Computer Lab 8 – Ligand-Protein Docking

Computational docking of ligands to proteins is an important method in drug design. In this course, you have come to appreciate that a protein is a very flexible entity and must be thought of as an ensemble of conformations. Most drugs are small organic molecules and also exist as ensembles of conformations. The binding competent conformations in each of these two ensembles (protein and ligand) have to find each other in order for productive binding to occur. The goal in computational prediction of ligand-protein binding is to sample all possible docking conformations and calculate the *free energy* of each.

We have discussed one method for estimating the free energy of protein conformations using solvent ASA and bond torsions as terms in a very approximate force field. This force field was used to estimate the absolute free energy of a system but this is a crude approximation. The most rigorous approach to calculate the free energy involved in binding is to use a state of the art physics-based force field and carry out what are called free energy perturbation (FEP) molecular dynamics simulations. This method estimates the free energy *change* between two states, e.g., bound and unbound ligand, it does not calculate the absolute free energy of a system. It is not possible to calculate the absolute free energy of a system using molecular mechanics models and so we always calculate the relative free energy, i.e. the ΔG between two states. The FEP method also guides the system to various states to insure Boltzmann sampling

Historically, most physics-based force fields (CHARMM, AMBER, GROMOS) have focused on proteins and, more recently, on nucleic acids. A key component of recent progress in calculating the binding free energy (ΔG) of ligands is the development of a force field focused on drug-like molecules. Here is a quote describing this new force field. “Using the OPLS force field as a starting point, we have developed a new force field…that incorporates a robust model for non-bonded interactions (van der Waals parameters and partial charges) in conjunction with extensive training of torsional and covalent parameters against more than 10,000 representative organic compounds.” The potential energy function in the new force field is similar to the CHARMM potential energy function that you are familiar with. A key difference in the latest version of this new force field is the use of “off site” partial charge centers. This means that for some atoms the partial charges are not located at the atom centers. [Harder et al., JCTC 12:281, 2016].

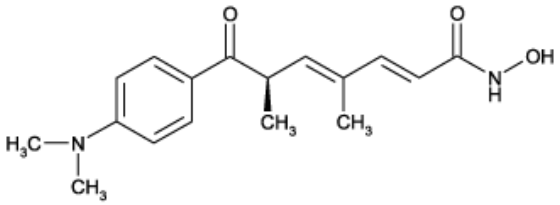
In this lab, you will use a program called AutoDock (autodock.scripps.edu) to study the binding of a ligand to a protein. This program uses another variation of an approximate force field to estimate the free energy. It does not use the above OPLS force field, rather it uses some common sense terms and a novel approach to speeding up the calculation of energies. You will only allow the ligand to vary in conformation and you will attempt to bind multiple, different ligand conformations with the single, binding-competent conformation of the protein. This task is easier than allowing *both* the protein and ligand conformations to vary but it is doable within our time constraints.

I. The system.

A. Histone Deacetylase. Histones are proteins containing high proportions of ARG and LYS and are net positively charged. These proteins are largely present in the nucleus and one of their functions is to interact with negatively charged DNA and promote tight packing of the chromatin. In order for DNA transcription to occur the histones are acetylated, the histones become less positively charged, interact less well with DNA, and the chromatin expands allowing polymerase access to the DNA.

In contrast, deacetylation of the histones by histone deacetylase promotes histone-DNA binding and chromatin condensation thereby inhibiting DNA transcription. Therefore, acetylation of histones is important in the regulation of gene expression.

B. Trichostatin A is a small molecule that binds to certain types of histone deacetylase enzymes and inhibits the enzymatic activity. This inhibition may interfere with the cell cycle and trichostatin has been used as an antibiotic. It is being tested for other clinical uses. The structure of trichostatin A is shown in the figure below. It has **seven** significant rotatable bonds.



A complex of human histone deacetlyase with bound trichostatin A has been crystallized and the X-ray diffraction structure is available. The PDB code is 1T64. The protein part of this crystal structure is, by definition, in the binding-competent conformation and you will use it in the experiment.

