interactions with mutated peptides. Therefore, it is necessary to explore all possible TAR RNA tertiary structures with different structural characteristics.

Using the primary sequence as a template, MC-Fold created a correct secondary structure of BIV TAR RNA, containing 2 bulges and a 4-amino-acid-hairpin-loop structure (Figure 2.2). Then, by exploring exhaustively the conformational search space of an RNA, MC-Sym generated 294 unique tertiary structures that satisfy all input constraints. These structures were further divided into four main clusters with similar RMSD (Table 3.1); each cluster is represented by an average RNA (Figure 3.5). Based on definition, an RMSD less than 3 Å indicates the new target is very much similar to the wild-type. However, since they are not exactly 0, there are observable differences in these newly generated TAR RNA structures.

Table 3.1 Table of Clusters' RMSDs.

	RMSD (Ångstroms)
Cluster 1	1.484
Cluster 2	1.597
Cluster 3	1.710
Cluster 4	1.845

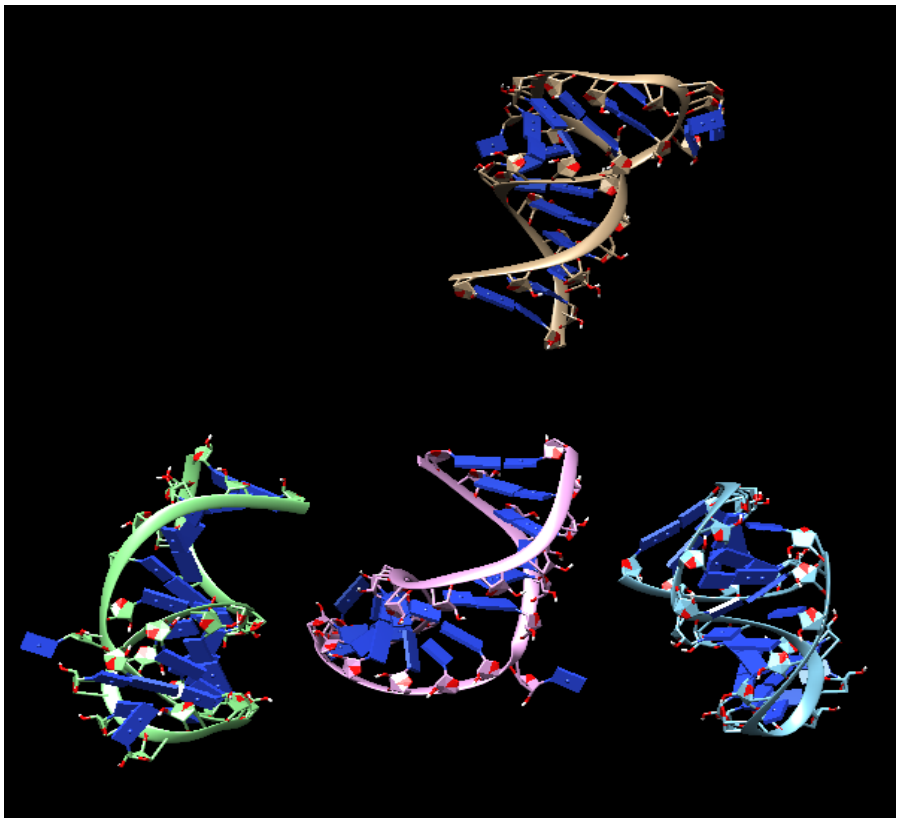

Figure 3.5 Average structure of each cluster. The average structure of cluster 1 having an RMSD of 1.484 Å is represented by a brown-backbone RNA. Blue-backbone RNA represents cluster 2 having an RMSD of 1.597 Å. Pink-backbone RNA represents cluster 3 having an RMSD of 1.710 Å. Green-backbone RNA represents cluster 4 having an RMSD of 1.845 Å.

3.3 Rigid and Flexible Docking

A control study was carried out to validate the parameters used in both rigid and flexible docking protocols and help assess the accuracy of docked poses. Using the protocol described in the method section, UCSF DOCK performed rigid docking simulation on wild-type Tat peptide and TAR RNA target. In this method, the ligand was kept conformationally rigid and allowed to move whereas the target remained static. An RMSD of 2.9687 Å generated from this rigid simulation further confirmed the appropriateness of all parameters. The low RMSD number indicates the docked peptide

did bind close to the wild-type binding site (Figure 3.6). Since the rigid docking method produced a native-like docked complex, flexible docking method was not carried out.

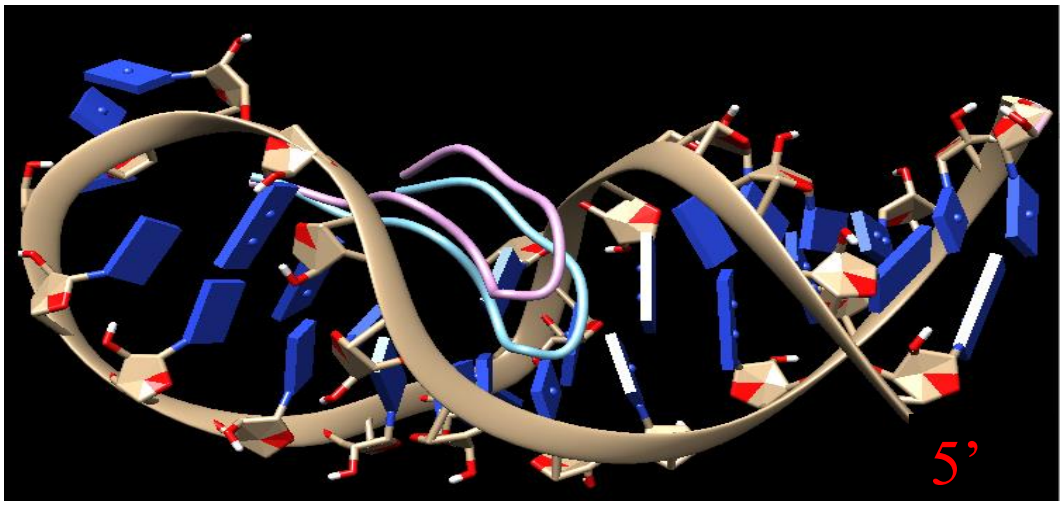

Figure 3.6 Rigid docking control study on wild-type TAR-Tat complex. Blue colored ligand represents rigid docked complex with an RMSD of 2.9687 Å. Pink colored peptide represents NMR wild-type.

Considering over 3 million structures were generated from both rigid and flexible docking simulations, it would be almost impossible to look at every single pose in detail. Hence, the two most reasonable criteria to evaluate selected poses to undergo all-atom modeling simulation are energy and location of binding. The first criterion includes docking energies generated by UCSF DOCK, including grid score, van der Waals, and electrostatic components. They consist of non-bonded (van der Waals and electrostatic interactions) and bonded (dihedral variation, bond stretching, and angle bending) terms.[15] The second criterion, arguably the most important, relies on the location of peptide binding. In current project, the location of binding was assumed to either be near or at the original active site. It is well documented that most proteins' structures can remain

unaffected from one or two points mutations. Therefore, using these criteria, three flexible poses were chosen, one for each mutated peptide.

3.3.1 K75G Mutant

The grid score energy and location of peptide binding of this K75G mutant complex have satisfied the two aforementioned criteria (Figure 3.7). UCSF DOCK estimated the energies based on an implementation of force field scoring. These scores are approximate molecular mechanics interaction energies, including van der Waals and electrostatics (see Equation 4). Hence, a combination of these components gave a total energy of -331.2 kcal/mol. However, note that this number is too large in magnitude due to the utilized force-field-based scoring function. Problems in estimating entropic contributions greatly diminish the accuracy of this method.[33] This inaccuracy is likely due to the absence of a physical model to explain this aspect. Currently, docking programs do not have a way to explicitly calculate the contribution by entropy. Furthermore, it also fails to include any solvation effects, ultimately preventing the calculation of desolvation energies.

The overall structures of both bound complexes, wild-type (Figure 3.7 A) and K75G mutant (Figure 3.7 B), share many similarities, ranging from location of binding to structural characteristics. Similar to the wild-type system, K75G mutated peptide was also observed to bind to TAR RNA at the major groove. The overall tertiary structure of the mutant ligand might play a major role in this interaction. Even though the two peptides do look quite different, they both shared a beta hairpin motif (Figure 3.8). This motif is formed by two adjacent and anti-parallel strands in primary structure and

connected by a short loop of two to five amino acids. Being structurally wider, the peptide of K75G mutant differentiates it from the wild-type peptide. This might explain why the C-terminal tail is located outside of RNA's cavity. Due to this relocation, glycine and arginine amino acids at positions 74, 76, 77, and 81, located on the C-terminal tail, formed many hydrogen bonds with 5'-end nucleotides (Table 3.2). Here, 11 out of 16 K75G mutant hydrogen bonds were created by positively charged arginine amino acids. Similar to the wild-type system (also note Table C.1), the presence of arginine greatly stabilizes the interaction. On the other hand, a few hydrogen bonds also formed between N-terminal amino acids and 3'-end nucleotides, including Arg-70:C23, Thr-72:C-26, and Arg-73:G-27 pairs. These hydrogen bonds clearly show the peptide has bound to the target tail end.

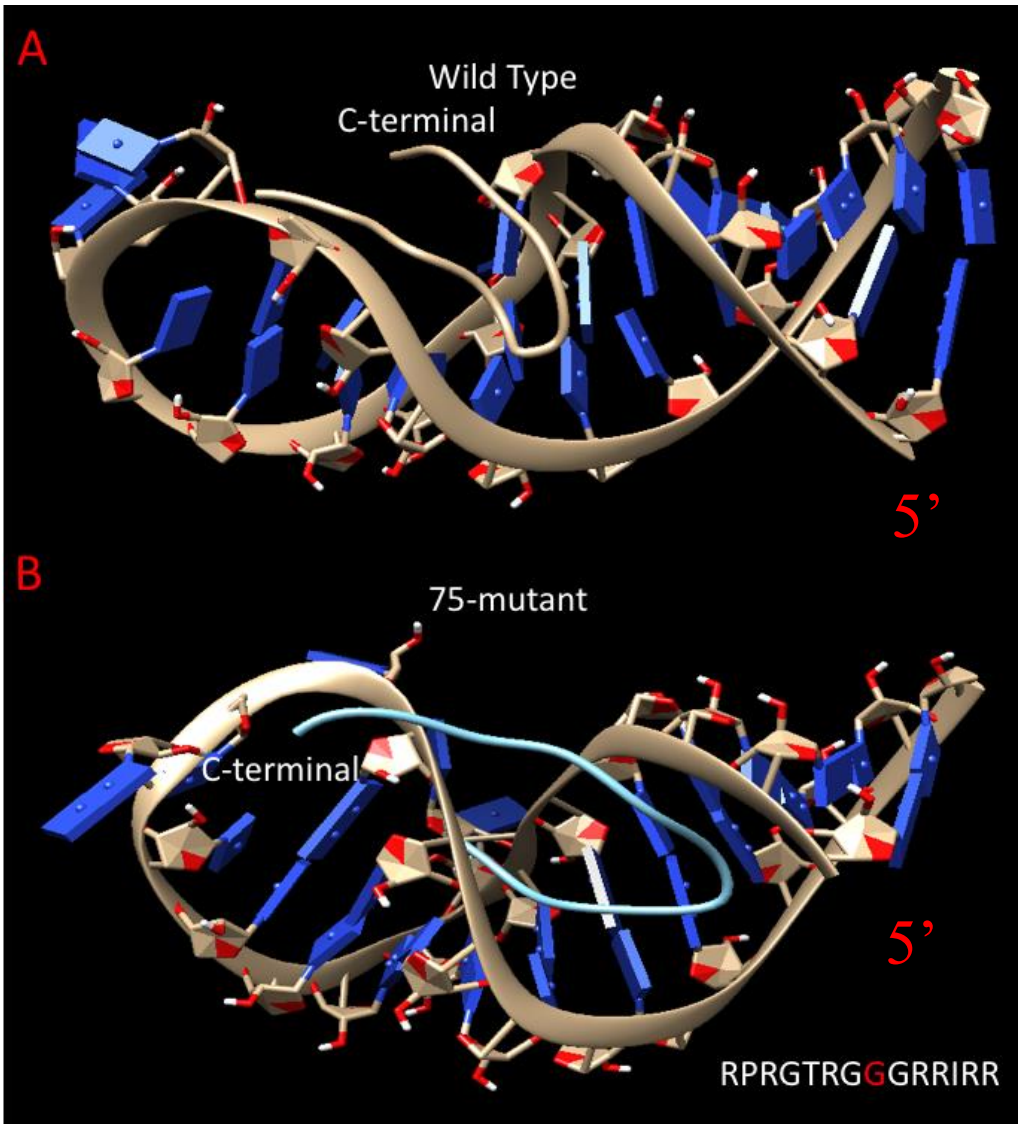

Figure 3.7 Structural comparison between wild-type complex (A) and K75G mutant flexible docked complex (B). A combination of electrostatics and van der Waal energies gave the K75G mutant complex a grid score of -331.2 kcal/mol. Similar to the wild-type, the mutated peptide was also bound to TAR RNA at the major groove. This binding location might be facilitated by the beta hairpin structure of K75G mutant ligand. The ligand forms a total of 15 hydrogen bonds with TAR RNA, which includes 4 in the major groove and 2 in the minor groove. Furthermore, all base pairs in RNA are connected to each other by hydrogen bonds, except the bulges and nucleotides located in the loop. Note that the two targets also share many similarities ranging from G11-C25 pair, coaxial structure, to a large major groove, resulting in a low RMSD of 1.82 Å; base triple does not exist in K75G mutant RNA.

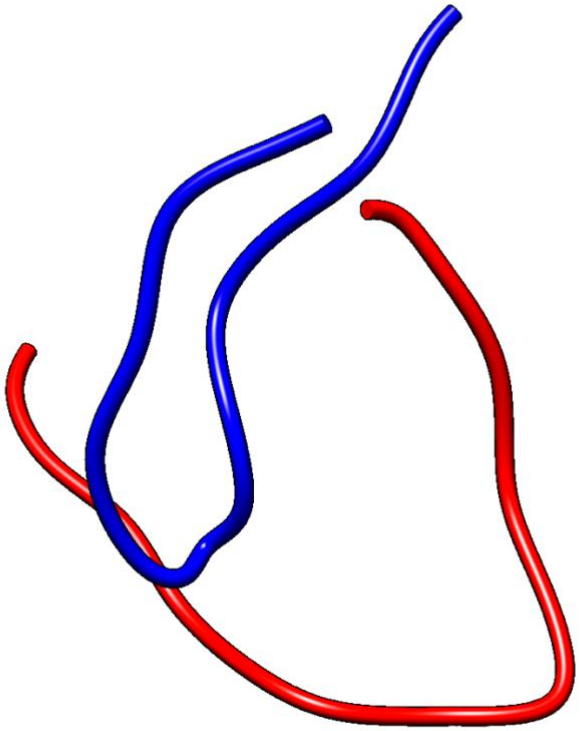

Figure 3.8 Structural comparison of wild-type and K75G mutated peptides with an aligned RMSD of 2.36 Å for aligned peptides. Blue peptide represents the wild type, whereas the red peptide is K75G mutant. They both share a beta hairpin structure. However, the turn region of the mutated peptide is smaller, consisting of arginine and glycine at positions 73 and 74, respectively. In the wild type, glycine, lysine, glycine, and arginine amino acids at positions 74, 75, 76, and 77 form the turn region.

Table 3.2 Table of possible hydrogen bonds formed between K75G mutated peptide and RNA TAR.

Donor	Acceptor
Cytosine-6 N4 (Major Groove)	Glycine-74 O
Cytosine-23 N4 (Major Groove)	Arginine-70 O
Cytosine-26 N4 (Major Groove)	Threonine-72 O
Arginine-68 NH2	Guanine-22 O1P
Arginine-70 NH2	Cytosine-8 O1P
Glycine-71 N	Guanine-9 O6 (Major Groove)
Arginine-73 NH1	Guanine-27 O1P
Arginine-73 NH2	Guanine-27 O1P
Arginine-73 NH2	Guanine-27 O5’
Glycine-76 N	Cytosine-6 O1P
Arginine-77 NH1	Guanine-5 O1P
Arginine-77 NH2	Guanine-5 O1P
Arginine-77 N	Uracil-19 O2P
Arginine-81 NH2	Cytosine-17 O2 (Minor Groove)
Arginine-81 NH2	Uracil-19 O2 (Minor Groove)

Note. Major Groove indicates these possible hydrogen bonds are found in the RNA major groove. Minor Groove indicates these possible hydrogen bonds are found in the RNA minor groove.

3.3.2 R78G Mutant

This R78G mutant complex was chosen based on its grid score energy and location of binding (Figure 3.9). The grid score energy of -344.8 kcal/mole was likely lower than expected due to significant access to the non-major groove like region. Even though guanine at position 11 still formed hydrogen bonds with C-25, the overall structure differs significantly than that of the wild type. Rather than having a linear plane, guanine and Cytosine’s base pairs seem to be twisted. Still, the RMSD between two RNA structures is 1.736Å, indicating small differences. This distortion coupled with the glycine mutation at position 78 might explain why the peptide had its structure extended.

Having a beta-beta link structure, R78G mutant ligand greatly differs from the wild-type structurally, resulting in an aligned RMSD of 2.54 Å (Figure 3.10). Whereas the N-terminal tail still maintained hydrogen bonds in the major groove, its C-terminal tail was observed to make contacts with nucleotides located near the 3'-end. Arginine amino acids still played a major role in this interaction as well, forming 10 out 13 hydrogen bonds (Table 3.3). They can be seen scattered throughout the complex, consisting of positions 68, 70, 73, 77, and 80. At the same time, threonine at position 72 did form a couple hydrogen bonds with guanine-8 and cytosine 25.

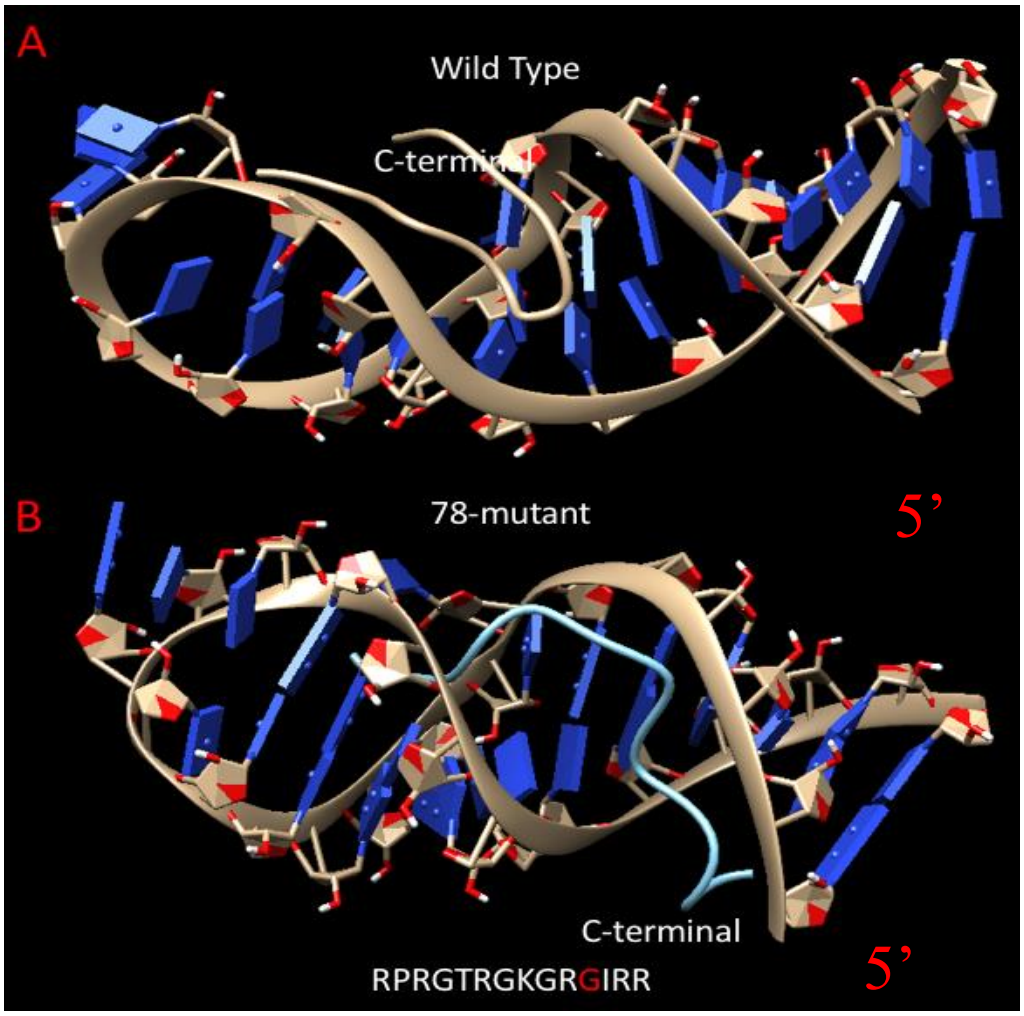

Figure 3.9 Structural comparison between wild-type complex (A) and R78G mutant flexible docked complex (B). A combination of electrostatics and van der Waal energies gave the R78G mutant complex a grid score of -344.9 kcal/mol. Similar to the wild-type, the mutated peptide also bound to TAR RNA at the major groove. This binding location might be facilitated by a beta-beta link structure of R78G mutant ligand. The ligand forms a total of 15 hydrogen bonds with TAR RNA, which includes 4 in the major groove and 1 in the minor groove; the lone hydrogen bond in the minor groove is formed by Guanine-11, an important nucleotide with a role in widening the RNA structure. Furthermore, all base pairs in RNA are connected to each other by hydrogen bonds, except the bulges and nucleotides located in the loop. Note that the two targets also share many similarities ranging from G11-C25 pair to a large major groove, resulting in a low RMSD of 1.736 Å. However, the lack of the base triple and structural characteristics of a coaxial structure and the presence of U10-G11-U12 and C25 bulges differentiate the K75G mutant RNA from the wild-type RNA. Note that these new bulges are observed in dot-bracket (Figure C.8). Furthermore, C-25 also makes a hydrogen bond contact with U-12, which does not exist in the wild-type RNA.

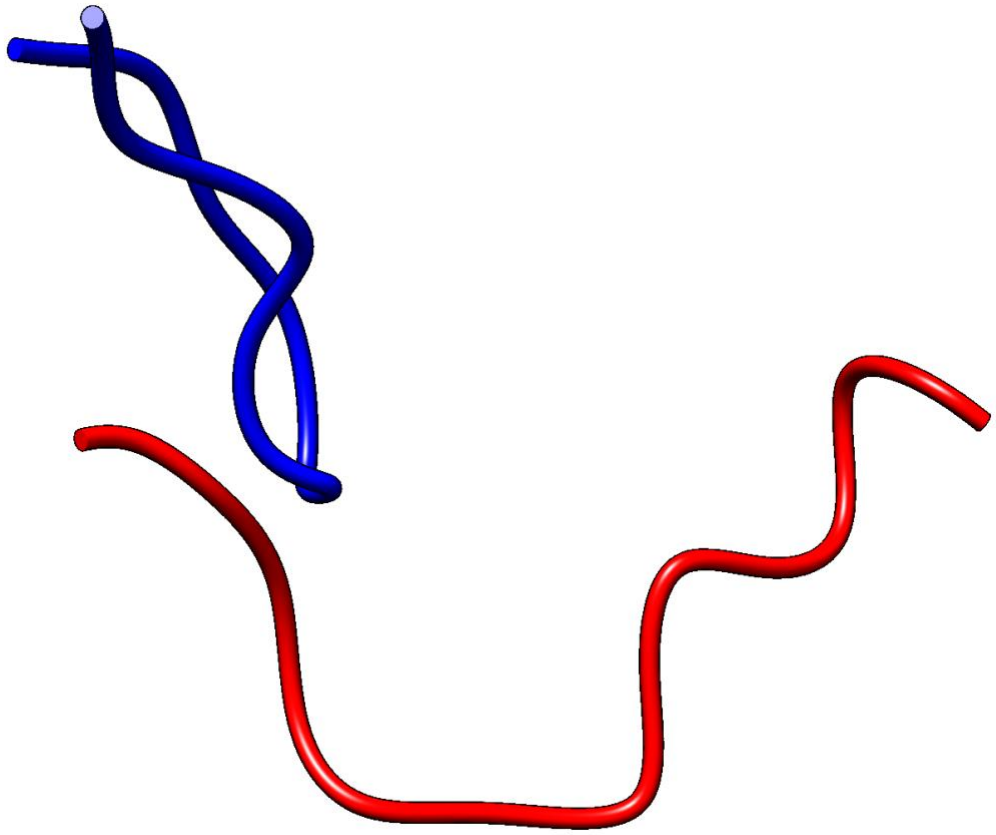

Figure 3.10 Structural comparison between wild-type and R78G mutated peptides. Blue peptide represents the wild-type, whereas the red peptide is R78G mutant. No longer having a beta hairpin, R78G mutant ligand differs greatly from the wild-type structurally, resulting in an aligned RMSD of 2.54 Å for aligned peptides. The new arrangement has its form stretched out to support a bigger loop and shorter strands. Furthermore, the C-terminal strand has 3 kinks while the N-terminal does not. Thus, having the above structural characteristics confirms this mutant peptide clearly has a beta-beta link structure (bb(ppg)bb).

