BIOINF525: INTRODUCTION TO BIOINFORMATICS LAB SESSION 3

Protein Structure Visualization and Small Molecule Docking
http://bioboot.github.io/bioinf525_w16/module1/#1.3
Drs. Barry Grant
Jan 2016

Section 1: Introduction to the RCSB Protein Data Bank (PDB)

The PDB archive is the major repository of information about the 3D structures of large biological molecules, including proteins and nucleic acids. Understanding the shape of these molecules helps to understand how they work. This knowledge can be used to help deduce a structure's role in human health and disease, and in drug development. The structures in the PDB range from tiny proteins and bits of DNA or RNA to complex molecular machines like the ribosome composed of many chains of protein and RNA.

In the first section of this lab we will interact with the main US based PDB website (note there are also sites in Europe and Japan).

Visit: http://www.pdb.org/ and answer the following questions

NOTE: The "Analyze" -> "PDB Statistics" on the PDB home page should allow you to determine most of these answers.

Q1: What proportion of PDB entries does X-ray crystallography account for?

Q2: Type **HIV** in the search box on the home page and determine how many HIV-1 protease structures are in the current PDB?

Now download the **"PDB File (Text)"** for the HIV-1 protease structure with the PDB identifier **1HSG**. Examine the contents of your downloaded file in a text editor or at the **terminal**. The terminal is the main interface window on UNIX based computer systems. To open a terminal on the classroom computers you can click the black terminal icon on the top toolbar. Typical Unix commands for examining your downloaded file are:

```
> more ~/Downloads/1hsg.pdb          ## view the file text (use 'q' to quit)
> gedit ~/Downloads/1hsg.pdb         ## open the file in a basic text editor
```

NOTE: You can type **1HSG** in the PDB search box to jump to its entry and then click **"Download Files"** to the right of the top display. When viewing your downloaded file stop when you come the lines beginning with the word "ATOM". We will discuss this ubiquitous PDB file format when you have got this far.

Section 2: Visualizing the HIV-1 protease structure

The HIV-1 protease [1] is an enzyme that is vital for the replication of HIV. It cleaves newly formed polypeptide chains at appropriate locations so that they form functional proteins. Hence, drugs that target this protein could be vital for suppressing viral replication. A handful of drugs - called *HIV-1 protease inhibitors* (saquinavir, ritonavir, indinavir, nelfinavir, etc.) [2] - are currently commercially available that inhibit the function of this protein, by binding in the catalytic site that typically binds the polypeptide.

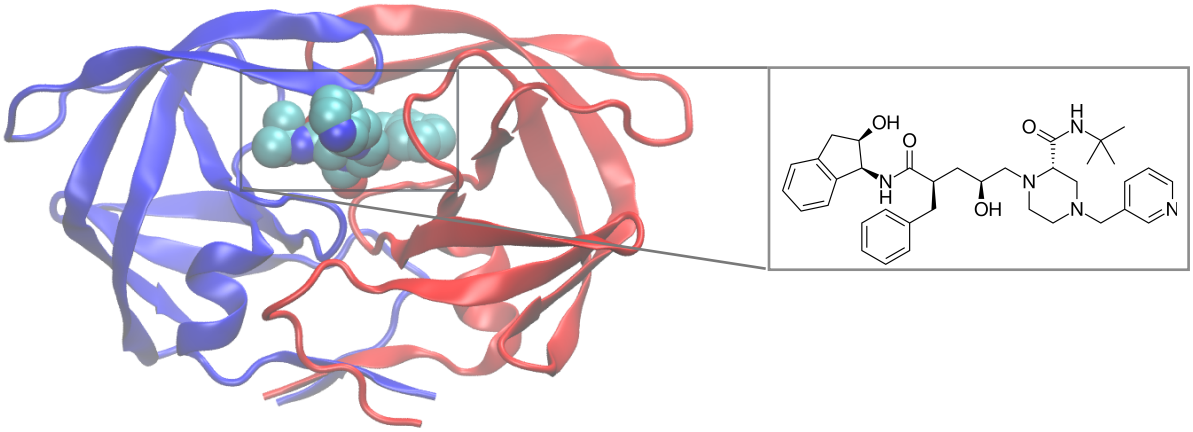


Figure 1. HIV-1 protease structure in complex with the small molecule indinavir.