II. Description of the experiment.

In a real life experiment you may not have a crystal structure of the protein-ligand complex, rather you would be using a crystal structure or homology model of the apoprotein and attempting to determine if a *certain conformation* of a ligand could fit into a binding cleft on a *certain conformation* of the protein. In these types of experiments one usually has a pretty good idea where on the protein a ligand would bind so the search is limited to a specific surface area on the protein. Since the protein structure would be of the apoprotein, i.e. the conformation without ligand binding, one would have to allow at least this limited area of the protein to be flexible to accommodate the ligand. In our experiment, we know the protein is in the binding competent conformation so we will keep the protein atoms fixed and we will limit the search to a specific area of the protein surface. The purpose of this experiment is to demonstrate the feasibility of this type of method and to give you hands-on experience with the method.

You will use the program **AutoDock** to dock different conformations and orientations of **trichostatin A** to a model of human **histone deacetylase** and calculate the energies of binding for each ligand orientation. From analysis of the results you should be able to choose a cluster of similar ligand conformations and orientations that are most likely to

represent the native binding of the ligand. This distribution of complexes will be compared to the X-ray structure of the complex containing both protein and ligand in the binding competent conformation. You will also compare the crystal structure of the *ligand-binding protein conformation* with a typical *non-binding protein conformation* to appreciate the difficulty of the problem if no structure of the complex is known.

III. Methodology of AutoDock.

Docking simulations require two basic methods: A conformational search and a force field to evaluate the energetics of each conformation. One could attack the problem with brute force and calculate the interaction energies of a zillion random docked conformations but that would be very inefficient. AutoDock uses something called a genetic algorithm to narrow the search. In this method, a reasonable number of very different conformations are generated (global search) and the best of these are varied slightly in successive trials (local search). We won't go into this sophisticated search method in detail. The thing for you to know is that the different conformations of the ligand are obtained by varying the torsion angles of rotatable bonds. This is similar to the MC simulations of peptides in a vacuum that you did with the LINUS programs. You start out by making a number (~150) of different conformations of the ligand and these are translated and rotated to interact with a specified surface area of the protein.

A. AutoDock Force Field. The AutoDock energetic calculations use a hybrid force field containing typical molecular mechanics terms for potential energy E_{pot}, a term for entropy (configurational entropy S_{config}) and a term that contributes to the free energy directly (desolvation E_{desolv}). From a combination of these terms **apparent free energies** of binding may be calculated.

B. The potential energy terms are familiar.

$$\mathbf{E_{pot} = A(LJ) + B(Hbond) + C(Coulomb)}$$

where A, B, C are weighting factors found empirically using a database of cystallographic protein-ligand complexes, and LJ=Lennard-Jones, Hbond=hydrogen bond scoring potential, Coulomb=Coulombs law.

The desolvation term uses the same concepts of buried accessible surface area that we discussed in class,

$$\mathbf{E_{desolv} = D(ASA\ burial)}$$

where D is a weighting factor. This term estimates a *free energy* because it takes into account the change in entropy of solvating water upon protein atom burial.

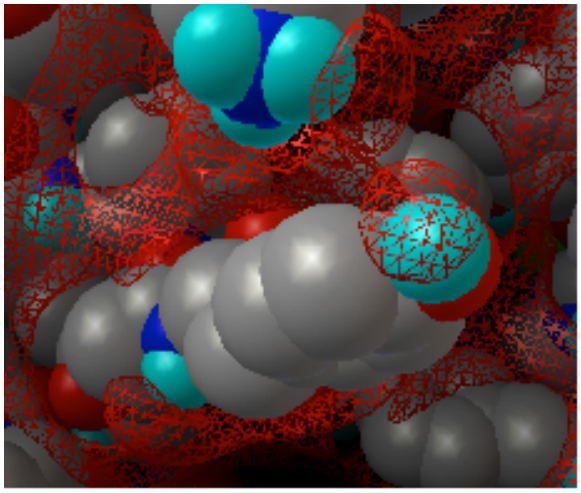
The configurational entropic term is proportional the number of rotatable bonds in the ligand which lose rotational freedom upon binding.

$$\mathbf{S_{config} = Q(\#Torsions)}$$

where Q is a weighting factor.