Table 3.3 Table of possible hydrogen bonds formed between R78G mutated peptide and RNA TAR.

Donor	Acceptor
Cytosine-6 N4 (Major Groove)	Arginine-77 O
Guanine-11 N1 (Major Groove)	Threonine-72 O
Guanine-11 N2 (Minor Groove)	Threonine-72 O
Arginine-68 NH1	Adenine-10 O1P
Arginine-68 NH2	Adenine-10 O1P
Arginine-70 N	Guanine-8 O6 (Major Groove)
Arginine-70 NH2	Uracil-7 O4’
Threonine-72 N	Cytosine-20 O1P
Arginine-73 N	Cytosine-5 O1P
Arginine-73 NH1	Guanine-6 N7 (Major Groove)
Arginine-77 NH1	Adenine-25 O1P
Arginine-77 NH2	Adenine-25 O1P
Arginine-80 NH2	Cytosine-23 O1P

Note. Major Groove indicates these possible hydrogen bonds are found in the RNA major groove. Minor Groove indicates these possible hydrogen bonds are found in the RNA minor groove.

3.3.3 K75G-R78G Mutant

Similar to the other two complexes, its unique grid-score energy and location of binding made this K75G-R78G mutant complex a very suitable candidate for all-atom simulation (Figure 3.11). With a grid-score energy of -325.4 kcal/mol, it is the most unstable system. Behaving similarly to R78G mutant, its peptide is also bound to the RNA target perpendicularly. However, this particular ligand does not bind deeply into the major groove. Furthermore, this complex has a different set of hydrogen bond contacts due to reversal of polarity for the peptide’s terminal ends; the C-terminal end is now located in the hairpin loop while the N-terminal tail resides in the far-right end of the target. This might be due to a very narrow minor groove. Despite many differences, the RMSD between the two associated RNAs is still quite low, at only 1.76 Å.

Being a beta-beta link structure like the R78G mutant, this peptide differs structurally from the wild-type, resulting in an aligned RMSD of 1.77 (Figure 3.12). Based on Table 3.4, out of 17 possible hydrogen bonds formed between the RNA target and Tat peptide, positively charged arginine amino acids is responsible for 16 of them, specifically at positions 68 and 71 in the N-terminal tail, 77 and 81 in the C-terminal tail, and 73 in the loop (Table 3.4). The locations of these hydrogen bonds can be found mostly in the hairpin loop, as well as, the 5'-3' ends of the target.

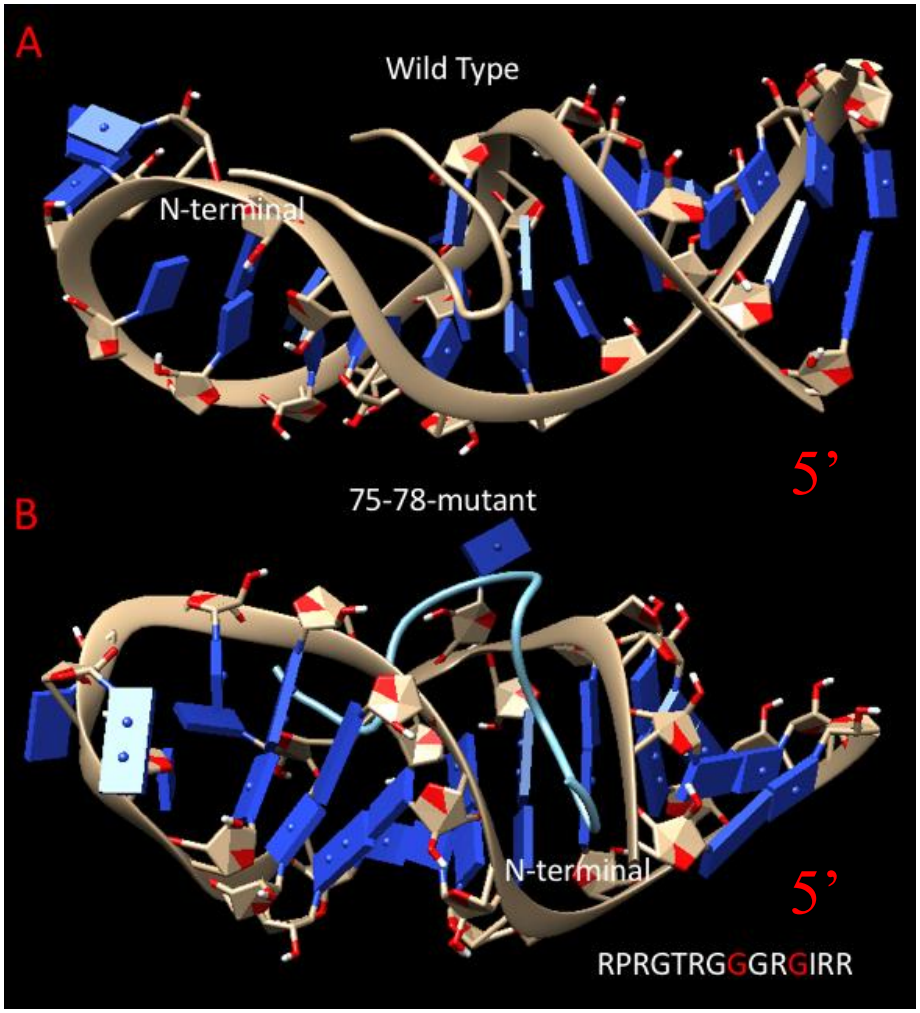

Figure 3.11 Structural comparison between wild-type complex (A) and K75G-R78G mutant flexible docked complex (B). A combination of electrostatics and van der Waal energies gave the K75G-R78G mutant complex a grid score of -325.4 kcal/mol. Unlike the other 2 complexes, the mutated peptide does not bind to TAR RNA at the major groove. The ligand forms a total of 15 hydrogen bonds with TAR RNA, which includes 4 in the major groove. Furthermore, all base pairs in RNA are connected to each other by hydrogen bonds, except the bulges and nucleotides located in the loop. The low RMSD of 1.76 Å results from K75G-R78G mutant RNA having G11-C25 pair and a coaxial-like structure; base triple does not exist.

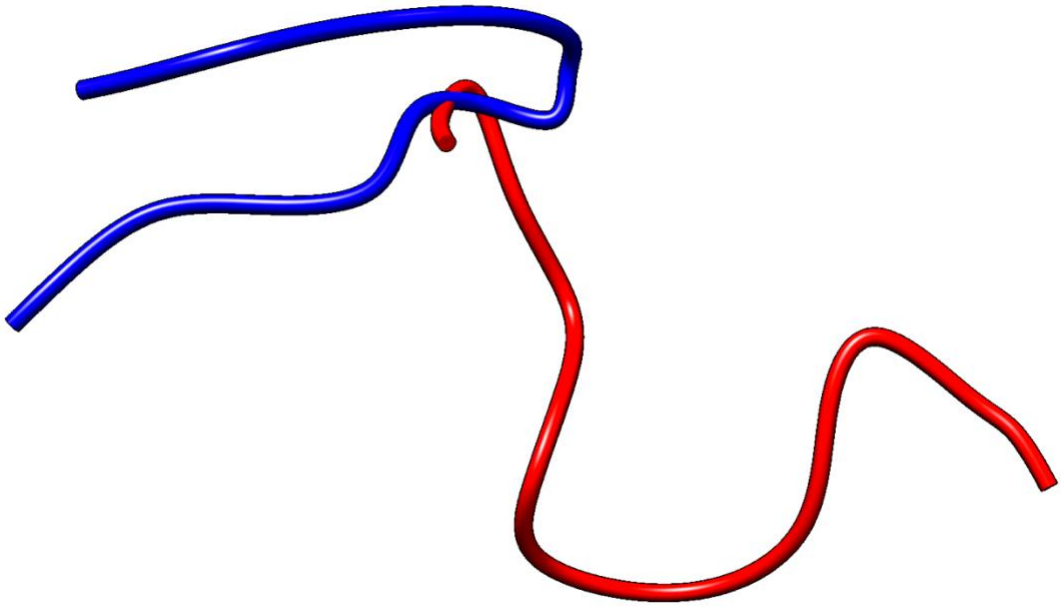

Figure 3.12 Structural comparison of wild-type and K75G-R78G mutated peptides. Blue peptide represents the wild type, whereas red peptide is K75G-R78G mutant. No longer having a beta hairpin, K75G-R78G mutant ligand differs from the wild-type structurally, resulting in an aligned RMSD of 1.77 Å for aligned peptides. The new arrangement has its form stretched out to support a slightly wider loop and shorter strands. Furthermore, the N-terminal strand has 2 kinks while the N-terminal has one. Thus, having the above structural characteristics confirms this mutant peptide clearly has a beta-beta link structure (bb(abp)bb).

Table 3.4 Table of possible hydrogen bonds formed between K75G-R78G mutated peptide and RNA TAR.

Donor	Acceptor
Cytosine-6 N4 (Major Groove)	Proline-69 O
Cytosine-25 N4 (Major Groove)	Arginine-70 O
Arginine-68 N	Cytosine-6 O1P
Arginine-68 NH2	Cytosine-23 O1P
Arginine-70 N	Adenine-28 N7 (Major Groove)
Arginine-70 NH1	Guanine-27 O1P
Arginine-70 NH2	Guanine-27 O1P
Arginine-70 NH2	Guanine-27 O5’
Arginine-70 NH2	Adenine-28 O1P
Arginine-73 NH1	Cytosine-6 O2P
Arginine-73 NH2	Cytosine-6 O2P
Argnine-73 NH2	Cytosine-23 O2P
Arginine-77 N	Guanine-11 O1P
Arginine-77 NH1	Adenine-21 N7 (Major Groove)
Arginine-81 N	Uracil-12 O4’
Arginine-81 NH1	Guanine-11 O2P
Arginine-81 NH1	Uracil-12 O2P

Note. Major Groove indicates these possible hydrogen bonds are found in the RNA major groove.

3.4 All-atom Modeling Simulation

3.4.1 Root Mean Square Deviation

Figure 3.13 shows RMSD as a function of MD for all systems, containing water molecules, ions, peptides, and RNAs. Each system is labeled with different colors: wild-type, K75G mutant in black, R78G mutant in red, and K75G-R78G mutant in green; this color scheme is used throughout section 3.4 for all three mutant systems. These values were calculated by comparing every single conformation of the 50,000 generated by each simulation to their respective starting structures. This step helps confirm whether each simulation has reached convergence. Based on Figure 3.13, all three systems experienced

a large increase in RMSD values instantaneously and stabilized after 20 ns. The wild-type also experienced the same trend (Figure C.1). The mean values of RMSD are 4.32, 4.21, and 3.94 nm for R78G mutant, K75G-R78G mutant, and K75G mutant, respectively. The low value of K75G mutant is clearly due to the unique structural characteristics of both peptide and RNA target. As mentioned previously, both structures are very similar to the wild-type. Regarding the other two mutants, it is necessary to analyze specific flexible regions of their systems.

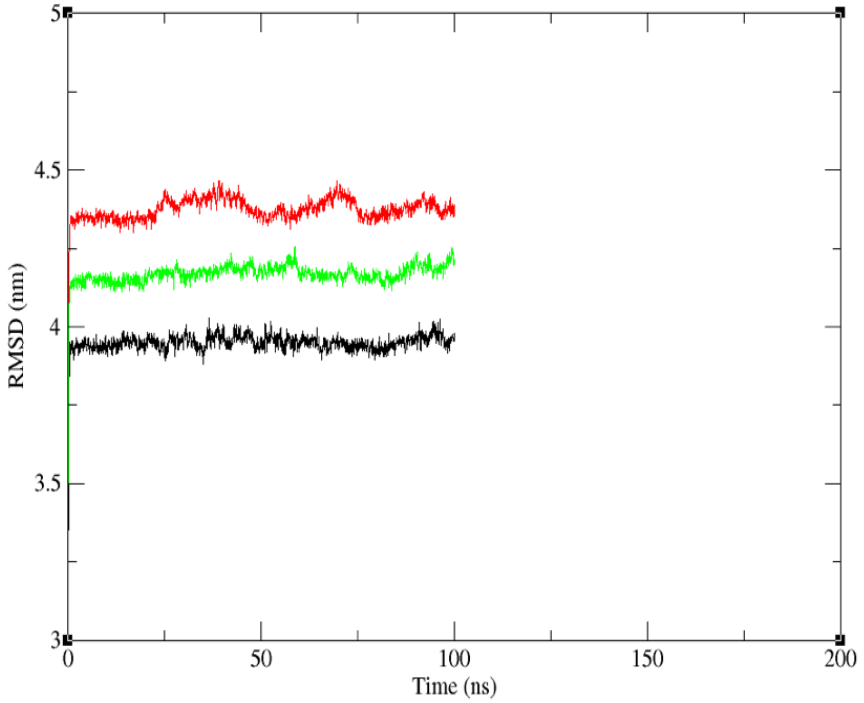

Figure 3.13 RMSD values of K75G mutant, R78G mutant, and K75G-R78G mutant systems. Each system is color coded: R78G mutant in red, K75G-R78G mutant in green, and K75G mutant in black. These values are calculated by comparing all atoms of every single pose of the 50,000 generated by each simulation to their respective starting structures over the course of 100 ns. For a brief interval after the simulation has started, all three mutants experienced a drastic increase in RMSD. Then, they all stabilized after 20 ns and had mean RMSD values of 4.32, 4.21, and 3.94 nm for R78G mutant, K75G-R78G mutant, and K75G mutant, respectively.

3.4.2 Radius of Gyration

Another important piece of information allowing a closer look at these complexes is radius of gyration, which measures the compactness of a structure overtime. As is evident from Figure 3.14, the radius of gyration values of K75G mutant TAR RNA remains very stable, in its folded form over the course of 100 ns at 300 K. The same can be said for R78G mutant RNA, starting at 0.943 nm and ending at 0.978 nm. The radius of gyration of the wild-type RNA also remained quite stable (see also Figure C.2). However, K75G-R78G mutant RNA experiences the biggest increase in its radius; it starts from roughly about 0.875 nm and ends at 0.95 nm. Note that the difference is very small, less than 1 Å. According to Figure 3.15, the K75G-R78G mutant peptide expresses a similar behavior as its TAR RNA, having its radius of gyration growing slightly larger, from 1.385 nm to 1.412 nm. On the other hand, the opposite is seen for K75G and R78G mutants. Their gyration radiuses contract from 1.487 nm to 1.386 nm for R78G mutant and 1.349 nm to 1.324 nm for K75G mutant. The radius of gyration of the wild-type peptide remained quite stable (see also Figure C.3).

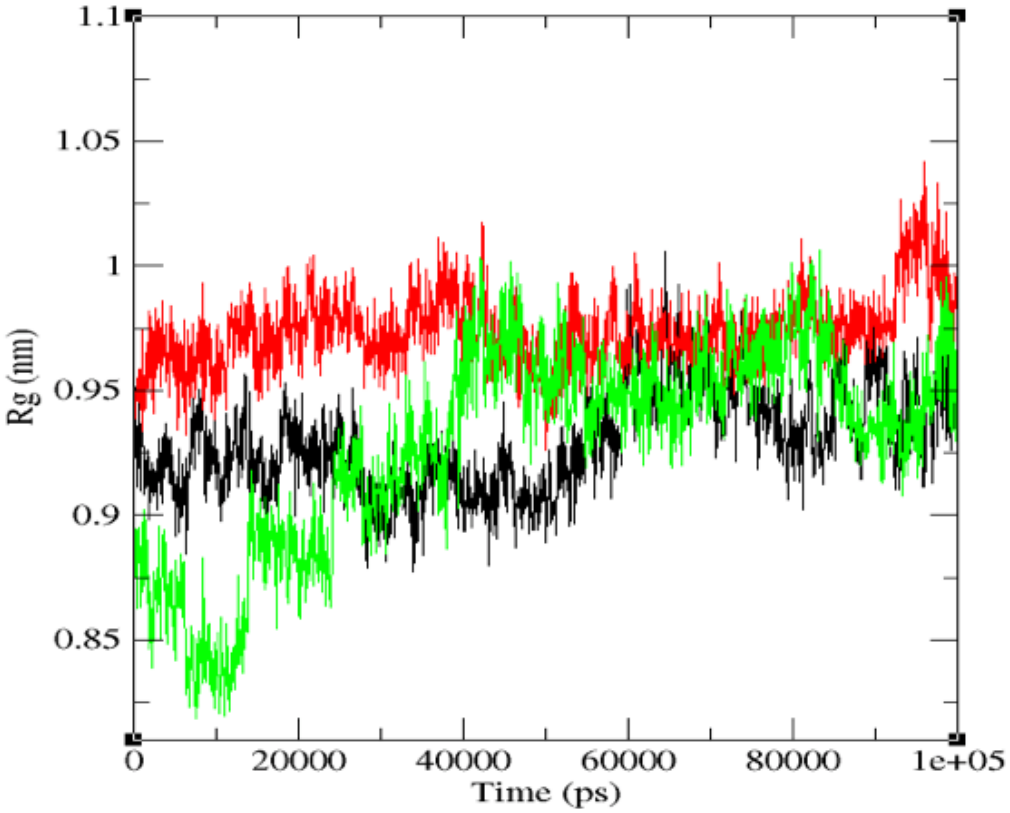

Figure 3.14 Radius of gyration values of all three RNA's. The color scheme is identical to that of RMSD. Over the course of 100 ns, the radius of gyration of each RNA structure is carefully monitored for any changes. Both K75G and R78G mutant RNA's radiuses remained very stable. On the other hand, RNA of the K75G-R78G mutant system experienced an increase in radius of gyration, growing from 0.875 nm to 0.95 nm.

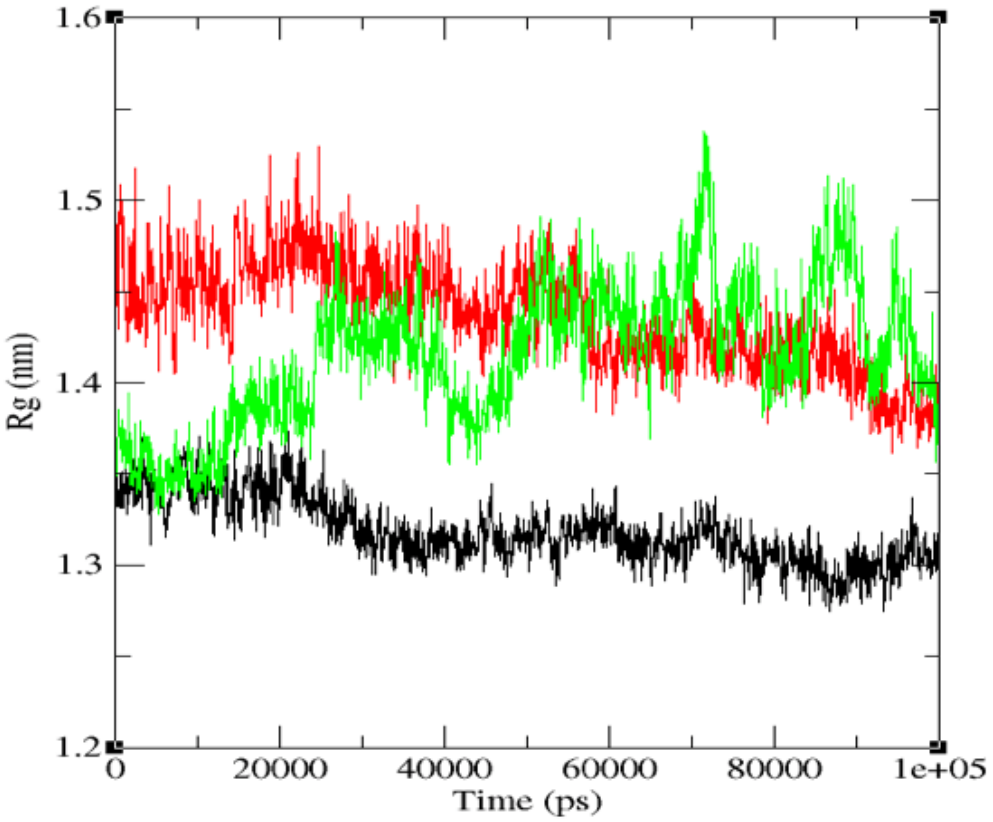

Figure 3.15 Radius of gyration values for all three mutant peptides. The color scheme is identical to that of RMSD. Unlike their RNA counterparts, peptides of both K75G and R78G mutant systems actually contracted over the course of simulation. The K75G-R78G mutant peptide expresses a similar behavior as its TAR RNA, having its radius of gyration growing slightly larger, from 1.385 nm to 1.412 nm.

3.4.3 Root Mean Square Fluctuation

As mentioned previously, with the help of root mean square fluctuation values, one can assess the flexibility of different regions and how specific amino acids can affect the interaction. The plot and structures in Figure 3.16 show RMSF values of RNA calculated from equilibrium trajectories of three systems, with blue representing the lowest-most stable and red the highest-most fluctuating in term of flexibility. A common characteristic shared among three TAR RNA's is the high flexibility level of A-18, U-19, and U-20

located in the hairpin loop, especially A-18; this unique characteristic is also shared by the wild-type TAR target (Figures C.4 and C.5). It is very interesting to see that the hairpin loop of R78G mutant expresses the highest level flexibility when compared to the other two. Other flexible regions can be seen in K75G and K75G-R78G mutant structures. G-9 and G-11 show a medium level of flexibility in K75G mutant. The K75G-R78G mutant has slightly different flexible regions. The 5' and 3' prime ends have a moderate level of flexibility, which means there might have been interactions between nucleotides located there and mutated peptides.