In this section we will use the 2Å resolution X-ray crystal structure of HIV-1 protease with a bound drug molecule indinavir (PDB ID: 1HSG) [3]. We will use the **VMD molecular viewer** to visually inspect the protein, the binding site and the drug molecule. After exploring features of the complex we will move on to computationally dock a couple of drug molecules into the binding site of HIV-1 protease to see how well computational docking can reproduce the crystallographically observed binding pose. If time permits, we will also calculate the electrostatic surface of the protein to better appreciate how the drug interacts with the protein.

From your Unix terminal load the 1HSG structure into VMD using the command below:

```
> vmd 1hsg.pdb
```

You should see the protein structure displayed as lines and water molecules as little red dots. Use the mouse to zoom and rotate. Once you have the hang of rotation we will start exploring different "*Graphical Representations*".

VMD can display molecules in various ways by choosing different options in the *Graphical Representations* window shown in **Figure 2**. You can access this window by clicking **Graphics > Representations** from the small **VMD Main** window.

NOTE. Each representation is defined by three main parameters: (1) the *drawing method*, (2) the *selected atoms* to be included in the representation, and (3) the *coloring method* (see Figure 2 labels 1-3)

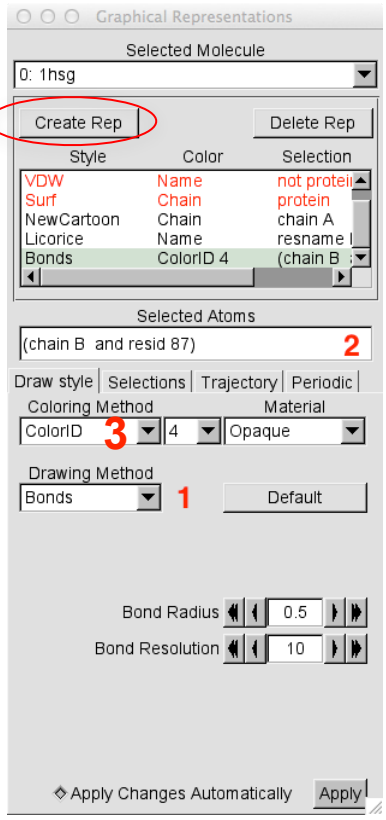
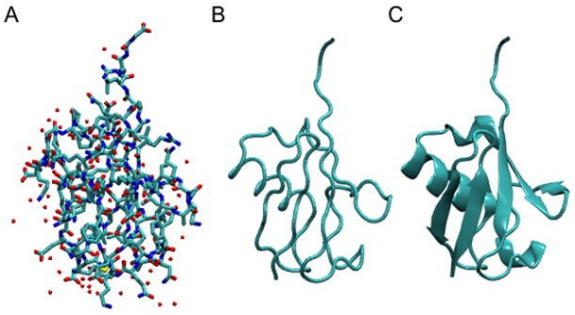


Figure 2. The VMD graphical Representation window. Note that (**1**) the *drawing method* defines which graphical representation is used and (**2**) the selection determines which part of the molecule is drawn, and (**3**) defines the color it is displayed with. You are encouraged to explore different drawing styles (*Drawing Methods* - labeled **1**) including *Licorice*, *Tube* and *NewCartoon* (see below for examples A-C).



Also try different selections by entering text in the (*Selected Atoms* box - labeled **2**). Some examples to try include:

```
chain A and backbone
resname ASP
within 5 of resname MK1
```

Using Atom Selections

Now type "`protein`" in the *Selected Atoms* text box (labeled **2** in Figure 2) and show the protein using the **Cartoon** representation and color by **chain** (see label **3** in Figure 2.)