In summary, the AutoDock force field estimates total **free energy** of binding by including terms for (1) protein-ligand interaction potential energy, (2) the entropy change of water upon ligand binding and (3) the entropy change of ligand rotational freedom upon binding. (Burn that last sentence into your memory!)

AutoDock uses a limited number of atom types and the energy values for each atom pair interaction are *pre-calculated on a grid around the protein*. The protein surface of concern is placed in a cubic lattice grid and at each vertex of the grid, a hypothetical atom type is positioned and the **free energy** is calculated at that point in space for a particular atom type interaction; carbon-carbon, hydrogen-oxygen, etc. These pre-calculated grids are analogous to the probability density maps you calculated in class. They can be displayed in a molecular graphics program. For example, the figure at right shows the pre-calculated hydrogen bond acceptor energy contoured at a specific value of -0.35 kcal/mol (hydrogen atoms are colored cyan). In other words, if hydrogen bond acceptor atoms were on the red mesh locations, they would be scored as having a hydrogen bond interaction energy of -0.35. The map could be displayed with other contour values just as your probability density maps could be contoured for various probabilities.



The advantage of pre-calculating all the possible atom-atom interactions on the grid is that it speeds up the energy calculations of each trial docking. The energy of interaction at that point in space just needs to be looked up in a table instead of calculated over and over again during the simulation.

B. Analysis of the results. Autodock will **cluster** the different successfully docked ligand conformations according to conformational similarity (using RMSD of atomic postions). The average energies of the clusters are calculated and AutoDock provides graphical tools to inspect the cluster conformations. So, in keeping with the spirit of this course, we will consider ensembles (clusters) of ligand conformations and orientations rather than individual conformations.

IV. Lab Details.
The program Autodock has not yet been compiled for the cluster kirin. Therefore, I ran multiple jobs on a different cluster so that you would each have a different output to analyze. Each job took about 4-6 hours. Your individual completed results are in the tarball *docking_*.tar* in your permanent home directory on kirin. Login to a Mac, sftp to

kirin, cd to your permanent home directory and fetch the *docking_yourJHEDID.tar* file. Unpack this tarball on the Mac and cd to the new *docking_yourJHEDID/* directory. The files I started with were:

1t64.dpf	Docking run parameters
1t64_lig.pdbqt	Coordinates and atomic charges of ligand
1t64_lig.A.map	Interaction energy maps for atoms in ligand
1t64_lig.C.map	"
1t64_lig.HD.map	"
1t64_lig.N.map	"
1t64_lig.OA.map	"
1t64_lig.d.map	Desolvation energy map
1t64_lig.e.map	Electrostatics energy map
1t64_lig.maps.fld	Header info for map files
1t64_prot.pdbqt	Coordinates and atomic charges of protein
ligand.pdb	Coordinates of ligand (from the PDB)
protein.pdb	Coordinates of protein in binding competent (from PDB)
apoprotein.pdb	Coordinates of nonbinding-competent protein (from MD)

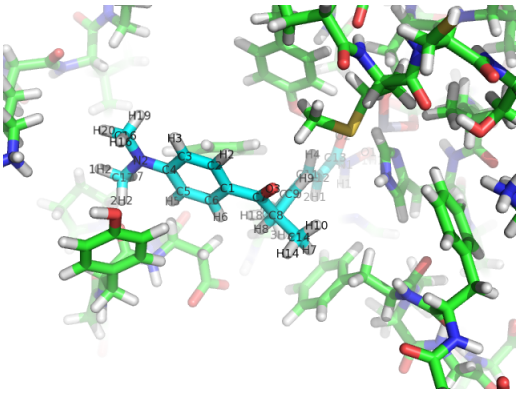
and the results are in the *1t64.dlg* file.