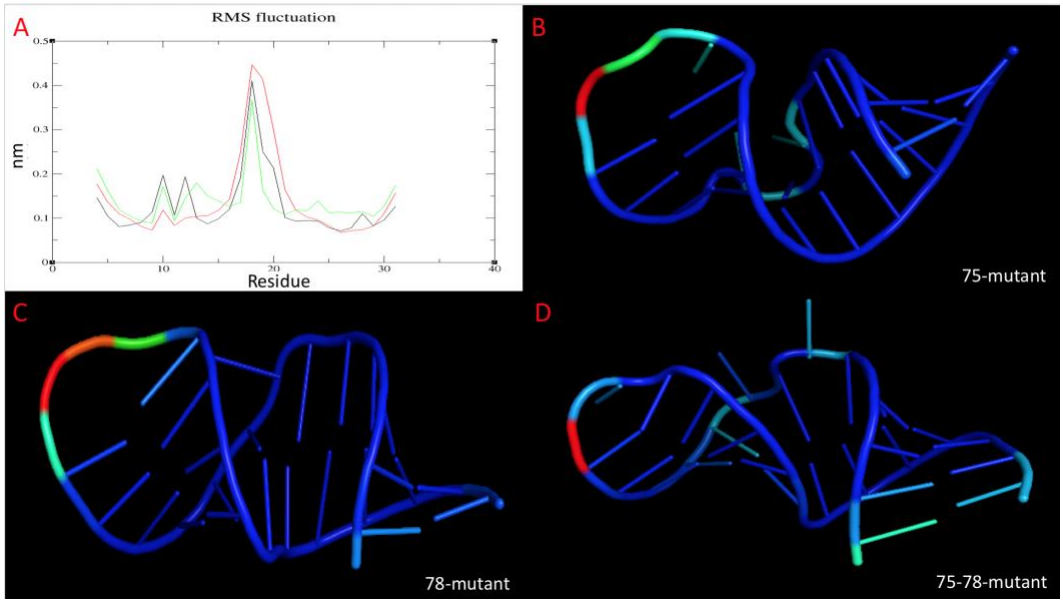

Figure 3.16 Root Mean Square Fluctuation values (A) for all three RNA's (B-D). Colors ranging from blue representing the most stable to red the most fluctuating. All three RNA structures show high level of flexibility in their hairpin loop, especially nucleotides A-18, U-19, and U-20. According to Figure 3.16-A, C-15 is the most flexible having a value over 3.5 nm in all RNA's. Other flexibles regions are also observed as well. Both Guanine nucleotides at positions 9 and 11 have a moderate level of flexibility. K75G-R78G mutant RNA differs from the other structures due to its flexible 5' and 3' ends.

All mutated peptides show high level of flexibility throughout their structures (Figure 3.17). K75G mutant peptide exhibits an astounding flexibility level at

both terminal ends of its structure. Furthermore, the mutated glycine amino acid at position 75 (Figure 3.17 A) is quite flexible as well. It might have introduced flexibility to nearby amino acids including Arg-73, Gly-74 and Arg-77. Looking at the R78G mutant, its backbone appears to not have moved as much compared to its K75G mutant counterpart. However, the R78G mutant peptide still shows similar flexibility characteristics, such as flexible terminals and center. When comparing K75G-R78G mutant peptide to the K75G and R78G mutant peptides, it shows similar flexible terminals, center, and other characteristics. However, it varies in the second half of the structure (starting from Arg-76 and ending at Arg-81). This section is more flexible than the corresponding sections from the other two peptides. Another interesting observation is throughout the three peptides, mutated arginine at position 78 stays very rigid. On the other hand, the control study using the wild-type peptide shows that amino acids at positions 75 and 78 have the highest level of flexibility when compared to other amino acids (see also Figures C.6 and C.7).

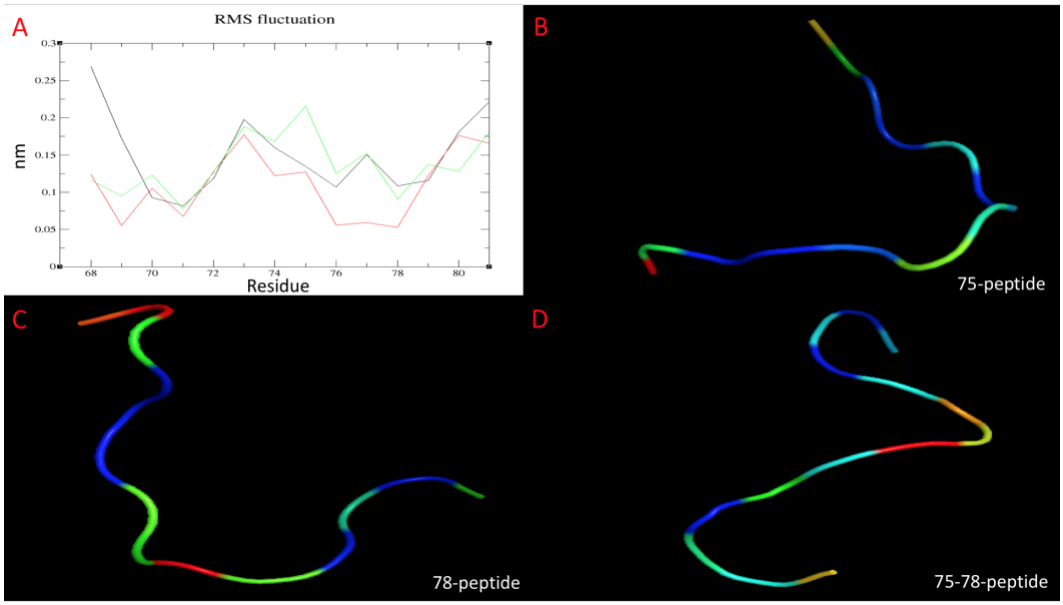

Figure 3.17 Root Mean Square Fluctuation values (A) for all three mutant peptides (B-D). Colors ranging from blue representing the most stable to red the most fluctuating. Unlike their RNA counterparts, all three mutant peptides show high level of flexibility throughout their structures. They all share three common characteristics, which include highly flexible C-terminal tail, amino acids in the loop, and a very rigid amino acid at position 78. Different than the other two peptides, K75G mutant peptide also has a very flexible N-terminal tail as well. Mutated glycine amino acid at position 75 in K75G-R78G mutant seems to be very flexible, having a value of over 0.2 nm.

3.4.4 Hydrogen Bond Analysis

In order to investigate the detailed interactions between mutated peptides and RNA targets, hydrogen bond analysis was carried out for each system. Only those with occupancy higher than 10% for K75G mutant, R78G mutant, and K75G-R78G mutant systems are shown in Tables 3.5 through 3.7. It should be noted that VMD in conjunction with GROMACS does not simply allow characterization of individual hydrogen bonds, but for the tens of thousands of frames, a more coarse-grained assessment is provided.

Similar to the hydrogen bond analysis done for the DOCKed system, Table 3.5 shows, for comparable GROMACS results, positively charged arginine amino acids still

play a significant role in this interaction; this characteristic is also observed in the wild-type system (see also Table C.2). Ten out of 18 pairs are formed between K75G mutated peptide and TAR RNA involving arginine. Furthermore, amino acids, ranging from Arg-68 to Lys-75, make 13 residue-nucleotide contacts with the target, indicating the first half of this peptide plays a major role in the interaction. Amino acids involved in forming high occupancy contacts can be seen interacting with nucleotides located near the target's end, such as Arg-73:G-27, Arg-70:C-8, Gly-71:G-9, and Arg-73:A-28 pairs. By making nine residue-nucleotide bonds out of 18, the role played by both Arg-73 and Thr-72 is shown to be important. This is further confirmed by Arg-73:G-27 and Thr-72:C-26 pairs, which show up in both coarse-grained and all-atom modeling simulations.

Table 3.5 Possible residue-nucleotide contacts with hydrogen bonds formed between K75G mutant peptide and RNA TAR.

Peptide	RNA	Occupancy
Arginine-73	**Guanine-27**	**93.92%**
Arginine-70	**Cytosine-8**	**91.08%**
Glycine-71	**Guanine-9**	**88.24%**
Arginine-78	Guanine-22	81.37%
Arginine-73	Adenine-28	77.45%
Arginine-80	Uracil-20	58.82%
Threonine-72	Uracil-7	57.82%
Threonine-72	Cytosine-25	52.94%
Glycine-76	**Cytosine-6**	**40.20%**
Thronine-72	Cytosine-26	37.25%
Glycine-71	Cytosine-8	36.27%
Arginine-81	Adenine-21	35.29%
Arginine-73	Cytosine-26	30.39%
Arginine-81	**Uracil-19**	**25.49%**
Threonine-72	**Uracil-24**	**24.51%**
Glycine-74	**Cytosine-6**	**17.65%**
Arginine-73	Uracil-7	13.73%
Arginine-73	Cytosine-26	12.75%

Note. These pairs were generated using the default parameters of VMD. Bold indicates these residue-nucleotide contacts are formed in both UCSF DOCK and GROMACS systems.

With only 13 intermolecular electrostatic interactions, the R78G mutant system (Table 3.6). has fewer than 18 contacts formed in the K75G mutant interaction. However, they do share a common characteristic–the presence of many arginine amino acids; nine out of 13 bonds are formed by arginine. Those with high occupancy are observed to be located near the C-terminal end of this particular peptide, consisting of Arg-77, Arg-78, Gly-78, and Arg-80. They make contacts with nucleotides near the target's end, such as C-26, G-27, and A-28. One exception is the existence of many residue-nucleotide pairs formed by Arg-68. It makes a total of four contacts, mainly near

the hairpin loop of TAR RNA. Unlike the K75G mutant system, the results generated by VMD in this system do not match with any hydrogen bonds found in UCSF DOCK.

Table 3.6 Possible residue-nucleotide contacts with hydrogen bonds formed between R78G mutant peptide and RNA TAR.

Peptide	RNA	Occupancy
Arginine-77	Adenine-28	95.76%
Arginine-80	Guanine-27	93.08%
Glycine-78	Cytosine-26	86.27%
Arginine-80	Cytosine-26	84.31%
Arginine-77	Guanine-27	75.49%
Proline-69	Cytosine-23	74.51%
Arginine-68	Uracil-12	50.00%
Proline-69	Adenine-13	37.25%
Arginine-68	Guanine-11	31.37%
Arginine-68	Uracil-20	27.45%
Arginine-68	Guanine-22	26.47%
Arginine-70	Guanine-11	13.73%

Note. These pairs were generated using the default parameters of VMD.

The K75G-R78G mutant peptide makes even fewer electrostatic bonds with TAR RNA than the other two systems (Table 3.7). It is clear that this interaction is dominated by contacts formed by arginine amino acids as well. However, there are only three pairs with an occupancy higher than 50%. They are Arg-68:U-24, Arg-73:U-7, and Gly-78:G-11 pairs. Interestingly, the table shows that most contacts are formed near the target 5'-3' ends, such as C-6, U-7, and G-9.

Table 3.7 Possible residue-nucleotide contacts with hydrogen bonds formed between K75G-R78G mutant peptide and RNA TAR.

Peptide	RNA	Occupancy
Arginine-68	**Uracil-24**	**93.35%**
Arginine-73	Uracil-7	91.92%
Glycine-78	Guanine-11	89.22%
Threonine-72	Guanine-9	49.02%
Arginine-73	**Cytosine-6**	**47.06%**
Glycine-71	Guanine-9	34.31%
Arginine-68	Cytosine-23	30.39%
Arginine-73	**Cytosine-23**	**28.43%**
Proline-69	**Cytosine-6**	**28.43%**
Arginine-68	Cytosine-6	10.78%

Note. These pairs were generated using the default parameters of VMD. Bold indicate these residue-nucleotide contacts are formed in both UCSF DOCK and GROMACS systems.

3.4.5 Absolute Binding Free Energies

After performing two different sets of GROMACS simulation for each system, the binding free energy can be calculated by adding all contributions by complex, ligand, and restraint using Equation 5. Note that the energy contribution by decoupling ligand from the complex during state F→D must be reversed. Based on the results in Table 3.8, the absolute binding free energies are -67.9, -69.3, -62.7, and -7.477 kcal/mol for K75G mutant, R78G mutant, K75G-R78G mutant, and wild-type, respectively.

Table 3.8 Absolute binding free energies of K75G mutant, R78G mutant, K75G-R78G mutant, and wild-type.

Systems	Complex (kcal/mol)	Ligand (kcal/mol)	Restraint (kcal/mol)	G-binding (kcal/mol)
K75G Mutant	406.536	333.125	5.455	-67.956
R78G Mutant	541.832	463.547	8.921	-69.344
K75G-R78G Mutant	501.199	429.433	9.012	-62.754
Wild-Type	406.296	391.913	6.906	-7.477

4 DISCUSSION

The TAR-Tar interaction has long been studied due to it being a prime target for the discovery of novel antiviral treatments.[6] This specific RNA-peptide complex plays a major role in the life cycle of HIV by up-regulating transcription. Unfortunately, many have tried and failed to synthesize molecules with satisfactory potency and selectivity for further commercial development. Based on previous studies done on Tat peptide and TAR RNA, we reasoned that mutating specific amino acids in the peptide would provide both the peptide and the complex more configurational entropy in order to see different interaction behaviors. This larger configurational entropy is introduced by an increase in local flexibility. Here, this project reports that three new mutant peptides bind to RNA targets with different behaviors and affinities.

The most surprising result was the observation that the binding of R78G mutant peptide to RNA target leads to a more stable interaction; this observation contradicts an earlier study done by the Lustig group.[13] Based on both grid energy and absolute binding free energy generated by UCSF DOCK and GROMACS (Table 3.8 and Table C.3), respectively, R78G mutant has the lowest binding energy. One of the major contributions to this low binding energy may come from the 3-nucleotide bulge observed in R78G mutant RNA secondary structure. As indicated by Weeks and Crothers, Lustig and coworkers, and Smith and coworkers, introducing a larger bulge than the 1-nucleotide bulge in the wild-type would lead to a tighter peptide binding.[5,10,11] A larger bulge size is believed to better provide a more suitable peptide binding conformation for TAR RNA, maybe due to a larger major groove or, possibly, an increase in local flexibility. Another

contribution that all three systems might have experienced during binding is an increase in conformational entropy induced by mutations in peptide.

Due to the glycine mutation at positions 75 and 78, believed to enclose the hinged region of BIV Tat, these single and K75G-R78G mutant systems evolve to have structural characteristics not found in the wild-type. It is clear from the literature that the binding of arginine-rich Tat to TAR would give rise to distinctive conformational changes that can be easily replicated by the binding of argininamide.[34,35] Hence, these mutated peptides are observed to bind to the target near its 5'-3' ends. Note that all TAR RNA structures in this current study do not possess a U10-A13-U21 base triple network, but do retain nucleotides essential for peptide binding, including G11-C25 and G14-C23 base pairs, and U10 bulge.[4] However, the absence of this base triple would have little to no effect on the TAR-Tat interaction as indicated by mutagenesis experiments done by Frankel.[12] Similar to the wild-type interaction, positively charged arginine amino acids still play an important role in stabilizing the binding, especially the contribution by Arg-68, Arg-70, Arg-73, and Arg-77. Ultimately, the change in the location of peptide binding for each mutant system leads to lower binding energies than that of the wild-type. No longer are these mutant peptides bound largely in the major groove; they are observed to bind near the 5' and 3' ends with the help of positively charged arginine amino acids, as indicated by hydrogen bond analysis in both UCSF DOCK and GROMACS.

It should be noted that this project proposes an alternative interaction scenario than what Nguyen proposed.[13] Here, it was believed that the K75G mutant interaction could actually enjoy some limited stabilization due to flexibility induced by glycine, resulting

in an increased stability as compared to both the R78G and K75G-R78G mutant systems. The contradiction could stem from the fact that Nguyen utilized a very coarse-grained methodology for calculation. It was a two-residue per move lattice approach, rather than the utilized all-atom approach used in this project. Furthermore, the outdated RNA-peptide potentials used in that project might not have accurately described the binding behavior of either RNA or peptide. Thus, these factors might offer an explanation for why the two results differ.

Although more advanced methods were utilized in this project, we might not accurately predict what actually occurs in *in vitro* experiments. The results obtained from this project heavily depend on the grid energy and the location of peptide binding, generated from coarse-grained modeling. First and foremost, the lower the grid energy, the more stable is an interaction. Hence, choosing the flexible DOCKed structures grid energy as one of the criteria does in fact shorten computational runtime significantly. The second assumption, arguably the most crucial in this project, relies on the location of peptide binding. Without establishing this assumption, thousands of mutant systems for each mutation have to be looked at, and possibly simulated by GROMACS. Based on previous studies done on both HIV and BIV TAR-Tat systems, alternative Tat peptide structures containing the arginine-rich RNA binding domain tend to bind to TAR RNA target near the major groove.[36,37] Therefore, assuming that K75G mutant, R78G mutant, and K75G-R78G mutant peptides would also bind to a location similar to the wild-type is reasonable. Applying these two assumptions results in only three mutant complexes out of over three million poses generated by UCSF DOCK that would undergo all-atom

modeling simulation. Unfortunately, this process might have skipped over some low-energy complexes that could very well depict what an actual interaction might look like. Furthermore, force fields involving RNA-protein interactions are not as useful as those associated exclusively with proteins.[15]

5 CONCLUSION

The project has used hierarchical approach, consisting of coarse-grained (UCSF DOCK) and all-atom modeling (GROMACS) techniques, to study how mutating amino acids at the hinge regions of BIV Tat peptide affect the TAR-Tat interaction. Both Python and Unix scripts were also developed to better automate the process and assist the analysis. From the coarse-grained, technique, we generated over 3 million structures of mutant complexes for the study, and from the all-atom modeling technique, we cumulatively used over 20 microseconds to simulate the binding of peptides to RNA targets. Both techniques have been described and analyzed.

Unique structural changes and binding free energies are observed in all mutant systems. From extending the structure to reversing the polarity of both terminal ends, it is clear that the glycine mutation at positions 75 and 78 suggests that the Tat peptide has the ability to adopt markedly different conformations than the wild-type peptide. Due to the structural changes in peptide, BIV TAR RNA also exhibits specific structural characteristics, including the lack of U10-A13-U24 base triple network and coaxial stem stack structure, and, potentially, the existence of new RNA bulges in the case of R78G mutant RNA. Hence, new possible hydrogen bonds at different locations are formed between these mutant peptides and new RNA targets. Interestingly, results from the absolute binding free energy technique proposes that the R78G mutant interaction is more stable than the other two, with the K75G-R78G mutant being the most unstable.

Finally, this work had provided the code and foundation for future studies to build up. Hopefully, *in vitro* and *in vivo* experiments can be carried out using the results obtained

from this project. Though peptide agents may not necessarily be themselves useful drug agents, other small molecules can be designed with similar features if these preliminary experimental studies are successful.

6 FUTURE STUDIES

The hierarchical approach utilized in this project can first be modified in order to generate RNA structures. One can further rigorously examine the set of 294 RNA targets addressing bulge and triple base network features. Any trends may suggest possible RNA stabilization in binding. This can further be explored by correlating UCSF DOCK and GROMACS configurations with the four different classes of 294 RNA targets.

Given the uncertainty of the quality of RNA-protein potentials outside the major groove and associated regions, one can alternatively fully characterize the network of hydrogen and van der Waals contacts in all regions.[15] This may help resolve the suspiciously large stabilization calculated for the mutants using the existing RNA-protein potentials. Additionally, one can assess the role of the relatively free 5' and 3' ends noted for the R78G and K75G-R78G mutants by docking to a TAR RNA fragment extended in that region. Lastly, one can further optimize the parameters used in GROMACS simulations. By extending the simulation time to one microsecond, more details in regard to the interaction might be revealed.

REFERENCES

1. HIV/AIDS: The Basics Understanding HIV/AIDS https://aidsinfo.nih.gov/understanding-hiv-aids/fact-sheets/19/45/hiv-aids--the-basics (accessed Jun 13, 2018).

2. Global HIV and AIDS statistics https://www.avert.org/global-hiv-and-aids-statistics (accessed Jun 13, 2018).

3. Shiels, M. S.; Engels, E. A. Evolving epidemiology of HIV-associated malignancies. *Curr. Opin. HIV* **2017**, *12*, 6-11.

4. Greenbaum, N. L. How Tat targets TAR: structure of the BIV peptide–RNA complex. *Structure* **1996**, *4*, 5-9.

5. Smith, C. A.; Crotty, S.; Harada, Y.; Frankel, A. D. Altering the context of an RNA bulge switches the binding specificities of two viral Tat proteins. *Biochemistry (N. Y.)* **1998**, *37*, 10808-10814.

6. Davidson, A.; Leeper, T. C.; Athanassiou, Z.; Patora-Komisarska, K.; Karn, J.; Robinson, J. A.; Varani, G. Simultaneous recognition of HIV-1 TAR RNA bulge and loop sequences by cyclic peptide mimics of Tat protein. *Proc. Nat. Acad. Sci.* **2009**, *106*, 11931-11936.

7. Chen, L.; Frankel, A. D. A peptide interaction in the major groove of RNA resembles protein interactions in the minor groove of DNA. *Proc. Nat. Acad. Sci.* **1995**, *92*, 5077-5081.

8. Hsieh, M.; Collins, E. D.; Blomquist, T.; Lustig, B. Flexibility of BIV TAR-Tat: models of peptide binding. *J. Biomol. Struct. Dyn.* **2002**, *20*, 243-251.

9. Tzeng, S.; Kalodimos, C. G. Protein activity regulation by conformational entropy. *Nature* **2012**, *488*, 236.

10. Weeks, K. M.; Crothers, D. M. RNA binding assays for Tat-derived peptides: implications for specificity. *Biochemistry (N. Y.)* **1992**, *31*, 10281-10287.

11. Lustig, B.; Bahar, I.; Jernigan, R. L. RNA bulge entropies in the unbound state correlate with peptide binding strengths for HIV-1 and BIV TAR RNA because of improved conformational access. *Nucleic Acids Res.* **1998**, *26*, 5212-5217.

12. Chen, L.; Frankel, A. D. An RNA-binding peptide from bovine immunodeficiency virus Tat protein recognizes an unusual RNA structure. *Biochemistry (N. Y.)* **1994**, *33*, 2708-2715.

13. Nguyen, L. T. *BIV TAR RNA binding glycine mutant Tat peptides: An integrated modeling and binding assay approach;* San José State University: 2015;

14. Shirts, M. R.; Chodera, J. D. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* **2008**, *129*, 124105.

15. Lang, P. T.; Brozell, S. R.; Mukherjee, S.; Pettersen, E. F.; Meng, E. C.; Thomas, V.; Rizzo, R. C.; Case, D. A.; James, T. L.; Kuntz, I. D. DOCK 6: Combining techniques to model RNA–small molecule complexes. *RNA* **2009**, *15*, 1219-1230.

16. Mukherjee, S.; Balius, T. E.; Rizzo, R. C. Docking validation resources: protein family and ligand flexibility experiments. *J. Chem. Info. Model.* **2010**, *50*, 1986-2000.

17. Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performace molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.

18. Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1*, 19-25.

19. Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E. UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **2004**, *25*, 1605-1612.

20. Humphrey, W.; Dalke, A.; Schulten, K. VMD: visual molecular dynamics. *J. Mol. Graph.* **1996**, *14*, 33-38.

21. Parisien, M.; Major, F. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* **2008**, *452*, 51.

22. Hoksza, D.; Svozil, D. Efficient RNA pairwise structure comparison by SETTER method. *Bioinformatics* **2012**, *28*, 1858-1864.

23. Čech, P.; Svozil, D.; Hoksza, D. SETTER: web server for RNA structure comparison. *Nucleic Acids Res.* **2012**, *40*, W48.

24. Hoksza, D.; Svozil, D. Multiple 3D RNA structure superposition using neighbor joining. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2015**, *12*, 520-530.

25. Čech, P.; Hoksza, D.; Svozil, D. MultiSETTER: web server for multiple RNA structure comparison. *BMC Bioinformatics* **2015**, *16*, 253.

26. O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**, *3*, 33.

27. Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696-3713.

28. Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An N· log (N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089-10092.

29. Hess, B.; Bekker, H.; Berendsen, H. J.; Fraaije, J. G. LINCS: a linear constraint solver for molecular simulations. *J. Comput. Chem*. **1997**, *18*, 1463-1472.

30. Parrinello, M.; Rahman, A. Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.* **1981**, *52*, 7182-7190.

31. Boresch, S.; Tettinger, F.; Leitgeb, M.; Karplus, M. Absolute binding free energies: a quantitative approach for their calculation. *J. Phys. Chem. B* **2003**, *107*, 9535-9551.

32. Davies, H.; Bignell, G. R.; Cox, C.; Stephens, P.; Edkins, S.; Clegg, S.; Teague, J.; Woffendin, H.; Garnett, M. J.; Bottomley, W. Mutations of the BRAF gene in human cancer. *Nature* **2002**, *417*, 949.

33. Schauperl, M.; Podewitz, M.; Waldner, B. J.; Liedl, K. R. Enthalpic and entropic contributions to hydrophobicity. *J. Chem. Theory Comput*. **2016**, *12*, 4600-4610.