Lets add a new representation by clicking the **"Create Rep"** (circled in Figure 2) and using the selection text **"not protein and not water"**

Add more representations (by clicking the **"Create Rep"** button) and hiding (by double clicking) or deleting previous ones (with the "**Delete Rep**" button) to explore different representations for both the ligand and the protein.

NOTE: you can use the residue name of the ligand "resname MK1" to select just the ligand.

Water molecules have the residue name HOH. Select and display all water molecules as red spheres. If you think the spheres are too big, how would you reduce their size?

Q3: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

Q4: There is a conserved water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have (see note below)?

NOTE: From the **VMD Main** window click **Mouse > Label > Atoms** and then click on the water in question to display its residue number. A short cut is to press the #1 key when your mouse is active in the OpenGL window.

Now you should be able to produce an image similar or even superior to **Figure 1A** and save it to an image file on disk with **VMD Main** window, **File > Render > Start Rendering**.

NOTE: You can chose different rendering engines including Tachyon (internal), which is commonly used for publication quality images.

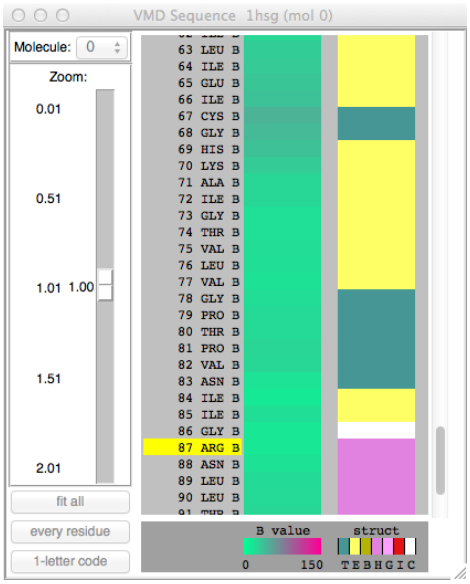
Optional: Generate and save a figure clearly showing the two distinct chains of HIV-protease along with the ligand. You might also consider showing the catalytic residues ASP 25 in each chain (we recommend Licorice for these side-chains). Email this figure to bjgrant@umich.edu for grading.

Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and subtrates, could enter the binding site?

Sequence Viewer Extension

When dealing with a protein for the first time, it is very useful to be able to find and display different amino acids quickly. The sequence viewer extension allows viewing of the protein sequence, as well as to easily pick and display one or more residues of interest.

To launch the Sequence Viewer click **VMD Main** window, **Extensions > Analysis > Sequence Viewer**. The different color scales beside the sequence correspond to the B-factor and Secondary structure type (the major ones being Extended (beta) in yellow and Helix in purple).



Q5: List the reside numbers and primary sequence in one-letter code of helix positions in chain B.

Secondary Structure codes used by STRIDE.

Letter Code	Secondary Structure
T	Turn
E	Extended conformation (β-sheets)
B	Isolated bridge
H	Alpha helix
G	3–10 helix
I	Pi helix
C	Coil

Q6: As you have hopefully observed HIV protease is a homodimer (i.e. it is composed of two identical chains). With the aid of the graphic display and the sequence viewer extension can you identify secondary structure elements that are likely to only form in the dimer rather than the monomer?

Section 3: *In silico* docking of drugs to HIV-1 protease

Docking algorithms require each atom to have a charge and an atom type that describes its properties. However, typical PDB structures don't contain this information. We therefore have to 'prep' the protein and ligand files to include these values along with their atomic coordinates. All this will be done in a tool called **AutoDock Tools** (**adt**).