First become familiar with the structures involved. Load the *protein.pdb* and *ligand.pdb* files from your docking directory on the Mac into a PyMol session.

```
pymol protein.pdb ligand.pdb
```

Center on the ligand with light blue bonds (it should be the case upon starting), show everything as sticks and inspect the interactions of the ligand and the protein as below. Email the TA (**achin14@jhu.edu**) the answers to the yellow highlighted questions below.

1. Which atom-atom contacts between the ligand and protein are important for binding? Fill in the table below. (Hints: What protein atoms or residues, if any, are the following inhibitor atoms close to? [see table below] Could they have hydrogen bond, electrostatic or hydrophobic interactions? You can label the ligand atoms by clicking ligand | L | atom name. Or double clicking on an atom will launch a menu giving the name of the atom. You can also use this menu to center on that atom. Center on different atoms of the ligand to improve your ability to move around that atom and inspect the local environment. Use the Display | Clip | 16 Angstrom Slab menu choice to clarify your view. If you think the interaction may be hydrophobic it may be useful to look at both the sticks and spheres representations. Finally, remember that hydrogens are usually not seen in X-ray diffraction so the direction that a hydrogen is pointing in this model may not be in the proper orientation for a hydrogen bond that exists in reality.) Your display should look something like the following figure for this part of the exercise,



Ligand atom	Protein residue or heteroatom	Type of interaction
C14, C15 (methyl) O2 (C=O) 1H1 (O-H) H1 (N-H)		

Write these interactions down in your submitted answer to the question above so you can refer to them later.

Leave the ligand as sticks; for the protein, hide everything and show. Clear the labels and remove the hydrogens from the ligand object. Minimize the PyMol session to keep it available for reference later in the lab.

Now analyze the docking experiment. Launch the AutoDockTools graphics program from the main Applications folder on the Mac. It's in the MGLTools-latest directory and is called, AutoDockTools-1.5.6. (It may take a minute to launch; ignore the error messages about python.)

To open the docking results file in AutoDockTools, click on the following menu items,

Analyze | Dockings | Open…

and browse to your directory containing the docking results, then choose the file *1t64.dlg* and click the Open button. Click OK when the WARNING widget appears. You should see a line representation of the ligand on the graphics screen. Change the line drawing to a stick representation by clicking the red filled circle under the lines (L) heading and then click the open circle under the bonds (B) heading on the control panel at the left of the graphics screen.

Load the analyses of the ligand conformations. Click on the following menu items,

Analyze | Conformations | Load…

A Conformation Chooser widget should appear. You can double click on the first ranked

conformation of the first cluster (1t64_lig 1_1) and the bond drawing for that conformation will appear in the graphics window and the protein-ligand interaction energetics of that conformation will appear in the window on top. One could scroll through this list and look at the different ligand conformations but without the context of the protein this is not very useful. Move the Conformation Chooser widget out of the way for now but keep it handy.

An easier way to scroll through the ligand conformations is to call up the Conformation Player. Click on the following menu items,

Analyze | Conformations | Play…

A typical movie control widget should appear. Click on the play arrow to see the range of ligand conformations sampled. Again, playing a movie of the conformations at this stage is not very useful but keep this widget available because you will use it below.