34. Puglisi, J. D.; Tan, R.; Calnan, B. J.; Frankel, A. D. Conformation of the TAR RNA-arginine complex by NMR spectroscopy. *Science* **1992**, *257*, 76-80.

35. Aboul-ela, F.; Karn, J.; Varani, G. The structure of the human immunodeficiency virus type-1 TAR RNA reveals principles of RNA recognition by Tat protein. *J. Mol. Biol.* **1995**, *253*, 313-332.

36. Hamy, F.; Felder, E. R.; Heizmann, G.; Lazdins, J.; Aboul-Ela, F.; Varani, G.; Karn, J.; Klimkait, T. An inhibitor of the Tat/TAR RNA interaction that effectively suppresses HIV-1 replication. *Proc. Nat. Acad. Sci.* **1997**, *94*, 3548-3553.

37. Huq, I.; Ping, Y.; Tamilarasu, N.; Rana, T. M. Controlling human immunodeficiency virus type 1 gene expression by unnatural peptides. *Biochemistry (N. Y.)* **1999**, *38*, 5172-5177.

APPENDICES

Appendix A Read Me for UCSF DOCK

Manual Protocol

1) Please go to http://ringo.ams.sunysb.edu/index.php/2015_DOCK_tutorial_with_Poly(ADP-ribose)_polymerase_(PARP)#III._Generating_Receptor_Surface_and_Spheres for a tutorial on how to use UCSF Dock.
2) Create 6 folders in UCSF Dock folder and name them: 00.files, 01.dockprep, 02.surface-spheres, 03.box-grid, and 04.dock.
3) Obtain 1MNB NMR-PDB file from protein data bank (https://www.rcsb.org/structure/1MNB) and place it in 00.file folder.
4) Create the "dockprep.mol2" file: open 1mnb.pdb in UCSF Chimera and go to "Tools-Surface/Binding Analysis-Dock Prep". Use all of the default setting and select AMBER ff14SB charge model. Save the final complex in 01.dockprep folder.
5) Create the "ligand.mol2": open "dockprep.mol2" in UCSF Chimera and go to "Select-Chain-B". Delete the receptor by go to "Actions-Atoms/Bonds-Delete". Save this file in 01.dockprep folder.
6) Create the "receptor_noH.mol2": open "dockprep.mol2" in UCSF Chimera and go to "Select-Chain-A". Delete the ligand by go to "Actions-Atoms/Bonds-Delete". From here, go to "Select-Chemistry-Element-H" and delete all the hydrogen atoms. Save this file in 01.dockprep folder.
7) Generate the receptor surface: open "receptor_noH.pdb" in UCSF Chimera and go to "Action-Surface-Show". Then go to "Tools-Structure Editing-Write DMS" and save the files as "receptor.dms" in 02.surface-spheres.
Starting from here, everything is done using terminal.
8) Placing spheres: create an input file name INSPH by inputting this command "vim INSPH" in terminal. Within it, write:

- receptor.dms
- R
- X
- 0.0
- 4.0
- 1.4
- receptor.sph

9) Run the program using the command: sphgen -i INSPH -o OUTSPH
10) Visualization of the spheres:

1. Run the program using the command: showsphere
2. Answer the following questions:
    a. Enter the name of sphere cluster file: receptor.sph
    b. Enter cluster number to process (<0 = all): -1
    c. Enter name for output file prefix: output_spheres
    d. Generate surface as well as pdb files (N>/Y)?: N

3. Open receptor_noH.pdb in UCSF Chimera and output_spheres.pdb

11) Select spheres of interest: run the following command: sphere_selector receptor.sph …/01.dockprep/ligand.mol2 8.0. A new selected_spheres.sph is generated.

1. Run the program using the command: showsphere
2. Answer the following questions:
    a. Enter the name of sphere cluster file: receptor.sph
    b. Enter cluster number to process (<0 = all): -1
    c. Enter name for output file prefix: output_spheres
    d. Generate surface as well as pdb files (N>/Y)?: N
3. Open receptor_noH.pdb in UCSF Chimera and output_spheres_selected.pdb. Go to "Select-Residue-SPH" and "Actions-Atoms/Bonds-Sphere".

12) Generate a box: go to 03.box-grid and run the following command: vim showbox.in. Within in, write:

- Y
- 8.0
- …/02.surface-spheres/selected_spheres.sph
- 1
- box.pdb

12.1) Save the file by doing the following: press ESC button and write :wq

12.2) Run the following command: showbox < showbox.in

13) Compute the potential energy in docking region: go to 03.box-grid and write the following command: vi grid.in. Within it, answer all the questions. Run the program by issuing the following command: grid -i grid.in

Parameters and values of Grid Computing

Parameter	Value
Compute_grids	Yes
Grid_spacing	0.4
Output_molecule	No
Contact_score	No
Energy_score	Yes
Energy_cutoff_distance	9999
Atom_model	A
Attractive_exponent	6
Repulsive_exponent	9
Distance_dielectric	Yes
Dielectric_factor	4
Bump_filter	Yes
Bump_overlap	0.75
Receptor_file	…/receptor.mol2
Box_file	Box.pdb
Vdw_definition_file	…/vdw_AMBER_parm99.defn
Score_grid_prefix	grid

14) Perform an energy minimization: go to 04.dock and write the following command: touch min.in. Within it, answer all the questions. Run the program by issuing the following command: dock6 -i min.in

Parameter and Value of Minimization Docking method.

Parameter	Value
Ligand_atom_file	…/ligand.mol2
Limit_max_ligands	No
Skip_molecule	No
Read_mol_solvation	No
Calculate_rmsd	Yes
Use_rmsd_references_mol	No
Use_database_filter	No
Orient_ligand	No
Use_internal_energy	Yes
Internal_energy_rep_exp	12
Flexible_ligand	No
Bump_filter	No
Score_molecules	Yes
Contact_score_primary	No
Contact_score_secondary	No
Grid_score_primary	Yes
Grid_score_secondary	No
Grid_score_rep_rad_scale	1
Grid_score_vdw_scale	1
Grid_score_es_scale	1
Grid_score_grid_prefix	…/grid
Multigrid_score_secondary	No
Dock3.5_score_secondary	No
Continuous_score_secondary	No
Descriptor_score_secondary	No
Gbsa_zou_score_secondary	No
Gbsa_hawkins_score_secondary	No
SASA_descriptor_score_secondary	No
Amber_score_secondary	No
Minimize_ligand	Yes
Simplex_max_iterations	1000
Simplex_tors_premin_iterations	0
Simplex_max_cycle	1
Simplex_score_converge	0.1
Simplex_cycle_converge	1.0

Simplex_trans_step	1.0
Simplex_rot_step	0.1
Simplex_tors_step	10.0
Simplex_random_seed	0
Simplex_restraint_min	Yes
Simplex_coefficient_restraint	10.0
Atom_model	All
Vdw_defn_file	…/vdw_AMBER_parm99.defn
Flex_drive_file	…/flex.defn
Flex_drive_file	…/flex_drive.tbl
Ligand_outfile_prefix	ligand.min
Write_orientations	No
Num_scored_conformers	1
Rank_ligands	no

15) Perform Rigid Docking: go to 04.dock and write the following command: touch rigid.in. Within it, answer all the questions. Run the program by issuing the following command: dock6 -i rigid.in

Parameters and Values of Rigid Docking method

Parameter	Value
Ligand_atom_file	ligand.mol2
Limit_max_ligands	No
Skip_molecule	No
Read_mol_solvation	No
Calculate_rmsd	Yes
Use_rmsd_reference_mol	No
Use_database_filter	No
Orient_ligand	Yes
Automated_matching	Yes
Receptor_site_file	…/selected_spheres.sph
Max_orientations	1000
Critical_points	No
Chemical_matching	No
Use_ligand_spheres	No
Use_internal_energy	Yes
Internal_energy_rep_exp	12
Flexible_ligand	No
Bump_filter	No
Score_molecules	Yes
Contact_score_secondary	no
Contact_score_secondary	No

Grid_score_primary	Yes
Grid_score_secondary	No
Grid_score_rep_rad_scale	1
Grid_score_vdw_scale	1
Grid_score_es_scale	1
Grid_score_grid_prefix	…/grid
Multigrid_score_secondary	No
Dock3.5_score_secondary	No
Continuous_score_secondary	No
Descriptor_score_secondary	No
Gbsa_zou_score_secondary	No
Gbsa_hawkins_score_secondary	No
SASA_descriptor_score_secondary	No
Amber_score_secdonary	No
Minimize_ligand	Yes
Simplex_max_iterations	1000
Simplex_tors_premin_iterations	0
Simplex_max_cycles	1
Simplex_score_converge	0.1
Simplex_cycle_converge	1.0
Simplex_trans_step	1.0
Simplex_rot_step	0.1
Simplex_tors_step	10.0
Simplex_random_seed	0
Simplex_restraint_min	No
Atom_model	All
Vdw_defn_file	…/vdw_AMBER_parm99.defn
Flex_defn_fiel	…/flex.defn
Flex_drive_file	…/flex_drive.tbl
Ligand_outfile_prefix	ligand.rgd
Write_orientations	No
Num_scored_conformers	5000
Write_conformations	No
Cluster_conformations	Yes
Cluster_rmsd_threshold	2.0
Rank_ligands	no

16) Perform Flexible Docking: go to 04.dock and write the following command: touch flex.in. Within it, answer all the questions. Run the program by issuing the following command: dock6 -i flex.in

Parameter and Value of Flexible Docking method

Parameter	Value
Ligand_atom_file	…/ligand.mol2
Limit_max_ligands	No
Skip_molecule	No
Read_mol_solvation	No
Calculate_rmsd	Yes
Use_rmsd_reference_mol	No
Use_database_filter	No
Orient_ligand	Yes
Automated_matching	Yes
Receptor_site_file	…/selected_spheres.sph
Max_orientations	1000
Critical_points	No
Chemical_matching	No
Use_ligand_spheres	No
Use_internal_energy	Yes
Internal_energy_rep_exp	12
Flexbile_ligand	Yes
User_specified_anchor	No
Limit_max_anchors	No
Min_anchor_size	5
Pruning_use_clustering	Yes
Pruning_max_orients	1000
Pruning_clustering_cutoff	100
Pruning_conformer_score_cutoff	100
Use_clash_overlap	No
Write_growth_tree	No
Bump_filter	No
Score_molecules	Yes
Contact_score_primary	No
Contact_score_secondary	No
Grid_score_primary	Yes
Grid_score_secondary	No
Grid_score_rep_rad_scale	1
Grid_score_vdw_scale	1
Grid_score_es_scale	1
Grid_score_grid_prefix	…/grid
Multigrid_score_secondary	No
Dock3.5_score_secondary	No
Continuous_score_secondary	No
Descriptor_score_secondary	No

Gbsa_zou_score_secondary	No
Gbsa_hawkins_score_secondary	No
SASA_descriptor_score_secondary	No
Amber_score_secondary	No
Minimize_ligand	Yes
Minimize_anchor	Yes
Minimize_flexible_growth	Yes
Use_advanced_simplex_parameters	No
Simplex_max_cycle	1
Simplex_score_converge	0.1
Simplex_cycle_converge	1.0
Simplex_trans_step	1.0
Simplex_rot_step	0.1
Simplex_tors_step	10.0
Simplex_anchor_max_iterations	500
Simplex_grow_max_iterations	500
Simplex_grow_tors_premin_iterations	0
Simplex_random_seed	0
Simplex_restraint_min	No
Atom_model	All
Vdw_defn_file	…/vdw_AMBER_parm99.defn
Flex_defn_file	…/flex.defn
Flex_drive_file	…/flex_drive.tbl
Ligand_outfile_prefix	ligand.flx
Write_orientations	No
Num_scored_conformers	5000
Write_conformations	No
Cluster_conformations	Yes
Cluster_rmsd_threshold	2.0
Rank_ligands	no

Automation Protocol

These scripts were written by Thanh Le and Arjit Misra. Arjit Misra has given his permission to publish these scripts.
Please note that these scripts are mainly used on an HPC.
Please also note, you will need to change the paths and environment.
Please also note, you will need to rename all the files.

Step 1: Use the code below to get the peptide and RNA close to each other.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
import os
import chimera
from chimera import runCommand as rc

from chimera import openModels, Molecule

ligandFile="Protein.lig.mol2"
for receptorFile in os.listdir("E:/BIV/Sept/receptors"):

    rc("open "+ligandFile)
    rc("open E:/BIV/Sept/receptors/"+receptorFile)

    mols = openModels.list(modelTypes=[Molecule])

    chains = [m.sequence('A') for m in mols]


    from StructSeqAlign import makeAlignment

    #from chimera import preferences
    mav = makeAlignment(chains, iterate=True, cutoff=500.0, iterLimit=None)

    rc("select#0")
    rc("delete selected")


    rc("write 1 E:/BIV/Sept/new_"+receptorFile)
    rc("close session")
```

Step 2: Use the code below to prepare the system.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# dock_rigid6.8.py

#System Requirements: UCSF Chimera, Linux/Mac OS (or Cygwin).
#Python script to prepare molecule for docking
##Windows: Chimera->File->Open->chimera_prep.py [Will not run as intended on
Windows. Command-line instructions not received thru os.system]
###Linux/Mac: chimera --nogui --silent --script chimera_prep.py
```

```
import os
import chimera
from chimera import runCommand as rc
from DockPrep import prep
from chimera import openModels
from chimera import MSMSModel

'''_____NOTE: Review and change directories
below_____'''

#ligandFile="1mnbpeptide75gly.pdb"
ligandFile="mutantpeptide.pdb"
nativeDirectory="/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant/"

'''_____End of
directories_____'''

if nativeDirectory.endswith("/")==False:
        nativeDirectory=nativeDirectory+"/"

#print(os.listdir(nativeDirectory));

for input_File in os.walk(nativeDirectory).next()[2]:

        print(input_File)

        if (input_File.endswith(".pdb")==False):
          continue;
#       if ("90" in input_File):
#         continue;
        if input_File==ligandFile:
          continue;
        structure_name=input_File.replace(".pdb","")
        os.chdir(nativeDirectory)
        dirList=os.walk('.').next()[1]

        print("**********"+structure_name+"**********"
)

        if (structure_name not in dirList):
          os.mkdir(structure_name)
          os.chdir(structure_name)
        else:
          os.chdir(structure_name)
```

```
directory = nativeDirectory+structure_name+"/";
dirList=os.listdir(directory)

if ('00.files' not in dirList):
  os.mkdir('00.files')
if '01.dockprep' not in dirList:
  os.mkdir('01.dockprep')
if '02.surface-spheres' not in dirList:
  os.mkdir('02.surface-spheres')
if '03.box-grid' not in dirList:
  os.mkdir('03.box-grid')
if '04.dock' not in dirList:
  os.mkdir('04.dock')
if '05.mini-virtual-screen' not in dirList:
  os.mkdir('05.mini-virtual-screen')
if '06.virtual-screen' not in dirList:
  os.mkdir('06.virtual-screen')



rc("open "+nativeDirectory+input_File)
rc("select:.a")
rc("select invert")
rc("delete selected")
rc("write 0 " + structure_name+".RNA.pdb")


models = chimera.openModels.list(modelTypes=[chimera.Molecule])
prep(models)

rc("addh")
rc("addcharge std") #chargeModel ff14SB by default

from WriteMol2 import writeMol2
writeMol2(models, directory+"01.dockprep/"+structure_name+".rec.mol2")

rc("delete element.H")
rc("write 0 " + directory+"01.dockprep/"+structure_name+".rec.noH.pdb")

if '90' in structure_name:
  rc("vdwdefine +0.01")
rc("surf") #generate surface using 'surf'
surf=openModels.list(modelTypes=[MSMSModel])[0] #get surf object
```

```
        from WriteDMS import writeDMS
        writeDMS(surf, directory+"02.surface-spheres/"+structure_name+".rec.dms")
        rc("close session")


os.chdir(nativeDirectory)
try:
        os.mkdir("Ligand")
        os.chdir("Ligand")
except OSError:
        os.chdir("Ligand")

rc("open "+nativeDirectory+ligandFile)
#Dock Prep Begins
models = chimera.openModels.list(modelTypes=[chimera.Molecule])
prep(models)

rc("addh")
rc("addcharge std") #chargeModel ff14SB by default
#Dock Prep Ends


from WriteMol2 import writeMol2
writeMol2(models, nativeDirectory + "Ligand/"+ligandFile[:-4]+".lig.mol2")

## -----

rc("close session")
```

Step 3: Use the code below to run rigid docking.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# dock_rigid6.8.py

#
#!!!DO NOT RUN THIS SCRIPT UNTIL you have run chimera_prep.py!!!
#System requirements: UCSF Dock 6.8, Sphgen 1.2 (recommended; see lines 17-19)
#Python script to perform rigid Protein-RNA docking with a
#given protein ligand and several RNA receptors (emulate MD)

#Usage: python2.7 dock_rigid6.8.py [ligand file] [receptor file]

import os
```

```
import sys

#This parameter should only be set as true if you have Sphgen_cpp 1.2 (or higher)
installed.
## Can be downloaded from
http://dock.compbio.ucsf.edu/Contributed_Code/sphgen_cpp.htm
sphgenNew=False;

'"_____NOTE: Review and change directories
below_____"'

dockDirectory="/home/blustig/perl5/dock/dock6+8/" #directory of dock installation, NO
BIN
sphgenNewDirectory="/home/blustig/perl5/dock/dock6+8/bin/sphgen_cpp.1.2/" #If
installed
nativeDirectory="/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant"

'"_____End of
directories_____"'


if (dockDirectory.endswith("/")==False):
        dockDirectory=dockDirectory+"/"
if (sphgenNewDirectory.endswith("/")==False):
        sphgenNewDirectory=sphgenNewDirectory+"/"
if nativeDirectory.endswith("/")==False:
        nativeDirectory=nativeDirectory+"/"

ligandFile = sys.argv[1]
if ligandFile not in os.listdir(nativeDirectory+"Ligand/"):
        print("Ligand file not found")
        exit()

receptorFile = sys.argv[2]
if receptorFile not in os.listdir(nativeDirectory):
        print("Receptor file not found")
        exit()

#output_File=open(nativeDirectory+"dockLog.out","wb")
#for input_File in os.listdir(nativeDirectory):
input_File=receptorFile

if (input_File.endswith(".pdb")==False):
        print("Receptor file must end with .pdb")
```

```
	exit();#continue;

structure_name=input_File.replace(".pdb","")
os.chdir(nativeDirectory)
dirList=os.listdir(".")

print("**********"+structure_name+"**********")

try:
	os.chdir(structure_name)
except OSError:
	output_File.write(structure_name+" folder not found. Did you run
chimera_prep.py?")
	exit();#continue

directory = nativeDirectory+structure_name+"/";


dirList=os.listdir(directory+"02.surface-spheres")
os.chdir(directory+"02.surface-spheres")
if "INSPH" in dirList:
	os.remove("INSPH")
f=open("INSPH", "w");
f.write(structure_name+'.rec.dms\n')
f.write('R\n')
f.write('X\n')
f.write('0.0\n')
f.write('4.0\n')
f.write('1.4\n')
f.write(structure_name+'.rec.sph')
f.close()

os.chdir(directory+"02.surface-spheres")
if sphgenNew:
	os.system(sphgenNewDirectory+"sphgen_cpp") #using new version of sphgen
(sphgen 1.2)
else:
	os.system(dockDirectory+"bin/sphgen -i INSPH -o OUTSPH") #using old sphgen
version in dock6

dirList=os.listdir(".");
if "input.in" in dirList:
	os.remove("input.in")
si=open("input.in", "wb");
```

```
si.write(structure_name+'.rec.sph\n')
si.write('-1\n')
si.write('N\n')
si.write('output_spheres\n')
si.write('N\n')
si.close();

os.system(dockDirectory + "bin/showsphere < input.in")



os.chdir(directory+"02.surface-spheres/")
os.system(dockDirectory + "bin/sphere_selector "+ nativeDirectory + structure_name
+"/02.surface-spheres/" + structure_name + ".rec.sph
"+nativeDirectory+"Ligand/"+ligandFile+" 8.0")

dirList=os.listdir(".");
if "input2.in" in dirList:
        os.remove("input2.in")
si=open(directory+"02.surface-spheres/input2.in", "wb");
si.write('selected_spheres.sph\n')
si.write('-1\n')
si.write('N\n')
si.write('output_spheres_selected\n')
si.write('N\n')
si.close();

os.system(dockDirectory + "bin/showsphere < input2.in")
#NOTE: Saves as output_spheres_selected_1

dirList=os.listdir(".");
for doc in dirList:
        if "output_spheres_selected" in doc:
          os.rename(doc, "output_spheres_selected.pdb");

os.chdir("../03.box-grid/");
dirList=os.listdir(".");
if "showbox.in" in dirList:
        os.remove("showbox.in")
si=open("showbox.in", "wb");
si.write('Y\n')
si.write('8.0\n')
si.write('../02.surface-spheres/selected_spheres.sph\n')
si.write('1\n')
```

```
si.write(structure_name+'.box.pdb\n')
si.close();

os.system(dockDirectory + "bin/showbox < showbox.in")

dirList=os.listdir(".");
if "grid.in" in dirList:
        os.remove("grid.in")
gi=open("grid.in", "wb");
gi.write('grid\n')
gi.write('compute_grids\tyes\n')
gi.write('grid_spacing\t0.4\n')
gi.write('output_molecule\tno\n')
gi.write('contact_score\tno\n')
gi.write('energy_score\tyes\n')
gi.write('energy_cutoff_distance\t9999\n')
gi.write('atom_model\ta\n')
gi.write('attractive_exponent\t6\n')
gi.write('repulsive_exponent\t9\n')
gi.write('distance_dielectric\tyes\n')
gi.write('dielectric_factor\t4\n')
gi.write('bump_filter\tyes\n')
gi.write('bump_overlap\t0.75\n')
gi.write('receptor_file\t../../'+structure_name+'/01.dockprep/'+structure_name+'.rec.mol2\
n')
gi.write('\nbox_file\t'+structure_name+'.box.pdb\n')
gi.write('\nvdw_definition_file\t'+dockDirectory+'parameters/vdw_AMBER_parm99.def
n\n')
gi.write('\nscore_grid_prefix\tgrid')
gi.close();

os.system(dockDirectory + "bin/grid -i grid.in")

os.chdir("../04.dock")


#Energy Minimization
dirList=os.listdir(".")
if "min.in" in dirList:
        os.remove("min.in")
minimize=open('min.in','w')
minimize.write("conformer_search_type                    rigid\n")
minimize.write("use_internal_energy                      yes\n")
minimize.write("internal_energy_rep_exp                  12\n")
```

```
minimize.write("internal_energy_cutoff 100.0\n")
minimize.write("ligand_atom_file\t"+nativeDirectory+"Ligand/"+ligandFile+"\n")
minimize.write("limit_max_ligands no\n")
minimize.write("skip_molecule no\n")
minimize.write("read_mol_solvation no\n")
minimize.write("calculate_rmsd yes\n")
minimize.write("use_rmsd_reference_mol yes\n")
minimize.write("rmsd_reference_filename\t"+nativeDirectory+"Ligand/"+ligandFile+"\n
")
minimize.write("use_database_filter no\n")
minimize.write("orient_ligand no\n")
minimize.write("bump_filter no\n")
minimize.write("score_molecules yes\n")
minimize.write("contact_score_primary no\n")
minimize.write("contact_score_secondary no\n")
minimize.write("grid_score_primary yes\n")
minimize.write("grid_score_secondary no\n")
minimize.write("grid_score_rep_rad_scale 1\n")
minimize.write("grid_score_vdw_scale 1\n")
minimize.write("grid_score_es_scale 1\n")
minimize.write("grid_score_grid_prefix ../03.box-grid/grid\n")
minimize.write("multigrid_score_secondary no\n")
minimize.write("dock3.5_score_secondary no\n")
minimize.write("continuous_score_secondary no\n")
minimize.write("footprint_similarity_score_secondary no\n")
minimize.write("pharmacophore_score_secondary no\n")
minimize.write("descriptor_score_secondary no\n")
minimize.write("gbsa_zou_score_secondary no\n")
minimize.write("gbsa_hawkins_score_secondary no\n")
minimize.write("SASA_score_secondary no\n")
minimize.write("amber_score_secondary no\n")
minimize.write("minimize_ligand yes\n")
minimize.write("simplex_max_iterations 1000\n")
minimize.write("simplex_tors_premin_iterations 0\n")
minimize.write("simplex_max_cycles 1\n")
minimize.write("simplex_score_converge 0.1\n")
minimize.write("simplex_cycle_converge 1.0\n")
minimize.write("simplex_trans_step 1.0\n")
minimize.write("simplex_rot_step 0.1\n")
minimize.write("simplex_tors_step 10.0\n")
minimize.write("simplex_random_seed 0\n")
minimize.write("simplex_restraint_min yes\n")
minimize.write("simplex_coefficient_restraint 10.0\n")
minimize.write("atom_model all\n")
```