3.1 Prepare the protein

The PDB file (1hsg.pdb) contains protein, ligand and water oxygen atoms. First we have to extract just the protein atoms, which are the lines that start with the keyword ATOM. Each protein chain is terminated with a line that starts with TER. (If you would like to confirm this, open 1hsg.pdb in a text editor and scroll through the text.) You could copy and paste the protein portion of the original PDB file into a new file or use the Unix command below to achieve the same thing:

```
> mkdir ~/lab3                    ## Create/make a new folder/directory
> cd ~/lab3                       ## Change to this 'working' directory

> egrep "^(ATOM|TER)" ~/Downloads/1hsg.pdb > ~/lab3/1hsg_protein.pdb
```

Launch AutoDock Tools (ADT) using the command `adt`

```
> adt &
```

Load the protein using **File > Read Molecule**. Select `1hsg_protein.pdb`. Click **Open**.

Note: In ADT, you can translate the molecule by clicking and holding down the right mouse button while moving the mouse, rotate by clicking and holding down the middle button and zoom in/out by using the scroll wheel of the mouse.

Bonds and atoms are shown in white. For better visualization, color the structure by atom type - **Color > By Atom Type**. Click **All Geometries** and then **OK**.

Q7: Can you locate the binding site visually? Note that crystal structures normally lack hydrogen atoms, why?

As we have already noted (remind me if we haven't) crystal structures normally lack hydrogen atoms. However, hydrogen atoms are required for appropriate treatment of electrostatics during docking so we need to add hydrogen atoms to the structure using **Edit > Hydrogen > Add**. Click **OK**. You should see a lot of white dashes where the hydrogens were added.

Now we need to get ADT to assign charges and atom type to each atom in the protein. We do this with **Grid > Macromolecule > Choose....** Choose **1hsg_protein** in the popup window and click Select Molecule.
ADT will merge non-polar hydrogens, assign charges and prompt you to save the macromolecule.

Click Save. This will create a file called **1hsg_protein.pdbqt** in the current folder. Open this in a text editor and look at the last two columns - these should be the charge and atom type for each atom.

Q8: Look at the charges. Does it make sense (*e.g.* based on your knowledge of the physiochemical properties of amino acids)?

Now, we need to define the 3D search space were ligand docking will be attempted. Remember the binding site that you observed in one of the earlier steps. Ideally, if we do not know the binding site, we will either define a box that encloses the whole protein or perhaps a specific region of the protein. In this case, to speed up the docking process, we will define a search space that encloses the known binding site.

To define the box, use **Grid > Grid Box....** This will draw a box with opposite faces colored in red, green and blue. Fiddle with the dials and see how you can enclose regions of the protein. In this instance we will use a Spacing (angstrom) of 1Å (this is essentially a scaling factor). So set this dial to 1.000. So that we all get consistent results, let us set the (x, y, z) center as (16, 25, 4) and the number of points in (x, y, z)-dimension as (30, 30, 30). Make a note of these values. We will need it later.

Close the Grid Options dialog by clicking **File > Close** w/out saving.

That is all we need to do with the protein file. Delete it from the display using **Edit > Delete > Delete Molecule** and select 1HSG_protein. Click **Delete Molecule** and **CONTINUE**.

3.2 Prepare the ligand
Like the protein, the ligand lacks hydrogen atoms. We need to add hydrogen atoms and also optionally define rotatable bonds that can be used for 'flexible docking'. You can run through this yourself using the instructions in **Appendix 1**. However, to expedite your progress we have already done this for you, at the command line type (*i.e.* back in your UNIX terminal):

```
> cd ~/lab3
> wget http://bioboot.github.io/bioinf525_w16/class-material/indinavir.pdbqt
```

This will download the ligand file for docking to your current directory.

3.3 Prepare a docking configuration file

Before we can perform the actual docking, we need to create an input file that defines the protein, ligand and the search parameters. We will create the input file in a text editor.
If you have used Unix/Linux before, open your favorite text editor (e.g. vi, emacs, nano) or use a GUI based editor such as `gedit`.

```
> gedit &
```

The input file should look something like:

```
receptor = 1hsg_protein.pdbqt
ligand = indinavir.pdbqt

num_modes = 50

out = all.pdbqt

center_x = XX
center_y = XX
center_z = XX

size_x = XX
size_y = XX
size_z = XX

seed = 2009
```

This defines your protein (`receptor`), ligand (`ligand`) number of docking modes to generate (`num_modes`). All the docked modes will be collated in a file defined by out (`all.pdbqt`). You should replace the **XX** with the center of your 3D search space (`center_x/y/z`) and the size of the box (`size_x/y/z`) that you defined in **Section 3.1** "*Prepare the protein*" above.