You will want to play through the conformations of different *clusters* of ligand conformations, so launch the clustering widget. Click on the following menu items,

Analyze | Clusterings | Show…

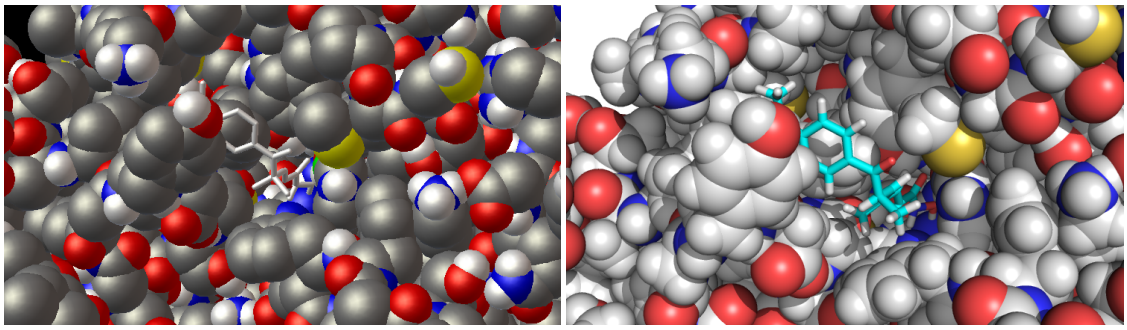
A bar graph should appear showing the number of ligand conformations in each cluster *versus* the average binding energy of each cluster. Move this widget to the side where you can click on it but also see the graphics screen. Now, if you click on a bar in the graph, that cluster becomes the active cluster for the movie player. Try it by alternately clicking a bar in the graph and the movie player. You should see that each bar cluster has similar conformations. In other words the ligand is exploring local conformational space.

Now let's put these ligand conformations in context by displaying the protein. Click on the following menu items,

Analyze | Macromolecule | Open…

Click OK in the warning box and find *1t64_prot.pdbqt* in the browser box that should appear, highlight it and click Open. Turn off the lines representation and turn on the space filling (C) representation as above and then color By atom type by clicking on the triangle under the color gradient icon and to the right of the label, 1t64_prot. This should launch a widget; click on By atom type. Close the widget.

Use the mouse buttons to translate, rotate and zoom the image. You want to be able to see the ligand in its binding site. (Rotate is left click, zoom is middle scroll and translate is right click.) Orient the protein so that it is similar to the orientation in your PyMOL session. Your two graphics screens should look something like those in the figures below although the ligand may be in a different orientation in the AutoDock display.



(Parenthetically, why are the relative atom radii different in these two images?)

Click on different clusters in the bar graph, play through the conformations of each cluster and decide which cluster is most similar to the X-ray complex structure shown in PyMOL.

2. Does the *lowest energy* cluster include the X-ray ligand conformation, or does the *most sampled* cluster include the X-ray conformation? (These may be the same cluster in your simulation.)
3. Are all of the ligand-protein interactions you identified above as being important satisfied in the conformations from the best cluster?

So far you should have successfully docked multiple conformations of a ligand to a static protein model known to be in the binding competent conformation. What if you did not have the binding competent conformation but rather just a model of the protein in the uncomplexed conformation?

Load the file *apoprotein.pdb* (included in the *docking_*/*) into the PyMol session containing the protein and ligand structures. Hide everything and show cartoon representations for both protein objects. You should be able to see that these are very similar proteins and are well within the range of flexibility expected for a native protein structure. Remember that these are the same protein, but one structure is in the average binding competent conformation and the other is just a sample of a ligand-free conformation. Now hide everything again and show everything as sticks. Center on a ligand atom, zoom into the ligand and clip to 16 Å. By alternatively activating the crystal structure protein object and the apoprotein object you can see which protein sidechains in the binding site are different.

4. Which apoprotein sidechains have VDW clash with the ligand?

All of these sidechains would have to be flexible in order for the ligand to bind. Also, they would all have to flex in a coordinated way to open up room for the ligand to fit! Finding such conformations in a conformational search would take a very long simulation and that is why we used the static binding-competent protein conformation.

If you were going to design an improved drug based on trichostatin A you might obtain the structure of a bacterial histone acetylase homologue, dock the trichostatin A molecule to it and try to identify places on the ligand where side groups are allowed in the bacterial

enzyme binding but not in the human enzyme binding. This would be a long-term research project but today you have seen the type of tools that are available and the thinking that goes into such a project.