```
minimize.write("vdw_defn_file\t" + dockDirectory +
"parameters/vdw_AMBER_parm99.defn\n")
minimize.write("flex_defn_file\t"+ dockDirectory + "parameters/flex.defn\n")
minimize.write("flex_drive_file\t" + dockDirectory + "parameters/flex_drive.tbl\n")
minimize.write("ligand_outfile_prefix\t"+structure_name+".min\n")
minimize.write("write_orientations                    no\n")
minimize.write("num_scored_conformers                 1\n")
minimize.write("rank_ligands                        no")
minimize.close()
os.system(dockDirectory + "bin/dock6 -i min.in -v -o minOutput.out")


#Footprint
dirList=os.listdir(".")
if "footprint.in" in dirList:
        os.remove("footprint.in")
ftprint=open('footprint.in','w')
ftprint.write("conformer_search_type                  rigid\n")
ftprint.write("use_internal_energy                   no\n")
ftprint.write("ligand_atom_file\t"+nativeDirectory+"Ligand/"+ligandFile+"\n")
ftprint.write("limit_max_ligands                     no\n")
ftprint.write("skip_molecule                        no\n")
ftprint.write("read_mol_solvation                    no\n")
ftprint.write("calculate_rmsd                       no\n")
ftprint.write("use_database_filter                   no\n")
ftprint.write("orient_ligand                        no\n")
ftprint.write("bump_filter                          no\n")
ftprint.write("score_molecules                       yes\n")
ftprint.write("contact_score_primary                  no\n")
ftprint.write("contact_score_secondary                 no\n")
ftprint.write("grid_score_primary                    no\n")
ftprint.write("grid_score_secondary                   no\n")
ftprint.write("multigrid_score_primary                 no\n")
ftprint.write("multigrid_score_secondary                no\n")
ftprint.write("dock3.5_score_primary                   no\n")
ftprint.write("dock3.5_score_secondary                  no\n")
ftprint.write("continuous_score_primary                 no\n")
ftprint.write("continuous_score_secondary                no\n")
ftprint.write("footprint_similarity_score_primary          yes\n")
ftprint.write("footprint_similarity_score_secondary         no\n")
ftprint.write("fps_score_use_footprint_reference_mol2          yes\n")
ftprint.write("fps_score_footprint_reference_mol2_filename\t"+nativeDirectory+"Ligand
/"+ligandFile+"\n")
ftprint.write("fps_score_foot_compare_type              Euclidean\n")
```

```
ftprint.write("fps_score_normalize_foot                    no\n")
ftprint.write("fps_score_foot_comp_all_residue                yes\n")
ftprint.write("fps_score_receptor_filename\t"+directory+"01.dockprep/"+structure_name
+".rec.mol2")
ftprint.write("fps_score_vdw_att_exp                    6\n")
ftprint.write("fps_score_vdw_rep_exp                    12\n")
ftprint.write("fps_score_vdw_rep_rad_scale                1\n")
ftprint.write("fps_score_use_distance_dependent_dielectric            yes\n")
ftprint.write("fps_score_dielectric                4.0\n")
ftprint.write("fps_score_vdw_fp_scale                1\n")
ftprint.write("fps_score_es_fp_scale                1\n")
ftprint.write("fps_score_hb_fp_scale                0\n")
ftprint.write("pharmacophore_score_secondary                no\n")
ftprint.write("descriptor_score_secondary                no\n")
ftprint.write("gbsa_zou_score_secondary                no\n")
ftprint.write("gbsa_hawkins_score_secondary                no\n")
ftprint.write("SASA_score_secondary                no\n")
ftprint.write("amber_score_secondary                no\n")
ftprint.write("minimize_ligand                no\n")
ftprint.write("atom_model                all\n")
ftprint.write("vdw_defn_file
"+dockDirectory+"parameters/vdw_AMBER_parm99.defn\n")
ftprint.write("flex_defn_file
"+dockDirectory+"parameters/flex.defn\n")
ftprint.write("flex_drive_file
"+dockDirectory+"parameters/flex_drive.tbl\n")
ftprint.write("ligand_outfile_prefix
"+structure_name+".fps_min\n")
ftprint.write("write_footprints                yes\n")
ftprint.write("write_hbonds                yes\n")
ftprint.write("write_orientations                no\n")
ftprint.write("num_scored_conformers                1\n")
ftprint.write("rank_ligands                no")
ftprint.close()
os.system(dockDirectory + "bin/dock6 -i footprint.in -v -o footprintOutput.out")

#Rigid Docking
dirList=os.listdir(".")
if "rgd.in" in dirList:
        os.remove("rgd.in")
rigidDock=open("rgd.in","w");
rigidDock.write("conformer_search_type                rigid\n")
rigidDock.write("use_internal_energy                yes\n")
rigidDock.write("internal_energy_rep_exp                12\n")
```

```
rigidDock.write("internal_energy_cutoff 100.0\n")
rigidDock.write("ligand_atom_file\t"+nativeDirectory+"Ligand/"+ligandFile+"\n")
rigidDock.write("limit_max_ligands\tno\n")
rigidDock.write("skip_molecule\tno\n")
rigidDock.write("read_mol_solvation\tno\n")
rigidDock.write("calculate_rmsd\tyes\n")
rigidDock.write("use_rmsd_reference_mol\tno\n")
rigidDock.write("use_database_filter\tno\n")
rigidDock.write("orient_ligand\tyes\n")
rigidDock.write("automated_matching\tyes\n")
rigidDock.write("receptor_site_file\t"+directory+"02.surface-
spheres/selected_spheres.sph\n")
rigidDock.write("max_orientations\t1000\n")
rigidDock.write("critical_points\tno\n")
rigidDock.write("chemical_matching\tno\n")
rigidDock.write("use_ligand_spheres\tno\n")
rigidDock.write("use_internal_energy\tyes\n")
rigidDock.write("internal_energy_rep_exp\t12\n")
rigidDock.write("flexible_ligand\tno\n")
rigidDock.write("bump_filter\tno\n")
rigidDock.write("score_molecules\tyes\n")
rigidDock.write("contact_score_primary\tno\n")
rigidDock.write("contact_score_secondary\tno\n")
rigidDock.write("grid_score_primary\tyes\n")
rigidDock.write("grid_score_secondary\tno\n")
rigidDock.write("grid_score_rep_rad_scale\t1\n")
rigidDock.write("grid_score_vdw_scale\t1\n")
rigidDock.write("grid_score_es_scale\t1\n")
rigidDock.write("grid_score_grid_prefix\t"+directory+"03.box-grid/grid\n")
rigidDock.write("multigrid_score_secondary\tno\n")
rigidDock.write("dock3.5_score_secondary\tno\n")
rigidDock.write("continuous_score_secondary\tno\n")
rigidDock.write("descriptor_score_secondary\tno\n")
rigidDock.write("gbsa_zou_score_secondary\tno\n")
rigidDock.write("gbsa_hawkins_score_secondary\tno\n")
rigidDock.write("SASA_descriptor_score_secondary\tno\n")
rigidDock.write("amber_score_secondary\tno\n")
rigidDock.write("minimize_ligand\tyes\n")
rigidDock.write("simplex_max_iterations\t1000\n")
rigidDock.write("simplex_tors_premin_iterations\t0\n")
rigidDock.write("simplex_max_cycles\t1\n")
rigidDock.write("simplex_score_converge\t0.1\n")
rigidDock.write("simplex_cycle_converge\t1.0\n")
rigidDock.write("simplex_trans_step\t1.0\n")
```

```
rigidDock.write("simplex_rot_step\t0.1\n")
rigidDock.write("simplex_tors_step\t10.0\n")
rigidDock.write("simplex_random_seed\t0\n")
rigidDock.write("simplex_restraint_min\tno\n")
rigidDock.write("atom_model\tall\n")
rigidDock.write("vdw_defn_file\t" + dockDirectory +
"parameters/vdw_AMBER_parm99.defn\n")
rigidDock.write("flex_defn_file\t"+ dockDirectory + "parameters/flex.defn\n")
rigidDock.write("flex_drive_file\t" + dockDirectory + "parameters/flex_drive.tbl\n")
rigidDock.write("ligand_outfile_prefix\t"+structure_name+".rgd\n")
rigidDock.write("write_orientations\tno\n")
rigidDock.write("num_scored_conformers\t5000\n")
rigidDock.write("write_conformations\tno\n")
rigidDock.write("cluster_conformations\tyes\n")
rigidDock.write("cluster_rmsd_threshold\t2.0\n")
rigidDock.write("rank_ligands\tno\n")
rigidDock.close()

os.system(dockDirectory + "bin/dock6 -i rgd.in -v -o dockOutput.out")
'''
with open("dockOutput.out") as dockOut:
        text=dockOut.read().replace("\n","")
if "Total elapsed time" in text:
        output_File.write(structure_name+" docked successfully\n")
else:
        output_File.write(structure_name+" was not docked. See dockOutput file\n")
'''
print(structure_name+" is done")


#output_File.close()
```

Step 4 (Optional): if one desires to remove poses with high clashes, run this code below.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#

import chimera
from chimera import runCommand as rc
import os
```

```
directoryOfStructures="/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant"

if directoryOfStructures.endswith('/')==False:
	directoryOfStructures=directoryOfStructures+'/'
os.chdir(directoryOfStructures)

for structure in os.walk('.').next()[1]:
	if '04.dock' not in os.walk(structure).next()[1]:
	  continue
	dockPlace=directoryOfStructures+structure+'/'
	receptorFile='01.dockprep/'+structure+'.rec.noH.pdb'
	ligandFile='04.dock/'+structure+'.rgd_scored.mol2'
	selectedFile=open(dockPlace+ligandFile.replace('scored','selected150'),'w')
	modelNum=0
	i=0
	scored=open(dockPlace+ligandFile,'r').readlines()
	while ("Name:" not in scored[i]):
	  i=i+1
	j=i

	rc('open '+dockPlace+receptorFile)
	rc('open '+dockPlace+ligandFile)
	while (j<len(scored)):
	  j=i+1
	  while(j<len(scored) and ("Name:" not in scored[j])):
		j=j+1

	  modelNum=modelNum+1;

	  rc('findclash #1.'+str(modelNum)+' test :.a savefile
'+os.getcwd()+'/clashes'+structure+'.txt')
	  with open('clashes'+structure+'.txt','r') as logFile:
		contacts=int(logFile.readlines()[6].split()[0])
		logFile.close()
	  if contacts<150:
		for k in range(i,j):
			selectedFile.write(scored[k])
	  i=j;
	  print("1."+str(modelNum)+"..."+str(contacts))

	  if(j==len(scored)):
		break
	rc('close session')
```

```
    selectedFile.close()
```

Step 5: Use the code below to run flexible docking.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

#Flexible docking with UCSF Dock 6.8 !!!!Will not work with 6.7 or previous!!!
##If you have changed the clashes cutoff and ran the script, edit line 39 to reflect that
change.
##Also ensure that the file actually exists!

#Flexible docking output is in the 04.dock/Flexible/Pose #x folder for each Pose x.

import os
import sys
#This parameter should only be set as true if you have Sphgen_cpp 1.2 (or higher)
installed.
## Can be downloaded from
http://dock.compbio.ucsf.edu/Contributed_Code/sphgen_cpp.htm
sphgenNew=True;

'''_____NOTE: Review and change directories
below_____'''

dockDirectory="/home/blustig/perl5/dock/dock6+8/" #directory of dock installation, NO
BIN
sphgenNewDirectory="/home/blustig/perl5/dock/dock6+8/bin/sphgen_cpp.1.2/" #If
installed
rigidDirectory="/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant"

'''_____End of
directories_____'''
sphgenNew=True;

if (dockDirectory.endswith("/")==False):
    dockDirectory=dockDirectory+"/"
if (sphgenNewDirectory.endswith("/")==False):
    sphgenNewDirectory=sphgenNewDirectory+"/"
if rigidDirectory.endswith("/")==False:
    rigidDirectory=rigidDirectory+"/"


os.chdir(rigidDirectory)
```

```
#for mutant_name in os.walk('.').next()[1]:
mutant_name = sys.argv[1]

if ('Ligand' in mutant_name):
	exit()#continue

poseFile=mutant_name+'.rgd_selected150.mol2'  #these are the best ranked poses for
each mutant: ex. new-structure0001_rgd_selected150.mol2

try:
	os.chdir(rigidDirectory+mutant_name+'/04.dock/')
except OSError:
	print(mutant_name+' folder not found. Check directories')
	exit()#continue

selected_poses=open(poseFile,'r')

if ('Flexible' not in os.listdir('.')):
	os.mkdir('Flexible')

os.chdir('Flexible')

############################################################
i=0
poses=selected_poses.readlines()
while ("Name:" not in poses[i]):		#brings iterator i to first pose
	i=i+1
j=i					#j is the beginning of the next pose. Thus lines i-j are one
pose.

poseNum=1;
while (j<len(poses)):
	j=i+1
	while(j<len(poses) and ("Name:" not in poses[j])):
	  j=j+1
	if ('Pose '+str(poseNum) not in os.listdir('.')):
	  os.mkdir('Pose '+str(poseNum))
	os.chdir('Pose '+str(poseNum))
	pose=open('lig.mol2','w')
	poseNum=poseNum+1
	for k in range(i,j):
	  pose.write(poses[k])
	i=j;
```

```
        os.chdir('..')
        pose.close()
        if(j==len(poses)):
          break

########################################
selected_poses.close()

#output_File=open("dockLog.out","wb")


mutant_directory=rigidDirectory+mutant_name+'/'

for ligand in os.walk(mutant_directory+"04.dock/Flexible").next()[1]:

        ligandDirectory=mutant_directory+"04.dock/Flexible/"+ligand+'/'
        os.chdir(ligandDirectory)
        os.system(dockDirectory + "bin/sphere_selector "+ mutant_directory + '02.surface-
spheres/' + mutant_name + ".rec.sph "+"lig.mol2 8.0")

        dirList=os.listdir(".");
        if "input2.in" in dirList:
          os.remove('input2.in')
        si=open("input2.in", "wb");
        si.write('selected_spheres.sph\n')
        si.write('-1\n')
        si.write('N\n')
        si.write('output_spheres_selected\n')
        si.write('N\n')
        si.close();

        os.system(dockDirectory + "bin/showsphere < input2.in")
        #NOTE: Saves as output_spheres_selected_1

        dirList=os.listdir(".");
        for doc in dirList:
          if "output_spheres_selected" in doc:
                  os.rename(doc, "output_spheres_selected.pdb");

        #os.chdir("../03.box-grid/");
        dirList=os.listdir(".");
        if "showbox.in" in dirList:
          os.remove('showbox.in')
        si=open("showbox.in", "wb");
```

```
        si.write('Y\n')
        si.write('8.0\n')
        si.write('selected_spheres.sph\n')  #../02.surface-spheres/
        si.write('1\n')
        si.write(mutant_name+'.box.pdb\n')
        si.close();

        os.system(dockDirectory + "bin/showbox < showbox.in")

        dirList=os.listdir(".");
        os.system('cp '+mutant_directory+'01.dockprep/'+mutant_name+'.rec.mol2'+'
'+mutant_name+'.rec.mol2')
        if "grid.in" in dirList:
          os.remove('grid.in')
        gi=open("grid.in", "wb");
        gi.write('grid\n')
        gi.write('compute_grids\tyes\n')
        gi.write('grid_spacing\t0.4\n')
        gi.write('output_molecule\tno\n')
        gi.write('contact_score\tno\n')
        gi.write('energy_score\tyes\n')
        gi.write('energy_cutoff_distance\t9999\n')
        gi.write('atom_model\ta\n')
        gi.write('attractive_exponent\t6\n')
        gi.write('repulsive_exponent\t9\n')
        gi.write('distance_dielectric\tyes\n')
        gi.write('dielectric_factor\t4\n')
        gi.write('bump_filter\tyes\n')
        gi.write('bump_overlap\t0.75\n')
        gi.write('receptor_file\t'+mutant_name+'.rec.mol2'+'\n')
        gi.write('\nbox_file\t'+mutant_name+'.box.pdb\n')
        gi.write('\nvdw_definition_file\t'+dockDirectory+'parameters/vdw_AMBER_parm9
9.defn\n')
        gi.write('\nscore_grid_prefix\tgrid')
        gi.close();

        os.system(dockDirectory + "bin/grid -i grid.in")

        #os.chdir("../04.dock")

        #Fixed Anchor Docking
        dirList=os.listdir(".")
        if "fxdAnchor.in" in dirList:
          os.remove("fxdAnchor.in")
```

```
        fixedAnchor=open("fxdAnchor.in",'w')
        fixedAnchor.write("conformer_search_type                        flex\n")
        fixedAnchor.write("user_specified_anchor                       no\n")
        fixedAnchor.write("limit_max_anchors                           no\n")
        fixedAnchor.write("min_anchor_size                             5\n")
        fixedAnchor.write("pruning_use_clustering                      yes\n")
        fixedAnchor.write("pruning_max_orients                         1000\n")
        fixedAnchor.write("pruning_clustering_cutoff                   100\n")
        fixedAnchor.write("pruning_conformer_score_cutoff                  100.0\n")
        fixedAnchor.write("pruning_conformer_score_scaling_factor              1.0\n")
        fixedAnchor.write("use_clash_overlap                           no\n")
        fixedAnchor.write("write_growth_tree                           no\n")
        fixedAnchor.write("use_internal_energy                         yes\n")
        fixedAnchor.write("internal_energy_rep_exp                     12\n")
        fixedAnchor.write("internal_energy_cutoff                      100.0\n")
        fixedAnchor.write("ligand_atom_file                            lig.mol2\n")
#../01.dockprep/"+structure_name+".
        fixedAnchor.write("limit_max_ligands                           no\n")
        fixedAnchor.write("skip_molecule                               no\n")
        fixedAnchor.write("read_mol_solvation                          no\n")
        fixedAnchor.write("calculate_rmsd                              yes\n")
        fixedAnchor.write("use_rmsd_reference_mol                          yes\n")
        fixedAnchor.write("rmsd_reference_filename                         lig.mol2\n")
#../01.dockprep/"+structure_name+".
        fixedAnchor.write("use_database_filter                         no\n")
        fixedAnchor.write("orient_ligand                               no\n")
        fixedAnchor.write("automated_matching                              yes\n")
        fixedAnchor.write("receptor_site_file
selected_spheres.sph\n")#../02.surface-spheres/
        fixedAnchor.write("max_orientations                            1000\n")
        fixedAnchor.write("critical_points                             no\n")
        fixedAnchor.write("chemical_matching                           no\n")
        fixedAnchor.write("use_ligand_spheres                          no\n")
        fixedAnchor.write("bump_filter                                 no\n")
        fixedAnchor.write("score_molecules                             yes\n")
        fixedAnchor.write("contact_score_primary                       no\n")
        fixedAnchor.write("contact_score_secondary                     no\n")
        fixedAnchor.write("grid_score_primary                          yes\n")
        fixedAnchor.write("grid_score_secondary                        no\n")
        fixedAnchor.write("grid_score_rep_rad_scale                    1\n")
        fixedAnchor.write("grid_score_vdw_scale                        1\n")
        fixedAnchor.write("grid_score_es_scale                         1\n")
        fixedAnchor.write("grid_score_grid_prefix
grid\n")#../03.box-grid/
```

```
        fixedAnchor.write("multigrid_score_secondary                    no\n")
        fixedAnchor.write("dock3.5_score_secondary                      no\n")
        fixedAnchor.write("continuous_score_secondary                    no\n")
        fixedAnchor.write("footprint_similarity_score_secondary             no\n")
        fixedAnchor.write("pharmacophore_score_secondary                    no\n")
        fixedAnchor.write("descriptor_score_secondary                   no\n")
        fixedAnchor.write("gbsa_zou_score_secondary                     no\n")
        fixedAnchor.write("gbsa_hawkins_score_secondary                    no\n")
        fixedAnchor.write("SASA_score_secondary                         no\n")
        fixedAnchor.write("amber_score_secondary                       no\n")
        fixedAnchor.write("minimize_ligand                           yes\n")
        fixedAnchor.write("minimize_anchor                           yes\n")
        fixedAnchor.write("minimize_flexible_growth                     yes\n")
        fixedAnchor.write("use_advanced_simplex_parameters                  no\n")
        fixedAnchor.write("simplex_max_cycles                          1\n")
        fixedAnchor.write("simplex_score_converge                      0.1\n")
        fixedAnchor.write("simplex_cycle_converge                      1.0\n")
        fixedAnchor.write("simplex_trans_step                        1.0\n")
        fixedAnchor.write("simplex_rot_step                          0.1\n")
        fixedAnchor.write("simplex_tors_step                         10.0\n")
        fixedAnchor.write("simplex_anchor_max_iterations                500\n")
        fixedAnchor.write("simplex_grow_max_iterations                  500\n")
        fixedAnchor.write("simplex_grow_tors_premin_iterations             0\n")
        fixedAnchor.write("simplex_random_seed                         0\n")
        fixedAnchor.write("simplex_restraint_min                      no\n")
        fixedAnchor.write("atom_model                                all\n")
        fixedAnchor.write("vdw_defn_file
"+dockDirectory+"parameters/vdw_AMBER_parm99.defn\n")
        fixedAnchor.write("flex_defn_file
"+dockDirectory+"parameters/flex.defn\n")
        fixedAnchor.write("flex_drive_file
"+dockDirectory+"parameters/flex_drive.tbl\n")
        fixedAnchor.write("ligand_outfile_prefix
"+mutant_name+".fixedAnchor\n")
        fixedAnchor.write("write_orientations                        no\n")
        fixedAnchor.write("num_scored_conformers                       100\n")
        fixedAnchor.write("write_conformations                       no\n")
        fixedAnchor.write("cluster_conformations                     yes\n")
        fixedAnchor.write("cluster_rmsd_threshold                     2.0\n")
        fixedAnchor.write("rank_ligands                             no")
        fixedAnchor.close()
        os.system(dockDirectory + "bin/dock6 -i fxdAnchor.in -v -o
fixedAnchorOutput.out")
```