Save the file as `config.txt` in the folder containing the protein and ligand `.pdbqt` files.

Again if necessary you can download a pre-prepared file from here:

```
> wget http://bioboot.github.io/bioinf525_w16/class-material/config.txt
```

3.4 Docking indinavir into HIV-1 protease

For this portion of the practical, we will use a program called **Autodock Vina** [4]. Autodock Vina is a fast docking program that requires minimal user intervention and is often employed for high-throughput virtual screening. We will run it from a terminal.

Make sure you are in the folder containing `1hsg_protein.pdbqt, indinavir.pdbqt` and `config.txt` files.

Run **vina** to perform the docking. We will keep a log of all program output in a file **log.txt**

```
> vina --config config.txt --log log.txt
```

NOTE: This will take a few minutes depending on how fast your computer is. While you wait, if you are interested, read [1] for a review on HIV-1 protease structure, function and drug discovery.

Once the run is complete, you should have two new files **all.pdbqt**, which contains all the docked modes, and **log.txt**, which contains a table of calculated affinities based on AutoDock Vina's scoring function [4]. The best docked mode, according to AutoDock Vina, is the first entry in **all.pdbqt**.

In order to visualize the docks and compare to the crystal conformation of the ligand we will process the **all.pdbqt** to a PDB format file that can be loaded into VMD. To do this we will use the following commands:

```
> egrep "^(MODEL|ENDMDL|HETATM)" all.pdbqt > docking_results.pdb
> vmd -m ~/Downloads/1hsg.pdb docking_results.pdb
```

NOTE: The `–m` flag in the above VMD command enables us to load *multiple* input structures, in this case the protein, the extracted ligand and the docks.

Begin by displaying the protein in cartoon representation, indinavir in licorice (also called stick representation) and the docked conformations as licorice as well.

Cycle through the docks by clicking the playback control buttons on the lower right hand corner of the **VMD Main** window.

Q9: Qualitatively, how good are the docks? Is the crystal binding mode reproduced? Is it the best conformation according to AutoDock Vina?

NOTE: To assess the results quantitatively we could calculate the RMSD (root mean square distance) between each of the docking results and the known crystal structure (see for example **Extensions** > **Analysis** > **RMSD Calculator**). However, for most applications we wont know the answer *a priori* so the calculated binding affinity (see your **log.txt** file for details), scoring function and additional docking energy assessments become all-important.

Q10: What one part of this lab or associated lecture material is still confusing?
If appropriate please also indicate the question number from this lab instruction pdf and answer the question in the following anonymous form:
http://tinyurl.com/bioinf525-lab1-3

(Optional) Section 4: Generating the electrostatic surface of the protein

By now you might have realized that electrostatics play a very important role in docking molecules. The purpose of this optional exercise is to generate and view the electrostatic potential of HIV-1 protease and to see how a ligand interacts with this charged surface.

As we found out earlier, PDB structures do not contain charge or radii information for the atoms in the structure. However, knowledge of charges is crucial for electrostatic calculations. Hence, we need to add this information to our PDB file. To do this we will use the python script `pdb2pqr` as described below to generate a new file with charge and radii information taken from the AMBER molecular mechanics force field.

```
>  pdb2pqr —-ff=amber 1hsg_protein.pdb 1hsg_protein.pqr
```

Q11: Inspect the output file (using gedit for example), where are the new charge and radii information stored in this PQR file?

VMD has a simple plugin for computing and viewing electrostatic surfaces generated using a program called Adaptive Poisson-Boltzmann Solver (APBS) [5]. We will load our new PQR file into VMD and call the plugin extension to call APBS.