```
        #Flexible Docking
        dirList=os.listdir(".")
        if "flx.in" in dirList:
          os.remove("flx.in")
        flexDock=open("flx.in","wb");
        flexDock.write("conformer_search_type                    flex\n")
        flexDock.write("user_specified_anchor                    no\n")
        flexDock.write("limit_max_anchors                        no\n")
        flexDock.write("min_anchor_size                          5\n")
        flexDock.write("pruning_use_clustering                   yes\n")
        flexDock.write("pruning_max_orients                      1000\n")
        flexDock.write("pruning_clustering_cutoff                100\n")
        flexDock.write("pruning_conformer_score_cutoff                100.0\n")
        flexDock.write("pruning_conformer_score_scaling_factor            1.0\n")
        flexDock.write("use_clash_overlap                        no\n")
        flexDock.write("write_growth_tree                        no\n")
        flexDock.write("use_internal_energy                      yes\n")
        flexDock.write("internal_energy_rep_exp                  12\n")
        flexDock.write("internal_energy_cutoff                   100.0\n")
        flexDock.write("ligand_atom_file                         lig.mol2\n")
#../01.dockprep/"+structure_name+".
        flexDock.write("limit_max_ligands                        no\n")
        flexDock.write("skip_molecule                            no\n")
        flexDock.write("read_mol_solvation                       no\n")
        flexDock.write("calculate_rmsd                           yes\n")
        flexDock.write("use_rmsd_reference_mol                   yes\n")
        flexDock.write("rmsd_reference_filename                  lig.mol2\n")
#../01.dockprep/"+structure_name+".
        flexDock.write("use_database_filter                      no\n")
        flexDock.write("orient_ligand                            yes\n")
        flexDock.write("automated_matching                       yes\n")
        flexDock.write("receptor_site_file
selected_spheres.sph\n") #../02.surface-spheres/
        flexDock.write("max_orientations                         1000\n")
        flexDock.write("critical_points                          no\n")
        flexDock.write("chemical_matching                        no\n")
        flexDock.write("use_ligand_spheres                       no\n")
        flexDock.write("bump_filter                              no\n")
        flexDock.write("score_molecules                          yes\n")
        flexDock.write("contact_score_primary                    no\n")
        flexDock.write("contact_score_secondary                  no\n")
        flexDock.write("grid_score_primary                       yes\n")
        flexDock.write("grid_score_secondary                     no\n")
        flexDock.write("grid_score_rep_rad_scale                 1\n")
```

```
        flexDock.write("grid_score_vdw_scale                    1\n")
        flexDock.write("grid_score_es_scale                    1\n")
        flexDock.write("grid_score_grid_prefix                  grid\n") #../03.box-
grid/
        flexDock.write("multigrid_score_secondary               no\n")
        flexDock.write("dock3.5_score_secondary                 no\n")
        flexDock.write("continuous_score_secondary              no\n")
        flexDock.write("footprint_similarity_score_secondary       no\n")
        flexDock.write("pharmacophore_score_secondary           no\n")
        flexDock.write("descriptor_score_secondary              no\n")
        flexDock.write("gbsa_zou_score_secondary                no\n")
        flexDock.write("gbsa_hawkins_score_secondary            no\n")
        flexDock.write("SASA_score_secondary                    no\n")
        flexDock.write("amber_score_secondary                   no\n")
        flexDock.write("minimize_ligand                        yes\n")
        flexDock.write("minimize_anchor                        yes\n")
        flexDock.write("minimize_flexible_growth                yes\n")
        flexDock.write("use_advanced_simplex_parameters            no\n")
        flexDock.write("simplex_max_cycles                      1\n")
        flexDock.write("simplex_score_converge                  0.1\n")
        flexDock.write("simplex_cycle_converge                  1.0\n")
        flexDock.write("simplex_trans_step                     1.0\n")
        flexDock.write("simplex_rot_step                       0.1\n")
        flexDock.write("simplex_tors_step                      10.0\n")
        flexDock.write("simplex_anchor_max_iterations             500\n")
        flexDock.write("simplex_grow_max_iterations               500\n")
        flexDock.write("simplex_grow_tors_premin_iterations         0\n")
        flexDock.write("simplex_random_seed                     0\n")
        flexDock.write("simplex_restraint_min                  no\n")
        flexDock.write("atom_model                            all\n")
        flexDock.write("vdw_defn_file
"+dockDirectory+"parameters/vdw_AMBER_parm99.defn\n")
        flexDock.write("flex_defn_file
"+dockDirectory+"parameters/flex.defn\n")
        flexDock.write("flex_drive_file
"+dockDirectory+"parameters/flex_drive.tbl\n")
        flexDock.write("ligand_outfile_prefix
"+mutant_name+".flx\n")
        flexDock.write("write_orientations                     no\n")
        flexDock.write("num_scored_conformers                   100\n")
        flexDock.write("write_conformations                    no\n")
        flexDock.write("cluster_conformations                  yes\n")
        flexDock.write("cluster_rmsd_threshold                  2.0\n")
        flexDock.write("rank_ligands                          no")
```

```
        flexDock.close();

        os.system(dockDirectory + "bin/dock6 -i flx.in -v -o dockOutput.out")
        "'with open("dockOutput.out") as dockOut:
          text=dockOut.read().replace("\n","")
        if "Total elapsed time" in text:
          output_File.write(mutant_name+" docked successfully\n")
        else:
          output_File.write(mutant_name+" was not docked. See dockOutput file\n")
        print(mutant_name+" is done")'


        #output_File.close()
```

#sbatch flexdock11-20.sh & sbatch flexdock21-30.sh & sbatch flexdock41-50.sh & sbatch flexdock51-60.sh & sbatch flexdock71-80.sh & sbatch flexdock81-90.sh & sbatch flexdock101-110.sh & sbatch flexdock111-120.sh & sbatch flexdock131-140.sh & sbatch flexdock141-150.sh & sbatch flexdock161-170.sh & sbatch flexdock171-180.sh & sbatch flexdock191-200.sh & sbatch flexdock201-210.sh & sbatch flexdock221-230.sh & sbatch flexdock231-240.sh & sbatch flexdock251-260.sh & sbatch flexdock261-270.sh & sbatch flexdock281-290.sh & sbatch flexdock291-294.sh

Step 6: Use the code below to extract grid score, VDW, and electrostatics components for every single pose generated by flexible docking.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
#grid_score.py
#Script to get the grid score, grid Vander Waal energy, and grid electrostatic energy for each structure generated by
##flexible docking.


import os

def getScore():

        directoryOfStructures="/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant/"
        energies=[]
          #stores the name/identifiable info and energy of each pose generated by flexible docking
          ##each line is in csv format
```

```
        if directoryOfStructures.endswith('/')==False:
          directoryOfStructures=directoryOfStructures+'/'
        os.chdir(directoryOfStructures)

        for structure in os.walk('.').next()[1]:
          if 'Ligand' in structure:
                continue

          if '04.dock' not in os.walk(directoryOfStructures+structure).next()[1]:
                continue

          if 'Flexible' not in os.listdir(directoryOfStructures+structure+'/'+'04.dock/'):
                energies.append('******************'+structure+' has not been docked
yet******************')
                continue

          dockPlace=directoryOfStructures+structure+'/'+'04.dock/Flexible/'

          for Pose in os.walk(dockPlace).next()[1]:
                flexibleFile=structure+'.flx_scored.mol2'
                i=0

                if flexibleFile not in os.listdir(dockPlace+Pose):
                      energies.append(": "+structure+' Rigid-Pose_'+Pose.replace('Pose
','')+' has not yet been docked')
                      continue

                scored=open(dockPlace+Pose+'/'+flexibleFile,'r').readlines()
                while (i<len(scored) and ("Name:" not in scored[i])):
                      i=i+1
                j=i
                flexPoseNum=0  #the index of each pose generated by flexible docking -
NOT Rigid docking
                while (j<len(scored)):
                      j=i+1
                      while(j<len(scored) and ("Name:" not in scored[j])):
                            j=j+1
                      #Here j is end of the structure
                      g=i
                      while ("Grid_Score:" not in scored[g]):
                            g = g+1
                      flexPoseNum = flexPoseNum+1
```

```
                        grid_score = float(scored[g].split()[2]) #splits line on white space
regex, converts to float
                        grid_vdw = float(scored[g+1].split()[2]) #Vander Waal Energy
                        grid_es = float(scored[g+2].split()[2]) #Electrostatic energy

                        energies.append(structure+','+'Rigid-'+Pose.replace('
','_')+','+'Flexible-
Pose_'+str(flexPoseNum)+','+str(grid_score)+','+str(grid_vdw)+','+str(grid_es))

                        i = j;
                        if(j == len(scored)):
                                break
        return energies

def scoreAllStructures():
        allStructuresEnergies=getScore()
        return allStructuresEnergies

def scoreAllMutants():
        allEnergies=["Structure Name,Rigid Pose Number,Flexible Pose Number,Grid
Score,Grid Vander Waal,Grid Electrostatic"]
        allEnergies = allEnergies + scoreAllStructures()
        return allEnergies

with open('/home/blustig/perl5/dock/dock6+8/bin/1ehzMutant/gridScoreEnergies.csv',
'w') as outFile:
        gridScores = scoreAllMutants()
        for score in gridScores:
          outFile.write(score+'\n')
outFile.close()
```

Appendix B Read Me for GROMACS and MBAR

These scripts were written by Thanh Le.
Please note that these scripts are mainly used on an HPC.
Please note that you will need to change the paths and environment.
Please note that you will need to change all file names.

GROMACS

Step 1: use the below code to create a system.

```
#!/bin/bash

echo "1" > 1
/home/blustig/perl5/mmpbsa/gromacs/bin/gmx pdb2gmx -f 1mnb_prep_noH.pdb -o
complex.gro -water tip3p < 1
/home/blustig/perl5/mmpbsa/gromacs/bin/gmx editconf -f complex.gro -o newbox.gro -bt
dodecahedron -d 1.0
/home/blustig/perl5/mmpbsa/gromacs/bin/gmx solvate -cp newbox.gro -cs -p topol.top -o
solv.gro
/home/blustig/perl5/mmpbsa/gromacs/bin/gmx grompp -f em.mdp -c solv.gro -p topol.top
-o ions.tpr
echo "14" > 14
/home/blustig/perl5/mmpbsa/gromacs/bin/gmx genion -s ions.tpr -o solv_ions.gro -p
topol.top -pname NA -nname CL -conc 0.15 -neutral < 14
```

Step 2: use the code below to run energy minimization.

```
#!/bin/bash

#SBATCH -p nodes
#SBATCH --job-name="mini1mnb"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 28

source /home/blustig/perl5/mmpbsa/gromacs/bin/GMXRC

gmx grompp -f em_real.mdp -c solv_ions.gro -p topol.top -o em.tpr
gmx mdrun -npme 5 -ntmpi 23 -pin on -v -stepout 1000 -deffnm em

echo "1 | 4" > inputndx
echo " " >> inputndx
echo "q" >> inputndx
```

gmx make_ndx -f em.gro -o index.ndx < inputndx

Step 3: use the code below to run equilibration.

```
#!/bin/bash

#SBATCH -p nodes
#SBATCH --job-name="nvt-npt1mnb-100ns"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 28

source /home/blustig/perl5/mmpbsa/gromacs/bin/GMXRC

gmx grompp -f nvt.mdp -c em.gro -p topol.top -n index.ndx -o nvt.tpr
gmx mdrun -npme 5 -ntmpi 23 -pin on -v -stepout 1000 -deffnm nvt

gmx grompp -f npt.mdp -c nvt.gro -t nvt.cpt -p topol.top -n index.ndx -o npt.tpr
gmx mdrun -npme 5 -ntmpi 23 -pin on -v -stepout 1000 -deffnm npt
```

Step 4: use the code below to run MD.

```
#!/bin/bash

#SBATCH -p nodes
#SBATCH --job-name=“md1mnb-100ns”
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 28

module purge
source /home/blustig/perl5/mmpbsa/gromacs/bin/GMXRC

gmx grompp -f md.mdp -c npt.gro -t npt.cpt -p topol.top -n index.ndx -o md.tpr
gmx mdrun -npme 5 -ntmpi 23 -pin on -v -stepout 1000 -deffnm md -dhdl dhdl
```

MDP files:

em.mdp

```
; LINES STARTING WITH ';' ARE COMMENTS
title                = Minimization          ; Title of run

; Parameters describing what to do, when to stop and what to save
```

```
integrator      = steep         ; Algorithm (steep = steepest descent minimization)
emtol           = 1000.0        ; Stop minimization when the maximum force < 10.0
kJ/mol
emstep      = 0.01      ; Energy step size
nsteps          = 50000                 ; Maximum number of (minimization) steps to
perform
energygrps      = system        ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to calculate the
interactions
nstlist             = 1             ; Frequency to update the neighbor list and long range
forces
cutoff-scheme   = Verlet
ns_type                 = grid              ; Method to determine neighbor list (simple,
grid)
rlist               = 1.0               ; Cut-off for making neighbor list (short range
forces)
coulombtype     = PME               ; Treatment of long range electrostatic interactions
rcoulomb        = 1.0               ; long range electrostatic cut-off
rvdw                = 1.0               ; long range Van der Waals cut-off
pbc         = xyz               ; Periodic Boundary Conditions
```

em_real.mdp

```
; LINES STARTING WITH ';' ARE COMMENTS
title               = Minimization      ; Title of run

; Parameters describing what to do, when to stop and what to save
integrator      = steep         ; Algorithm (steep = steepest descent minimization)
emtol           = 1000.0        ; Stop minimization when the maximum force < 10.0
kJ/mol
emstep      = 0.01      ; Energy step size
nsteps          = 50000                 ; Maximum number of (minimization) steps to
perform
energygrps      = Protein RNA       ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to calculate the
interactions
nstlist             = 1             ; Frequency to update the neighbor list and long range
forces
cutoff-scheme   = Verlet
ns_type                 = grid              ; Method to determine neighbor list (simple,
grid)
```

```
rlist                = 1.0              ; Cut-off for making neighbor list (short range
forces)
coulombtype      = PME              ; Treatment of long range electrostatic interactions
rcoulomb         = 1.0              ; long range electrostatic cut-off
rvdw             = 1.0              ; long range Van der Waals cut-off
pbc                 = xyz           ; Periodic Boundary Conditions
```

nvt.mdp

```
title       = Protein-ligand complex NVT equilibration
; Run parameters
integrator  = md        ; leap-frog integrator
nsteps      = 1000000     ; 1 * 2000000 = 2ns
dt          = 0.002       ; 1 fs
; Output control
nstxout     = 50000       ; save coordinates every 100.0 ps
nstvout     = 50000       ; save velocities every 100.0 ps
nstenergy   = 5000        ; save energies every 10.0 ps
nstlog      = 5000        ; update log file every 10.0 ps
energygrps  = RNA Protein
; Bond parameters
continuation    = no            ; first dynamics run
constraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds) constrained
lincs_iter      = 1             ; accuracy of LINCS
lincs_order     = 4             ; also related to accuracy
; Neighborsearching
cutoff-scheme   = Verlet
ns_type         = grid      ; search neighboring grid cells
nstlist         = 10        ; 20 fs, largely irrelevant with Verlet
rcoulomb        = 1.4       ; short-range electrostatic cutoff (in nm)
rvdw            = 1.4       ; short-range van der Waals cutoff (in nm)
; Electrostatics
coulombtype     = PME       ; Particle Mesh Ewald for long-range electrostatics
pme_order       = 4         ; cubic interpolation
fourierspacing  = 0.16      ; grid spacing for FFT
; Temperature coupling
tcoupl      = V-rescale                 ; modified Berendsen thermostat
tc-grps     = RNA_Protein Water_and_ions    ; two coupling groups - more accurate
tau_t       = 0.1   0.1                 ; time constant, in ps
ref_t       = 300   300                 ; reference temperature, one for each group, in K
; Pressure coupling
pcoupl      = no        ; no pressure coupling in NVT
; Periodic boundary conditions
```

```
pbc        = xyz       ; 3-D PBC
; Dispersion correction
DispCorr    = EnerPres  ; account for cut-off vdW scheme
; Velocity generation
gen_vel     = yes       ; assign velocities from Maxwell distribution
gen_temp    = 300       ; temperature for Maxwell distribution
gen_seed    = -1        ; generate a random seed
```

npt.mdp

```
title       = Protein-ligand complex NPT equilibration
; Run parameters
integrator  = md        ; leap-frog integrator
nsteps      = 1000000     ; 2 * 1,000,000 = 2ns
dt          = 0.002     ; 2 fs
; Output control
nstxout     = 50000       ; save coordinates every 100.0 ps
nstvout     = 50000       ; save velocities every 100.0 ps
nstenergy   = 5000        ; save energies every 10.0 ps
nstlog      = 5000        ; update log file every 10.0 ps
energygrps  = RNA Protein
; Bond parameters
continuation    = yes           ; first dynamics run
constraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds) constrained
lincs_iter      = 1             ; accuracy of LINCS
lincs_order     = 4             ; also related to accuracy
; Neighborsearching
cutoff-scheme   = Verlet
ns_type         = grid      ; search neighboring grid cells
nstlist         = 10        ; 20 fs, largely irrelevant with Verlet
rcoulomb        = 1.4       ; short-range electrostatic cutoff (in nm)
rvdw            = 1.4       ; short-range van der Waals cutoff (in nm)
; Electrostatics
coulombtype     = PME       ; Particle Mesh Ewald for long-range electrostatics
pme_order       = 4         ; cubic interpolation
fourierspacing  = 0.16      ; grid spacing for FFT
; Temperature coupling
tcoupl      = V-rescale                 ; modified Berendsen thermostat
tc-grps     = RNA_Protein Water_and_ions    ; two coupling groups - more accurate
tau_t       = 0.1   0.1                 ; time constant, in ps
ref_t       = 300   300                 ; reference temperature, one for each group, in K
; Pressure coupling
pcoupl      = Parrinello-Rahman         ; pressure coupling is on for NPT
```

```
pcoupltype  = isotropic                 ; uniform scaling of box vectors
tau_p       = 2.0                       ; time constant, in ps
ref_p       = 1.0                       ; reference pressure, in bar
compressibility = 4.5e-5                ; isothermal compressibility of water, bar^-1
refcoord_scaling    = com
; Periodic boundary conditions
pbc         = xyz       ; 3-D PBC
; Dispersion correction
DispCorr    = EnerPres  ; account for cut-off vdW scheme
; Velocity generation
gen_vel     = no        ; velocity generation off after NVT
```

md.mdp

```
title       = Protein-ligand complex MD simulation
; Run parameters
integrator  = md        ; leap-frog integrator
nsteps      = 50000000    ; 0.002 * 50,000,000 = 100,000 ps (100 ns)
dt          = 0.002     ; 2 fs
; Output control
nstxout             = 12500         ; suppress .trr output
nstvout             = 12500         ; suppress .trr output
nstenergy           = 1000      ; save energies every 2.0 ps
nstlog              = 1000      ; update log file every 2.0 ps
nstxout-compressed  = 1000      ; write .xtc trajectory every 2.0 ps
compressed-x-grps   = System
energygrps          = RNA Protein
; Bond parameters
continuation    = yes           ; first dynamics run
constraint_algorithm = lincs    ; holonomic constraints
constraints     = all-bonds     ; all bonds (even heavy atom-H bonds) constrained
lincs_iter      = 1             ; accuracy of LINCS
lincs_order     = 4             ; also related to accuracy
; Neighborsearching
cutoff-scheme   = Verlet
ns_type         = grid      ; search neighboring grid cells
nstlist         = 10        ; 20 fs, largely irrelevant with Verlet
rcoulomb        = 1.4       ; short-range electrostatic cutoff (in nm)
rvdw            = 1.4       ; short-range van der Waals cutoff (in nm)
; Electrostatics
coulombtype     = PME       ; Particle Mesh Ewald for long-range electrostatics
pme_order       = 4         ; cubic interpolation
fourierspacing  = 0.16      ; grid spacing for FFT
; Temperature coupling
```

```
tcoupl      = V-rescale                 ; modified Berendsen thermostat
tc-grps     = RNA_Protein Water_and_ions    ; two coupling groups - more accurate
tau_t       = 0.1   0.1                 ; time constant, in ps
ref_t       = 300   300                 ; reference temperature, one for each group, in K
; Pressure coupling
pcoupl      = Parrinello-Rahman         ; pressure coupling is on for NPT
pcoupltype  = isotropic                 ; uniform scaling of box vectors
tau_p       = 2.0                       ; time constant, in ps
ref_p       = 1.0                       ; reference pressure, in bar
compressibility = 4.5e-5                ; isothermal compressibility of water, bar^-1
; Periodic boundary conditions
pbc         = xyz       ; 3-D PBC
; Dispersion correction
DispCorr    = EnerPres  ; account for cut-off vdW scheme
; Velocity generation
gen_vel     = no        ; assign velocities from Maxwell distribution
```

MBAR

These scripts were written by Thanh Le.
Please note that these scripts are mainly used on an HPC.
Please note that you will need to change the paths and environment.
Please note that you will need to change all file names.

Complex System

Step 1: use the code below to create a system.

```
#!/bin/sh -l
#

module purge
module load gromacs-mpich mpich/gnu-4.8.5

echo "1" > 1
gmx pdb2gmx -f complex_noH.pdb -o complex.gro -water tip3p < 1
gmx editconf -f complex.gro -o newbox.gro -bt dodecahedron -d 1.0
gmx solvate -cp newbox.gro -cs -p topol.top -o solv.gro
gmx grompp -f em.mdp -c solv.gro -p topol.top -o ions.tpr
echo "14" > 14
gmx genion -s ions.tpr -o solv_ions.gro -p topol.top -pname NA -nname CL -conc 0.15 -
neutral < 14
```

Step 2: use the code below to run energy minimization.

```
#!/bin/sh -l
#

#SBATCH -p nodes
#SBATCH --job-name="md7578-00"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f enmin.00.mdp -c solv_ions.gro -p topol.top -o em.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm em

echo "1 | 4" > inputndx
echo " " >> inputndx
echo "q" >> inputndx
gmx make_ndx -f em.gro -o index.ndx < inputndx
```

Step 3: use the code below to run equilibration.

```
#!/bin/sh -l
#

#SBATCH -p nodes
#SBATCH --job-name="nvt-npt7578-00"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f nvt_0.mdp -c em.gro -p topol.top -n index.ndx -o nvt.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm nvt

gmx grompp -f npt_0.mdp -c nvt.gro -t nvt.cpt -p topol.top -n index.ndx -o npt.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm npt
```

Step 4: use the code below to run MD.

```
#!/bin/sh -l
```

```
#

#SBATCH -p nodes
#SBATCH --job-name="md75complex-00"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f prod_0.mdp -c npt.gro -t npt.cpt -p topol.top -n index.ndx -o md.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm md -dhdl dhdl
```