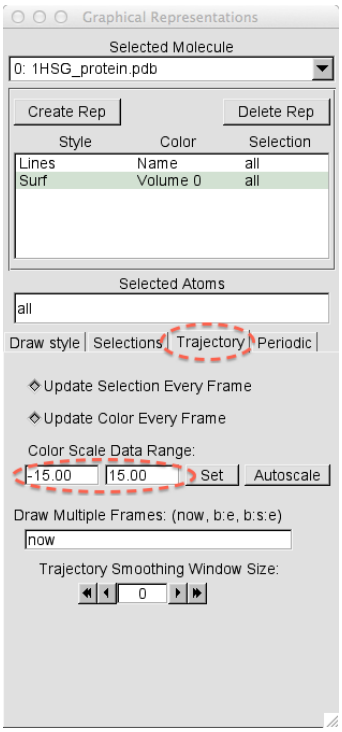
First open 1hsg_protein.pqr in VMD.

```
> vmd —pqr 1hsg_protein.pqr
```

Then open the VMD “APBS Extension” using **VMD Main** window, **Extensions > Analysis > APBS Electrostatics**. Then click **Run APBS**.

NOTE: that if a popup window appears that warns you that there are no charge or radii information in your input file then you have likely read in the wrong input file, or alternatively pdb2pqr did not work according to plan. Either way you need to go back and fix this before proceeding.

This may take a little while to run! During this time the **Run APBS** button will say **Stop APBS**. Be patient and let it run until a new window appears that will allow you to load your results (typically in a file called `prot.dx`) by accepting the defaults and clicking **OK**. You can also check in another terminal window with the `top` command whether `apbs` is still running. Ask Barry if you are unsure.



Once the calculation is done, a new small window will appear. In this new window click on the default "**load files into top molecule**" and press **OK**. Note that nothing will actually change in your graphics window until you add a suitable new *Graphical Representation.*

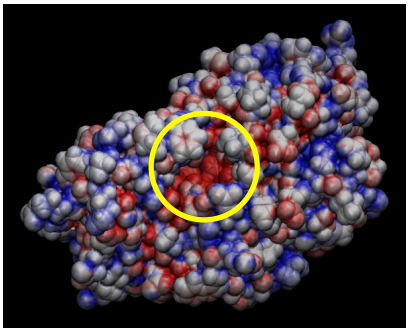
To do this switch to the *Graphical Representations* window and click **Create Rep** to add a new representation for our display of the electrostatic surface. For this representation click **Surf** as the *Drawing Method* and chose *Coloring Method* **Volume**. Now we need to set the **Color Scale Data Range** option under the *Trajectory* tab to a more suitable range (see Figure to right).

Start with the -5 and +5 for the negative and positive threshold. Make the positive threshold higher and negative threshold lower and observe what happens to the protein surface color.

If you don't already have the original 1hsg PDB file already loaded in your current VMD session you can load the ligand indinavir separately. Either way the ligand in licorice/stick representation.

Q12: Show/Hide the electrostatic surface (by double clicking its entry in the Graphical Representations window) and compare with the protein atoms. Does it match up?

Q13: Does anything on the protein surface (most notably in the binding site) stand out?



Q14: Why do you think the base of the binding site is largely negative (red in this figure)? How is this related to catalysis in this case? Does the protein surface charge (or lack thereof) correspond to complementary regions in the ligand?

Concluding remarks

In this practical, we have looked at how small molecules could be docked into a protein. This approach is widely used to detect binding sites and also to screen a library of small molecules to find potential drugs that could bind to a known binding site. Scoring functions play an important role in identifying a "good dock" and as such remains an area of active research. Several other considerations are worth noting including water molecules and ions in the binding site, flexibility of binding site residues, etc. Some docking programs (including AutoDock Vina) allow you to define a subset of flexible sidechains. Whilst this permits binding site rearrangement to accommodate distinct ligands, the computational search space increases many fold. Defining water molecules that are important in the binding site also remains an area of active research. In this practical, we have considered only protein-ligand docking. Protein-protein docking is also widely used, but not considered here due to the significantly higher search space that has to be considered. As the number of solved protein structures continues to grow, *in silico* docking will play an increasingly important role in the drug discovery process.

References:

1. Brik A, Wong CH. "HIV-1 protease: mechanism and drug discovery". Org. Biomol. Chem. (2003) 1:5–14.
   http://pubs.rsc.org/en/Content/ArticleLanding/2003/OB/b208248a

2. Wensing AM, van Maarseveen NM, Nijhuis M. "Fifteen years of HIV Protease Inhibitors: raising the barrier to resistance." Antiviral Res. (2009).
   http://www.sciencedirect.com/science/article/pii/S0166354209004902

3. Chen Z, Li Y, Chen E, Hall DL, Darke PL, Culberson C, Shafer JA, Kuo LC. "Crystal structure at 1.9-A resolution of human immunodeficiency virus (HIV) II protease complexed with L-735,524, an orally bioavailable inhibitor of the HIV proteases." J. Biol. Chem. (1994) 269:26344-26348.
   http://www.jbc.org/content/269/42/26344.long

4. Trott O, Olson AJ. "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading", J. Comput. Chem. (2009).
   http://onlinelibrary.wiley.com/doi/10.1002/jcc.21334/abstract

5. Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA. "Electrostatics of nanosystems: application to microtubules and the ribosome." Proc. Natl. Acad. Sci. (2001) 98:10037-10041.
   http://www.pnas.org/content/98/18/10037

Appendix I:

First, extract the ligand atoms from the PDB `1hsg.pdb`. As mentioned in Exercise 2, the ligand residue name for indinavir in the PDB file is MK1 and the lines start with the keyword HETATM for heteroatoms. In a terminal type:

```
> grep "^HETATM.*MK1" 1hsg.pdb > indinavir.pdb
```

Load the ligand structure into ADT using **File > Read Molecule** and select **indinavir.pdb**

Again, color by atom type. **Color > By Atom Type**. Click **All Geometries** and then **OK**.

Now we have to add polar hydrogen atoms. Add all hydrogen atoms initially. Non polar hydrogens will be merged in the next step. **Edit > Hydrogens > Add**. Select **All Hydrogens** and click OK.

Define this as the ligand in ADT so that ADT assigns partial charges and sets rotatable ligand bonds using **Ligand > Input > Choose....** Select `indinavir` and click **Select Molecule for AutoDock4**. You should see a message that confirms that non-polar hydrogens have been merged, charges added and rotatable bond detected. Click **OK**. The ligand will now have only polar hydrogens.

To check the rotatable bonds detected by ADT, go to **Ligand > Torsion Tree > Choose Torsions....** You should see 14 rotatable bonds.

Click **Done** and save the ligand file in PDBQT format. Do this using **Ligand > Output > Save as PDBQT....** Click **Save** to save the file as **indinavir.pdbqt**.
Quit ADT using **File > Exit > OK**.

Apendex II: Useful resources/links

This document and all associated input and output files are available online at http://thegrantlab.org/teaching/teaching.html under "**Structural Bioinformatics (BI527)**". All required software is freely available from the sites listed below.

VMD	http://www.ks.uiuc.edu/Research/vmd/
PyMol	http://www.pymolwiki.org/index.php/Main_Page
AutoDock Tools	http://autodock.scripps.edu
AutoDock Vina	http://vina.scripps.edu
APBS	http://www.poissonboltzmann.org/apbs

Appendix III: Basic Linux commands

```
ls -lrt          # list files in reverse order of time.
cd dir           # change directory to the directory 'dir'
pwd              # print the current working directory on the screen
rm file          # delete (remove) 'file'
mv file newfile  # rename file to newfile
cat file         # print the contents of file to the screen
more file        # print file to the screen with more navigation
mkdir dirname    # make a new directory/folder
```