MDP files

em.mdp

```
; LINES STARTING WITH ';' ARE COMMENTS
title               = Minimization          ; Title of run

; Parameters describing what to do, when to stop and what to save
integrator      = steep          ; Algorithm (steep = steepest descent minimization)
emtol           = 1000.0        ; Stop minimization when the maximum force < 10.0
kJ/mol
emstep      = 0.01      ; Energy step size
nsteps          = 50000                 ; Maximum number of (minimization) steps to
perform
energygrps      = RNA Protein           ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to calculate the
interactions
nstlist             = 1                 ; Frequency to update the neighbor list and long range
forces
cutoff-scheme   = Verlet
ns_type                     = grid                  ; Method to determine neighbor list (simple,
grid)
rlist               = 1.0                   ; Cut-off for making neighbor list (short range
forces)
coulombtype     = PME                   ; Treatment of long range electrostatic interactions
rcoulomb        = 1.0                   ; long range electrostatic cut-off
rvdw                = 1.0                   ; long range Van der Waals cut-off
pbc           = xyz                 ; Periodic Boundary Conditions
```

enmin.00.mdp

```
;==================================================
; Energy minimization
;==================================================

;--------------------------------------------------
; RUN CONTROL & MINIMIZATION
;--------------------------------------------------
define              = -DFLEXIBLE
integrator           = steep
nsteps              = 10000
emtol               = 100
emstep               = 0.01
nstcomm               = 100

;--------------------------------------------------
; OUTPUT CONTROL
;--------------------------------------------------
nstxout              = 250       ; save coordinates to .trr every 250 steps
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr

nstxout-compressed     = 500       ; xtc compressed trajectory output every 500 steps
compressed-x-precision = 1000
nstlog               = 500       ; update log file every 500 steps
nstenergy             = 500       ; save energies every 500 steps
nstcalcenergy         = 100

;--------------------------------------------------
; NEIGHBOR SEARCHING
;--------------------------------------------------
cutoff-scheme         = Verlet
ns-type              = grid
nstlist             = 1
rlist               = 1.0

;--------------------------------------------------
; BONDS
;--------------------------------------------------
constraints           = none

;--------------------------------------------------
; ELECTROSTATICS
```

```
;--------------------------------------------------
coulombtype          = PME
rcoulomb             = 1.0
pme-order            = 6
fourierspacing       = 0.10
ewald-rtol           = 1e-6

;--------------------------------------------------
; VDW
;--------------------------------------------------
vdw-type             = PME
rvdw                 = 1.0
vdw-modifier         = Potential-Shift
ewald-rtol-lj        = 1e-3
lj-pme-comb-rule     = Geometric
DispCorr             = EnerPres

;--------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;--------------------------------------------------
Tcoupl           = no
Pcoupl           = no
gen_vel          = no

;--------------------------------------------------
; FREE ENERGY CALCULATIONS
;--------------------------------------------------
free-energy          = yes
couple-moltype       = Protein_chain_B
couple-lambda0       = vdw-q
couple-lambda1       = none
couple-intramol      = yes
separate-dhdl-file   = yes
sc-alpha             = 0.5
sc-power             = 1
sc-sigma             = 0.3
init-lambda-state    = 0
bonded-lambdas       = 0.0 0.01 0.025 0.05 0.075 0.1 0.2 0.35 0.5 0.75 1.0 1.00 1.0 1.00
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
coul-lambdas         = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.25 0.5 0.75
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas          = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.00 0.0 0.00
0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0
nstdhdl              = 100
```

```
calc-lambda-neighbors    = -1
```

nvt_0.mdp

```
;==========================================================
; NVT equilibration
;==========================================================

;----------------------------------------------------
; RUN CONTROL
;----------------------------------------------------
define      = -DPOSRES
integrator  = sd           ; stochastic leap-frog integrator
nsteps      = 2000000          ; 2 * 5,000 fs = 10 ps
dt          = 0.002        ; 2 fs
comm-mode   = Linear       ; remove center of mass translation
nstcomm     = 100          ; frequency for center of mass motion removal

;----------------------------------------------------
; OUTPUT CONTROL
;----------------------------------------------------
nstxout              = 0         ; don't save coordinates to .trr
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr
nstxout-compressed     = 5000      ; xtc compressed trajectory output every 5000 steps
compressed-x-precision = 1000      ; precision with which to write to the compressed
trajectory file
nstlog               = 5000      ; update log file every 10 ps
nstenergy            = 5000      ; save energies every 10 ps
nstcalcenergy        = 100       ; calculate energies every 100 steps

;----------------------------------------------------
; BONDS
;----------------------------------------------------
constraint_algorithm   = lincs     ; holonomic constraints
constraints            = all-bonds  ; hydrogens only are constrained
lincs_iter             = 1         ; accuracy of LINCS (1 is default)
lincs_order            = 4         ; also related to accuracy (4 is default)
lincs-warnangle        = 30        ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation           = no        ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;----------------------------------------------------
```

```
; NEIGHBOR SEARCHING
;----------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC

;----------------------------------------------------
; ELECTROSTATICS
;----------------------------------------------------
coulombtype      = PME      ; Particle Mesh Ewald for long-range electrostatics
rcoulomb         = 1.0      ; short-range electrostatic cutoff (in nm)
ewald_geometry   = 3d       ; Ewald sum is performed in all three dimensions
pme-order        = 6        ; interpolation order for PME (default is 4)
fourierspacing   = 0.10     ; grid spacing for FFT
ewald-rtol       = 1e-6     ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;----------------------------------------------------
; VDW
;----------------------------------------------------
vdw-type               = PME
rvdw                   = 1.0
vdw-modifier           = Potential-Shift
ewald-rtol-lj          = 1e-3
lj-pme-comb-rule       = Geometric
DispCorr               = EnerPres

;----------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;----------------------------------------------------
tc_grps          = System
tau_t            = 1.0
ref_t            = 300
pcoupl           =  no

;----------------------------------------------------
; VELOCITY GENERATION
;----------------------------------------------------
gen_vel      = yes      ; Velocity generation is on (if gen_vel is 'yes', continuation should
be 'no')
gen_seed     = -1       ; Use random seed
gen_temp     = 300
```

```
;--------------------------------------------------
; FREE ENERGY CALCULATIONS
;--------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
couple-lambda1          = none
couple-intramol         = yes
separate-dhdl-file      = yes
sc-alpha                = 0.5
sc-power                = 1
sc-sigma                        = 0.3
init-lambda-state       = 0
bonded-lambdas          = 0.0 0.01 0.025 0.05 0.075 0.1 0.2 0.35 0.5 0.75 1.0 1.00 1.0 1.00
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
coul-lambdas            = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.25 0.5 0.75
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas             = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.00 0.0 0.00
0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0
nstdhdl                 = 100
calc-lambda-neighbors    = -1
```

npt_0.mdp

```
;==================================================
; NPT equilibration
;==================================================

;--------------------------------------------------
; RUN CONTROL
;--------------------------------------------------
define      = -DPOSRES
integrator   = sd           ; stochastic leap-frog integrator
nsteps      = 2000000       ; 2 * 50,000 fs = 100 ps
dt          = 0.002         ; 2 fs
comm-mode    = Linear       ; remove center of mass translation
nstcomm      = 100          ; frequency for center of mass motion removal

;--------------------------------------------------
; OUTPUT CONTROL
;--------------------------------------------------
nstxout             = 0         ; don't save coordinates to .trr
nstvout             = 0         ; don't save velocities to .trr
```

```
nstfout                 = 0         ; don't save forces to .trr
nstxout-compressed      = 5000      ; xtc compressed trajectory output every 5000 steps
compressed-x-precision = 1000       ; precision with which to write to the compressed
trajectory file
nstlog                  = 5000      ; update log file every 10 ps
nstenergy               = 5000      ; save energies every 10 ps
nstcalcenergy           = 100       ; calculate energies every 100 steps

;----------------------------------------------------
; BONDS
;----------------------------------------------------
constraint_algorithm    = lincs     ; holonomic constraints
constraints             = all-bonds  ; hydrogens only are constrained
lincs_iter              = 1         ; accuracy of LINCS (1 is default)
lincs_order             = 4         ; also related to accuracy (4 is default)
lincs-warnangle         = 30        ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation            = yes       ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;----------------------------------------------------
; NEIGHBOR SEARCHING
;----------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC

;----------------------------------------------------
; ELECTROSTATICS
;----------------------------------------------------
coulombtype     = PME     ; Particle Mesh Ewald for long-range electrostatics
rcoulomb        = 1.0     ; short-range electrostatic cutoff (in nm)
ewald_geometry  = 3d      ; Ewald sum is performed in all three dimensions
pme-order       = 6       ; interpolation order for PME (default is 4)
fourierspacing  = 0.10    ; grid spacing for FFT
ewald-rtol      = 1e-6    ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;----------------------------------------------------
; VDW
;----------------------------------------------------
vdw-type                = PME
```

```
rvdw                    = 1.0
vdw-modifier            = Potential-Shift
ewald-rtol-lj           = 1e-3
lj-pme-comb-rule        = Geometric
DispCorr                = EnerPres

;---------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;---------------------------------------------------
tc_grps         = System
tau_t           = 1.0
ref_t           = 300
pcoupl          = Berendsen
pcoupltype      = isotropic
tau_p           = 0.5               ; time constant (ps)
ref_p           = 1.0               ; reference pressure (bar)
compressibility  = 4.5e-05          ; isothermal compressibility of water (bar^-1)
refcoord-scaling = all

;---------------------------------------------------
; VELOCITY GENERATION
;---------------------------------------------------
gen_vel     = no     ; Velocity generation is off

;---------------------------------------------------
; FREE ENERGY CALCULATIONS
;---------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
couple-lambda1          = none
couple-intramol         = yes
separate-dhdl-file      = yes
sc-alpha                = 0.5
sc-power                = 1
sc-sigma                        = 0.3
init-lambda-state       = 0
bonded-lambdas          = 0.0 0.01 0.025 0.05 0.075 0.1 0.2 0.35 0.5 0.75 1.0 1.00 1.0 1.00
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
coul-lambdas            = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.25 0.5 0.75
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas             = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.00 0.0 0.00
0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0
nstdhdl                 = 100
```

```
calc-lambda-neighbors    = -1
```

prod_0.mdp

```
;==================================================
; Production simulation
;==================================================

;--------------------------------------------------
; RUN CONTROL
;--------------------------------------------------
integrator   = sd           ; stochastic leap-frog integrator
nsteps       = 500000       ; 2 * 500,000 fs = 1000 ps = 1 ns
dt           = 0.002        ; 2 fs
comm-mode    = Linear       ; remove center of mass translation
nstcomm      = 100          ; frequency for center of mass motion removal

;--------------------------------------------------
; OUTPUT CONTROL
;--------------------------------------------------
nstxout              = 0         ; don't save coordinates to .trr
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr
nstxout-compressed     = 1000      ; xtc compressed trajectory output every 1000 steps (2
ps)
compressed-x-precision = 1000      ; precision with which to write to the compressed
trajectory file
nstlog               = 1000      ; update log file every 2 ps
nstenergy            = 1000      ; save energies every 2 ps
nstcalcenergy        = 100       ; calculate energies every 100 steps

;--------------------------------------------------
; BONDS
;--------------------------------------------------
constraint_algorithm   = lincs      ; holonomic constraints
constraints            = all-bonds  ; hydrogens only are constrained
lincs_iter             = 1          ; accuracy of LINCS (1 is default)
lincs_order            = 4          ; also related to accuracy (4 is default)
lincs-warnangle        = 30         ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation           = yes        ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;--------------------------------------------------
```

```
; NEIGHBOR SEARCHING
;----------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC

;----------------------------------------------------
; ELECTROSTATICS
;----------------------------------------------------
coulombtype     = PME     ; Particle Mesh Ewald for long-range electrostatics
rcoulomb        = 1.0     ; short-range electrostatic cutoff (in nm)
ewald_geometry  = 3d      ; Ewald sum is performed in all three dimensions
pme-order       = 6       ; interpolation order for PME (default is 4)
fourierspacing  = 0.10    ; grid spacing for FFT
ewald-rtol      = 1e-6    ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;----------------------------------------------------
; VDW
;----------------------------------------------------
vdw-type              = PME
rvdw                  = 1.0
vdw-modifier          = Potential-Shift
ewald-rtol-lj         = 1e-3
lj-pme-comb-rule      = Geometric
DispCorr              = EnerPres

;----------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;----------------------------------------------------
tc_grps         = System
tau_t           = 1.0
ref_t           = 300
pcoupl          = Parrinello-Rahman
pcoupltype      = isotropic           ; uniform scaling of box vectors
tau_p           = 2                   ; time constant (ps)
ref_p           = 1.0                 ; reference pressure (bar)
compressibility = 4.5e-05             ; isothermal compressibility of water (bar^-1)

;----------------------------------------------------
; VELOCITY GENERATION
;----------------------------------------------------
```

```
gen_vel      = no       ; Velocity generation is off

;--------------------------------------------------
; FREE ENERGY CALCULATIONS
;--------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
couple-lambda1          = none
couple-intramol         = yes
separate-dhdl-file      = yes
sc-alpha                = 0.5
sc-power                = 1
sc-sigma                          = 0.3
init-lambda-state       = 0
bonded-lambdas          = 0.0 0.01 0.025 0.05 0.075 0.1 0.2 0.35 0.5 0.75 1.0 1.00 1.0 1.00
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
coul-lambdas            = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.25 0.5 0.75
1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas             = 0.0 0.00 0.000 0.00 0.000 0.0 0.0 0.00 0.0 0.00 0.0 0.00 0.0 0.00
0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0
nstdhdl                 = 100
calc-lambda-neighbors    = -1
```

Ligand System

Step 1: use the code below to create a system.

```
#!/bin/sh -l
#

module purge
module load gromacs-mpich mpich/gnu-4.8.5

echo "1" > 1
gmx pdb2gmx -f complex_noH.pdb -o complex.gro -water tip3p < 1
gmx editconf -f complex.gro -o newbox.gro -bt dodecahedron -d 1.0
gmx solvate -cp newbox.gro -cs -p topol.top -o solv.gro
gmx grompp -f em.mdp -c solv.gro -p topol.top -o ions.tpr
echo "13" > 13
gmx genion -s ions.tpr -o solv_ions.gro -p topol.top -pname NA -nname CL -conc 0.15 -
neutral < 13
```

Step 2: use the code below to run energy minimization.

```
#!/bin/sh -l
#

#SBATCH -p nodes
#SBATCH --job-name="mini7578-01"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f enmin.01.mdp -c solv_ions.gro -p topol.top -o em.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm em
```

Step 3: use the code below to run equilibration.

```
#!/bin/sh -l
#

#SBATCH -p nodes
#SBATCH --job-name="nvt-npt7578-01"
#SBATCH -o trial_%j.out
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f nvt_1.mdp -c em.gro -p topol.top -o nvt.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm nvt

gmx grompp -f npt_1.mdp -c nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm npt
```

Step 4: use the code below to run MD.

```
#!/bin/sh -l
#

#SBATCH -p nodes
#SBATCH --job-name="md75-ligand-01"
#SBATCH -o trial_%j.out
```

```
#SBATCH -c 1
#SBATCH -n 24

module purge
module load gromacs-mpich mpich/gnu-4.8.5

gmx grompp -f prod_1.mdp -c npt.gro -t npt.cpt -p topol.top -o md.tpr
mpirun -np 24 mdrun_mpi -pin on -v -stepout 1000 -deffnm md -dhdl dhdl -append
```

MDP files

em.mdp

```
; LINES STARTING WITH ';' ARE COMMENTS
title               = Minimization          ; Title of run

; Parameters describing what to do, when to stop and what to save
integrator      = steep         ; Algorithm (steep = steepest descent minimization)
emtol           = 1000.0        ; Stop minimization when the maximum force < 10.0
kJ/mol
emstep      = 0.01      ; Energy step size
nsteps          = 50000                 ; Maximum number of (minimization) steps to
perform
energygrps      = Protein       ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to calculate the
interactions
nstlist             = 1             ; Frequency to update the neighbor list and long range
forces
cutoff-scheme   = Verlet
ns_type                 = grid              ; Method to determine neighbor list (simple,
grid)
rlist               = 1.0               ; Cut-off for making neighbor list (short range
forces)
coulombtype     = PME               ; Treatment of long range electrostatic interactions
rcoulomb        = 1.0               ; long range electrostatic cut-off
rvdw            = 1.0               ; long range Van der Waals cut-off
pbc         = xyz               ; Periodic Boundary Conditions
```

enmin.01.mdp

```
;====================================================
; Energy minimization
;====================================================
```

```
;----------------------------------------------------
; RUN CONTROL & MINIMIZATION
;----------------------------------------------------
define              = -DFLEXIBLE
integrator           = steep
nsteps              = 10000
emtol               = 100
emstep               = 0.01
nstcomm               = 100

;----------------------------------------------------
; OUTPUT CONTROL
;----------------------------------------------------
nstxout              = 250       ; save coordinates to .trr every 250 steps
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr

nstxout-compressed     = 500       ; xtc compressed trajectory output every 500 steps
compressed-x-precision = 1000
nstlog               = 500       ; update log file every 500 steps
nstenergy             = 500       ; save energies every 500 steps
nstcalcenergy         = 100

;----------------------------------------------------
; NEIGHBOR SEARCHING
;----------------------------------------------------
cutoff-scheme         = Verlet
ns-type              = grid
nstlist             = 1
rlist               = 1.0

;----------------------------------------------------
; BONDS
;----------------------------------------------------
constraints           = none

;----------------------------------------------------
; ELECTROSTATICS
;----------------------------------------------------
coulombtype           = PME
rcoulomb            = 1.0
pme-order           = 6
fourierspacing        = 0.10
```

```
ewald-rtol          = 1e-6

;-------------------------------------------------
; VDW
;-------------------------------------------------
vdw-type             = PME
rvdw                 = 1.0
vdw-modifier         = Potential-Shift
ewald-rtol-lj        = 1e-3
lj-pme-comb-rule     = Geometric
DispCorr             = EnerPres

;-------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;-------------------------------------------------
Tcoupl           = no
Pcoupl           = no
gen_vel          = no

;-------------------------------------------------
; FREE ENERGY CALCULATIONS
;-------------------------------------------------
free-energy          = yes
couple-moltype       = Protein_chain_B
couple-lambda0       = vdw-q
couple-lambda1       = none
couple-intramol      = yes
separate-dhdl-file   = yes
sc-alpha             = 0.5
sc-power             = 1
sc-sigma             = 0.3
init-lambda-state    = 1
coul-lambdas         = 0.0 0.25 0.5 0.75 1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0
1.00 1.0 1.00 1.0
vdw-lambdas          = 0.0 0.00 0.0 0.00 0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8
0.85 0.9 0.95 1.0
nstdhdl              = 100
calc-lambda-neighbors    = -1
```

nvt_1.mdp

```
;=================================================
; NVT equilibration
;=================================================
```

```
;----------------------------------------------------
; RUN CONTROL
;----------------------------------------------------
define      = -DPOSRES
integrator  = sd           ; stochastic leap-frog integrator
nsteps      = 2000000      ; 2 * 2,000,000 fs = 4000 ps = 4 ns
dt          = 0.002        ; 2 fs
comm-mode   = Linear       ; remove center of mass translation
nstcomm     = 100          ; frequency for center of mass motion removal

;----------------------------------------------------
; OUTPUT CONTROL
;----------------------------------------------------
nstxout             = 0         ; don't save coordinates to .trr
nstvout             = 0         ; don't save velocities to .trr
nstfout             = 0         ; don't save forces to .trr
nstxout-compressed  = 5000      ; xtc compressed trajectory output every 5000 steps
compressed-x-precision = 1000   ; precision with which to write to the compressed
trajectory file
nstlog              = 5000      ; update log file every 10 ps
nstenergy           = 5000      ; save energies every 10 ps
nstcalcenergy       = 100       ; calculate energies every 100 steps

;----------------------------------------------------
; BONDS
;----------------------------------------------------
constraint_algorithm  = lincs     ; holonomic constraints
constraints           = all-bonds ; hydrogens only are constrained
lincs_iter            = 1         ; accuracy of LINCS (1 is default)
lincs_order           = 4         ; also related to accuracy (4 is default)
lincs-warnangle       = 30        ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation          = no        ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;----------------------------------------------------
; NEIGHBOR SEARCHING
;----------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC
```

```
;--------------------------------------------------
; ELECTROSTATICS
;--------------------------------------------------
coulombtype      = PME      ; Particle Mesh Ewald for long-range electrostatics
rcoulomb         = 1.0      ; short-range electrostatic cutoff (in nm)
ewald_geometry   = 3d       ; Ewald sum is performed in all three dimensions
pme-order        = 6        ; interpolation order for PME (default is 4)
fourierspacing   = 0.10     ; grid spacing for FFT
ewald-rtol       = 1e-6     ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;--------------------------------------------------
; VDW
;--------------------------------------------------
vdw-type              = PME
rvdw                  = 1.0
vdw-modifier          = Potential-Shift
ewald-rtol-lj         = 1e-3
lj-pme-comb-rule      = Geometric
DispCorr              = EnerPres

;--------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;--------------------------------------------------
tc_grps         = System
tau_t           = 1.0
ref_t           = 300
pcoupl          =  no

;--------------------------------------------------
; VELOCITY GENERATION
;--------------------------------------------------
gen_vel      = yes      ; Velocity generation is on (if gen_vel is 'yes', continuation should
be 'no')
gen_seed     = -1       ; Use random seed
gen_temp     = 300

;--------------------------------------------------
; FREE ENERGY CALCULATIONS
;--------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
```

```
couple-lambda1          = none
couple-intramol         = yes
separate-dhdl-file      = yes
sc-alpha                = 0.5
sc-power                = 1
sc-sigma                        = 0.3
init-lambda-state       = 1
coul-lambdas            = 0.0 0.25 0.5 0.75 1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0
1.00 1.0 1.00 1.0
vdw-lambdas             = 0.0 0.00 0.0 0.00 0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8
0.85 0.9 0.95 1.0
nstdhdl                 = 100
calc-lambda-neighbors    = -1
```

npt_1.mdp

```
;==========================================================
; NPT equilibration
;==========================================================

;----------------------------------------------------
; RUN CONTROL
;----------------------------------------------------
define      = -DPOSRES
integrator   = sd           ; stochastic leap-frog integrator
nsteps      = 2000000        ; 2 * 2,000,000 fs = 4000 ps = 4 ns
dt          = 0.002        ; 2 fs
comm-mode    = Linear       ; remove center of mass translation
nstcomm      = 100          ; frequency for center of mass motion removal

;----------------------------------------------------
; OUTPUT CONTROL
;----------------------------------------------------
nstxout              = 0         ; don't save coordinates to .trr
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr
nstxout-compressed     = 5000       ; xtc compressed trajectory output every 5000 steps
compressed-x-precision = 1000       ; precision with which to write to the compressed
trajectory file
nstlog               = 5000       ; update log file every 10 ps
nstenergy             = 5000       ; save energies every 10 ps
nstcalcenergy         = 100        ; calculate energies every 100 steps

;----------------------------------------------------
```

```
; BONDS
;---------------------------------------------------
constraint_algorithm   = lincs      ; holonomic constraints
constraints            = all-bonds  ; hydrogens only are constrained
lincs_iter             = 1          ; accuracy of LINCS (1 is default)
lincs_order            = 4          ; also related to accuracy (4 is default)
lincs-warnangle        = 30         ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation           = yes        ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;---------------------------------------------------
; NEIGHBOR SEARCHING
;---------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC

;---------------------------------------------------
; ELECTROSTATICS
;---------------------------------------------------
coulombtype      = PME      ; Particle Mesh Ewald for long-range electrostatics
rcoulomb         = 1.0      ; short-range electrostatic cutoff (in nm)
ewald_geometry   = 3d       ; Ewald sum is performed in all three dimensions
pme-order        = 6        ; interpolation order for PME (default is 4)
fourierspacing   = 0.10     ; grid spacing for FFT
ewald-rtol       = 1e-6     ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;---------------------------------------------------
; VDW
;---------------------------------------------------
vdw-type               = PME
rvdw                   = 1.0
vdw-modifier           = Potential-Shift
ewald-rtol-lj          = 1e-3
lj-pme-comb-rule       = Geometric
DispCorr               = EnerPres

;---------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;---------------------------------------------------
```

```
tc_grps         = System
tau_t           = 1.0
ref_t           = 300
pcoupl          = Berendsen
pcoupltype      = isotropic
tau_p           = 0.5               ; time constant (ps)
ref_p           = 1.0               ; reference pressure (bar)
compressibility  = 4.5e-05          ; isothermal compressibility of water (bar^-1)
refcoord-scaling = all

;-------------------------------------------------
; VELOCITY GENERATION
;-------------------------------------------------
gen_vel     = no     ; Velocity generation is off

;-------------------------------------------------
; FREE ENERGY CALCULATIONS
;-------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
couple-lambda1          = none
couple-intramol         = yes
separate-dhdl-file      = yes
sc-alpha                = 0.5
sc-power                = 1
sc-sigma                          = 0.3
init-lambda-state       = 1
coul-lambdas            = 0.0 0.25 0.5 0.75 1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0
1.00 1.0 1.00 1.0
vdw-lambdas             = 0.0 0.00 0.0 0.00 0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8
0.85 0.9 0.95 1.0
nstdhdl                 = 100
calc-lambda-neighbors    = -1
```

prod_1.mdp

```
;==================================================
; Production simulation
;==================================================

;-------------------------------------------------
; RUN CONTROL
;-------------------------------------------------
```

```
integrator   = sd           ; stochastic leap-frog integrator
nsteps       = 50000000       ; 2 * 50,000,000 fs = 100000 ps = 100 ns
dt           = 0.002        ; 2 fs
comm-mode    = Linear       ; remove center of mass translation
nstcomm      = 100          ; frequency for center of mass motion removal

;---------------------------------------------------
; OUTPUT CONTROL
;---------------------------------------------------
nstxout              = 0         ; don't save coordinates to .trr
nstvout              = 0         ; don't save velocities to .trr
nstfout              = 0         ; don't save forces to .trr
nstxout-compressed     = 1000       ; xtc compressed trajectory output every 1000 steps (2
ps)
compressed-x-precision = 1000       ; precision with which to write to the compressed
trajectory file
nstlog               = 1000       ; update log file every 2 ps
nstenergy            = 1000       ; save energies every 2 ps
nstcalcenergy        = 100        ; calculate energies every 100 steps

;---------------------------------------------------
; BONDS
;---------------------------------------------------
constraint_algorithm   = lincs      ; holonomic constraints
constraints            = all-bonds  ; hydrogens only are constrained
lincs_iter             = 1          ; accuracy of LINCS (1 is default)
lincs_order            = 4          ; also related to accuracy (4 is default)
lincs-warnangle        = 30         ; maximum angle that a bond can rotate before LINCS
will complain (30 is default)
continuation           = yes        ; formerly known as 'unconstrained-start' - useful for exact
continuations and reruns

;---------------------------------------------------
; NEIGHBOR SEARCHING
;---------------------------------------------------
cutoff-scheme   = Verlet
ns-type         = grid   ; search neighboring grid cells
nstlist         = 10     ; 20 fs (default is 10)
rlist           = 1.0    ; short-range neighborlist cutoff (in nm)
pbc             = xyz    ; 3D PBC

;---------------------------------------------------
; ELECTROSTATICS
;---------------------------------------------------
```

```
coulombtype      = PME      ; Particle Mesh Ewald for long-range electrostatics
rcoulomb         = 1.0      ; short-range electrostatic cutoff (in nm)
ewald_geometry   = 3d       ; Ewald sum is performed in all three dimensions
pme-order        = 6        ; interpolation order for PME (default is 4)
fourierspacing   = 0.10     ; grid spacing for FFT
ewald-rtol       = 1e-6     ; relative strength of the Ewald-shifted direct potential at
rcoulomb

;--------------------------------------------------
; VDW
;--------------------------------------------------
vdw-type              = PME
rvdw                  = 1.0
vdw-modifier          = Potential-Shift
ewald-rtol-lj         = 1e-3
lj-pme-comb-rule      = Geometric
DispCorr              = EnerPres

;--------------------------------------------------
; TEMPERATURE & PRESSURE COUPL
;--------------------------------------------------
tc_grps         = System
tau_t           = 1.0
ref_t           = 300
pcoupl          = Parrinello-Rahman
pcoupltype      = isotropic           ; uniform scaling of box vectors
tau_p           = 2                   ; time constant (ps)
ref_p           = 1.0                 ; reference pressure (bar)
compressibility  = 4.5e-05            ; isothermal compressibility of water (bar^-1)

;--------------------------------------------------
; VELOCITY GENERATION
;--------------------------------------------------
gen_vel      = no       ; Velocity generation is off (if gen_vel is 'yes', continuation should
be 'no')
gen_seed     = -1       ; Use random seed
gen_temp     = 300

;--------------------------------------------------
; FREE ENERGY CALCULATIONS
;--------------------------------------------------
free-energy             = yes
couple-moltype          = Protein_chain_B
couple-lambda0          = vdw-q
```

couple-lambda1 = none
couple-intramol = yes
separate-dhdl-file = yes
sc-alpha = 0.5
sc-power = 1
sc-sigma = 0.3
init-lambda-state = 1
coul-lambdas = 0.0 0.25 0.5 0.75 1.0 1.00 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas = 0.0 0.00 0.0 0.00 0.0 0.05 0.1 0.2 0.3 0.4 0.5 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0
nstdhdl = 100
calc-lambda-neighbors = -1

LINK TO LEVITT PAPER
http://www.pnas.org/content/108/9/3590

Appendix C Additional Figures and Tables

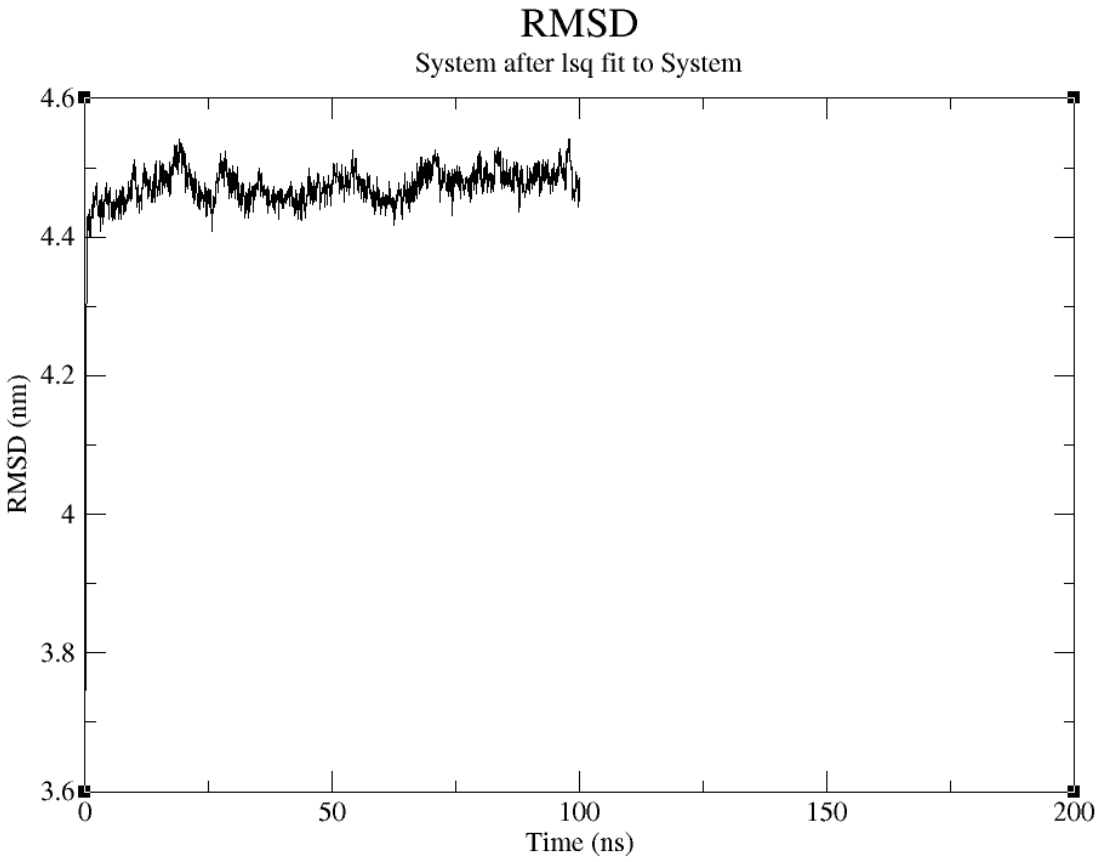

Figure C.1 RMSD of wild-type system.

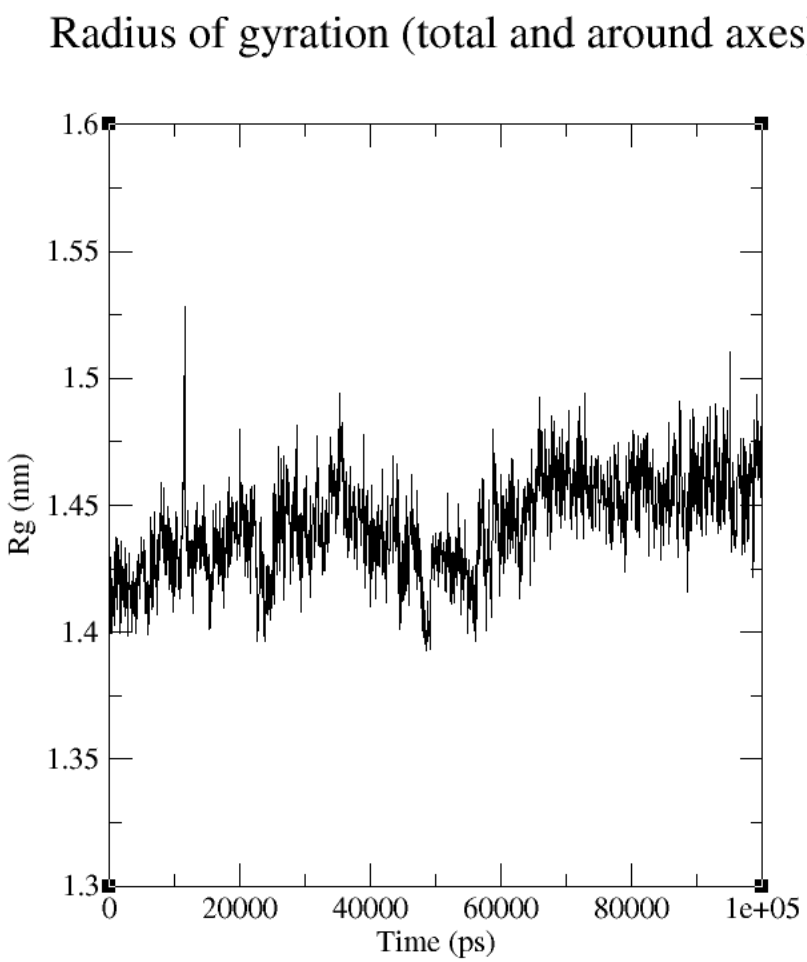

Figure C.2 Radius of gyration of wild-type TAR RNA.

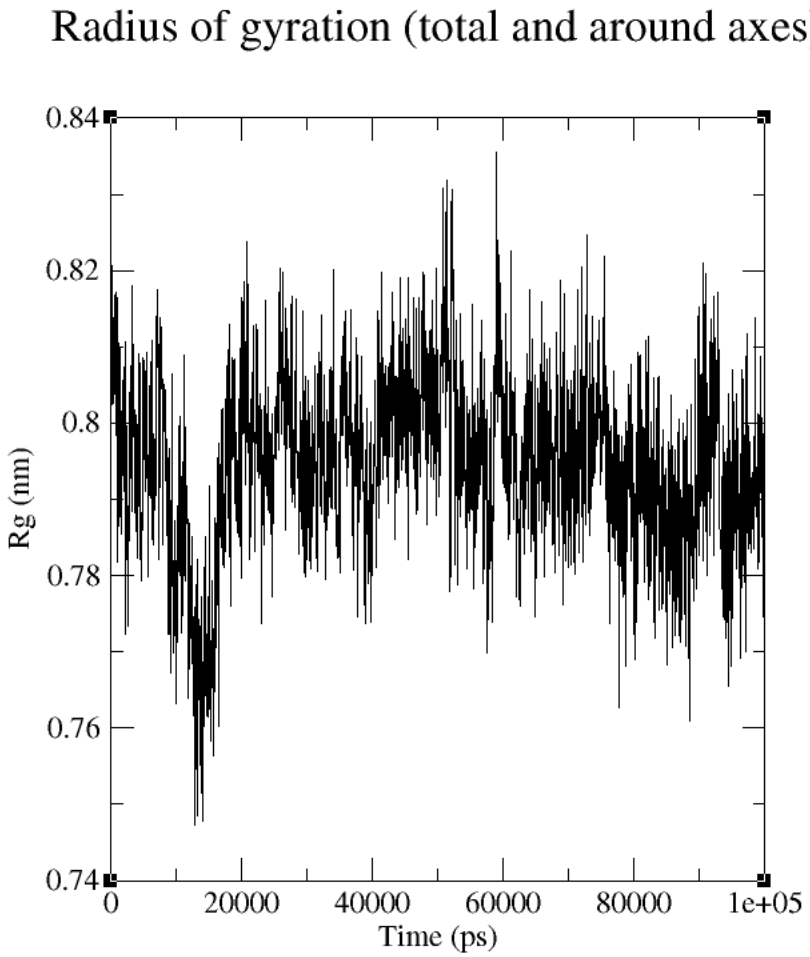

Figure C.3 Radius of gyration of wild-type peptide.

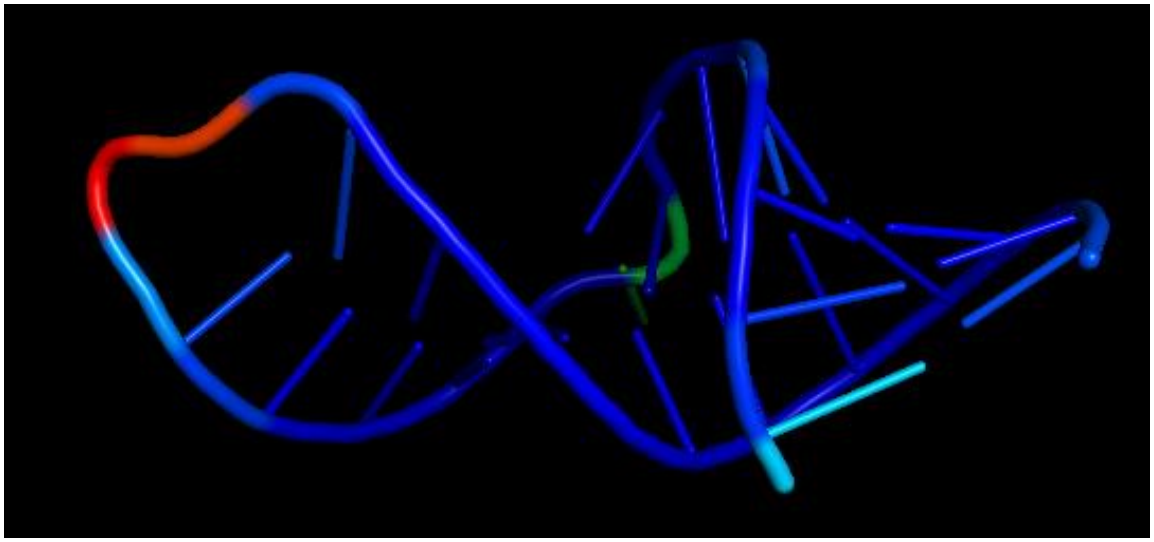

Figure C.4 RMSF representation of wild-type RNA.

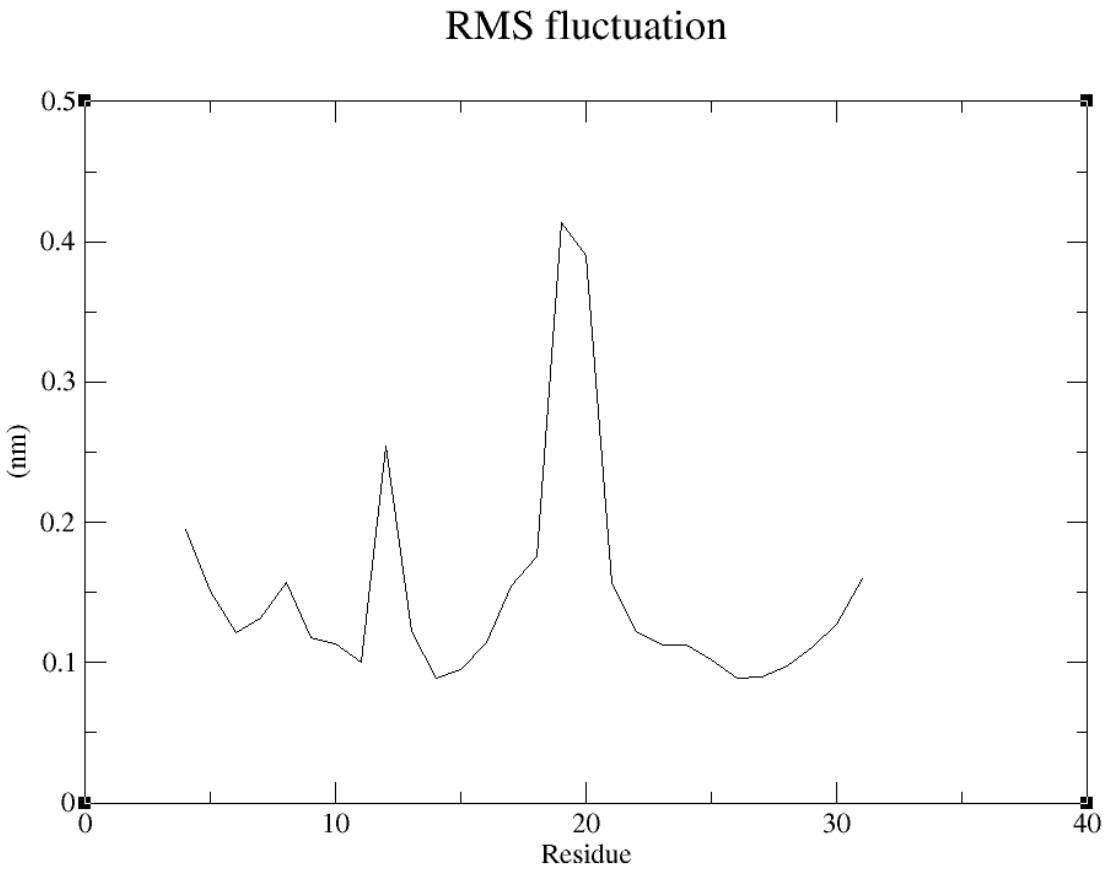

Figure C.5 RMSF value of each wild-type RNA nucleotide.

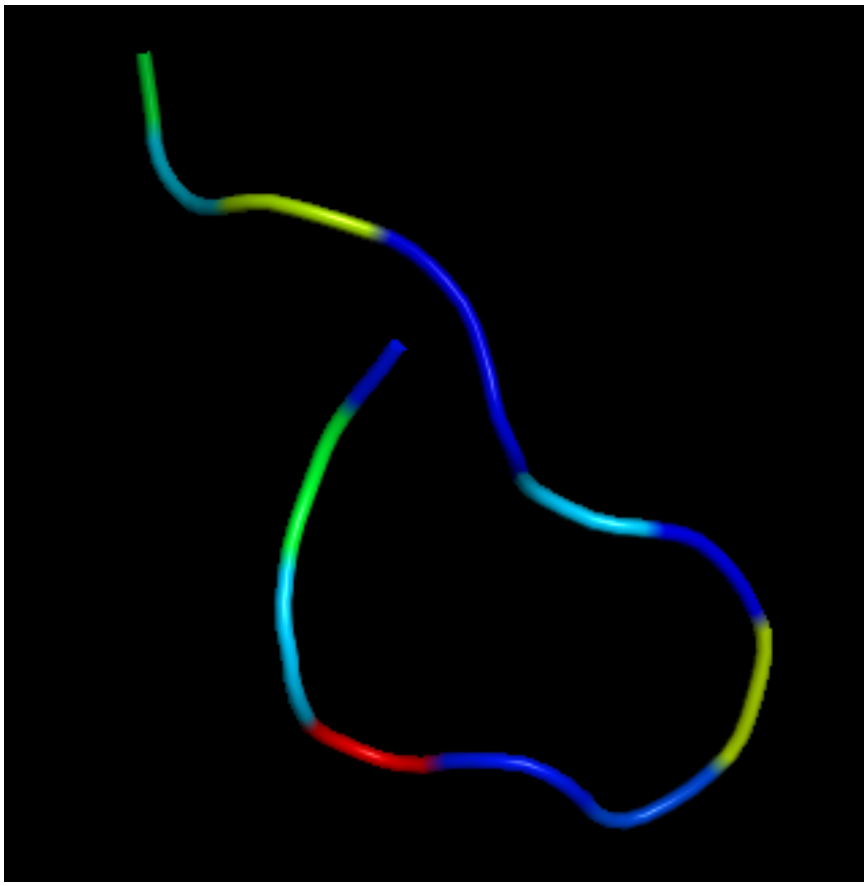

Figure C.6 RMSF representation of wild-type peptide.

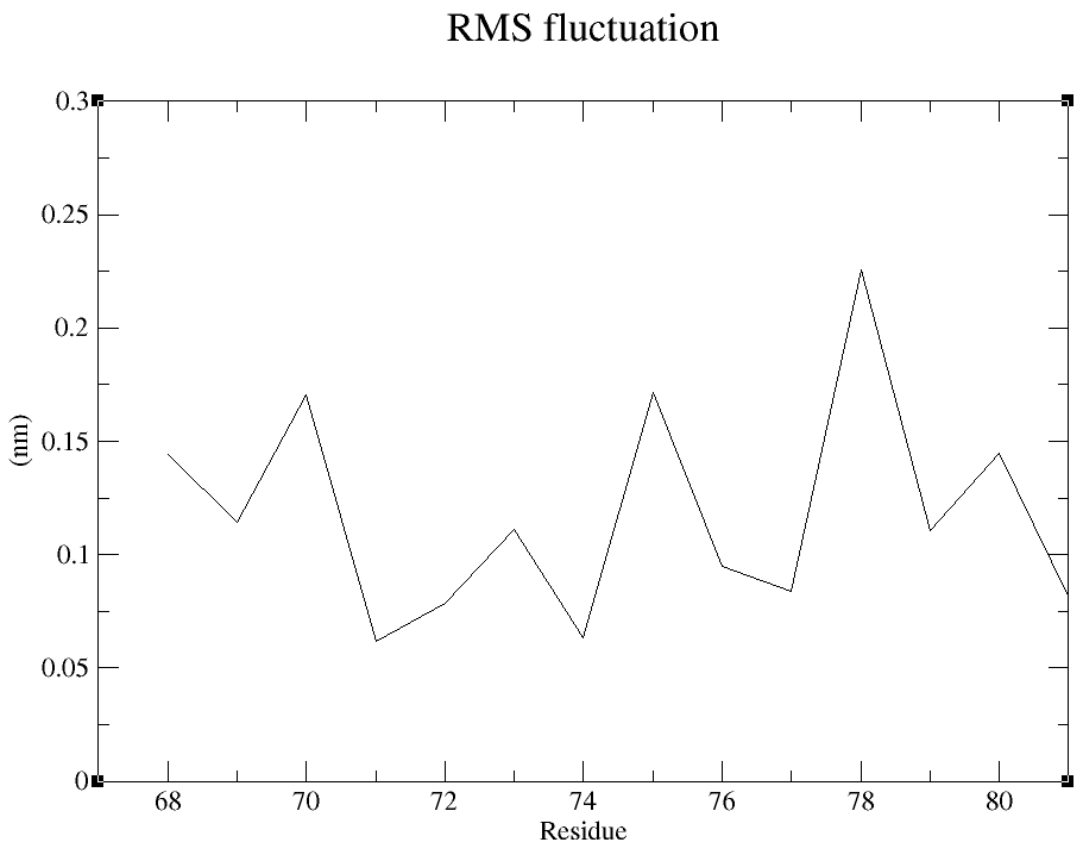

Figure C.7 RMSF of each wild-type Tat amino acid

((((((...((((....)))).))))))

Figure C.8 Dot-bracket representation of R78G mutant TAR RNA.

Table C.1 Possible hydrogen bonds formed between wild-type peptide and RNA TAR, column 1 and column 2, hydrogen bond donor and acceptor, respectively.

Donor	Acceptor
Arginine-68 NH2	Uracil-19 O4’
Glycine-71 N	Adenine-21 N7 (Major Groove)
Arginine-73 NH1	Guanine-11 O6 (Major Groove)
Lysine-75 N	Uracil-21 OP2
Adenine-21 N6 (Major Groove)	Proline-69 O
Cytosine-23 N4 (Major Groove)	Threonine-72 O
Cytosine-25 N4 (Major Groove)	Arginine-73 O

Table C.2 Possible residue-nucleotide contacts with hydrogen bonds formed between wild-type peptide and RNA TAR.

Peptide	RNA	Occupancy
Threonine-72	Cytosine-23	71.93%
Arginine-68	Uracil-20	50.35%
Arginine-68	Cytosine-17	43.46%
Arginine-77	Uracil-7	43.16%
Lysine-75	Uracil-24	35.81%
Arginine-77	Cytosine-6	25.77%
Arginine-73	Guanine-11	23.33%
Arginine-78	Cytosine-8	23.18%
Arginine-80	Guanine-22	19.28%
Arginine-72	Guanine-22	17.58%
Arginine-78	Uracil-7	15.33%
Arginine-80	Adenine-21	12.89%

Table C.3 Grid score, grid van der Waals energy, and grid electrostatics energy of DOCKed K75G mutant, R78G mutant, K75G-R78G mutant, and wild-type systems.

Systems	Grid Score (kcal/mol)	Grid VDW Energy (kcal/mol)	Grid Electrostatics Energy (kcal/mol)
K75G Mutant	-331.216	-158.420	-172.796
R78G Mutant	-344.817	-176.476	-168.341
K75G-R78G Mutant	-325.531	-159.535	-165.996
Wild-Type	-296.529	-144.076	-152.453