Using AutoLigand with AutoDockTools: A Tutorial

Written by Rodney M. Harris

The Scripps Research Institute
Molecular Graphics Laboratory
10550 N. Torrey Pines Rd.
La Jolla, California 92037-1000
USA

16 December 2009

Contents

USING AUTOLIGAND WITH AUTODOCKTOOLS: A TUTORIAL..... 1

Written by Rodney M. Harris 1

The Scripps Research Institute
Molecular Graphics Laboratory
10550 N. Torrey Pines Rd.
La Jolla, California 92037-1000
USA 1

16 December 2009 1

Contents.....2

Introduction.....4

FAQ – Frequently Asked Questions.....5

Overview: Using AutoLigand.....6

Exercise One: PDB Files Are Not Perfect: Editing a PDB File.....7

Procedure:.....8

Exercise Two: Preparing the Macromolecule. 9

Procedure:.....9

Exercise Three: Preparing the Grid Parameter File. 10

Procedure:.....10

Exercise Four: Starting AutoGrid 4.....13

Procedure:.....13

Exercise Five: Running AutoLigand via Command Line.....14

Exercise Six: Running AutoLigand via the GUI.....17

Exercise Seven: Visualizations.... 19

Files for Exercises:.....20

Input Files:.....20

Results Files.....20

Macromolecule.....20

AutoGrid..... 20

AutoLigand..... 20

Appendix 1: PMV Basics.....21

Modifier.....21
Pick.....21
Rotate.....21
Shift.....21
Key.....21
R.....21

Appendix 2: Dashboard Widget.....23

Introduction

AutoLigand is a tool used to identify and characterize ligand binding sites. This tutorial will show you how to use **AutoLigand**. We will use both a command line interface and a Graphical User Interface called **AutoDockTools**, or **ADT**, to set up and run **AutoLigand** jobs. Previous experience with **ADT** and **AutoGrid** is recommended.

FAQ – Frequently Asked Questions

1. *Where should I start **ADT**?*

   You should always start **ADT** in the same directory as the macromolecule and map files. You can start **ADT** from the command line in a Terminal by typing "**adt**" and pressing <Return> or <Enter>.

2. *Should I always add hydrogens?*

   Yes. For the macromolecule you should always add hydrogens, compute Gasteiger charges and then you must merge the non-polar hydrogens. Polar hydrogens are hydrogens that are bonded to electronegative atoms like oxygen and nitrogen. Non-polar hydrogens are hydrogens bonded to carbon atoms.

3. *How many **AutoGrid** grid maps do I need?*

   To run **AutoLigand**, you only need C, OA and HD maps plus the electrostatics 'e' map and the desolvation 'd' map.

4. *How big should the **AutoGrid** grid box be?*

   The grid volume should be large enough to cover the entire receptor with a few extra Å in each dimension. **Important**: **the grids should be done at a 1 Å spacing.** If the protein is very large, then you can break it up into overlapping grids and run **AutoLigand** on each of these grid sets, *e.g.* one covering the top half and one covering the lower half.

Overview: Using AutoLigand

AutoLigand is run to identify and/or characterize ligand binding sites and requires a 1 Å grid and map files generated by **AutoGrid**. The run result is a fill volume of continuously connected points output in a PDB file format.

When the **AutoLigand** GUI is used, a single fill is generated starting at the user specified point. The starting point can either be selected by clicking on an atom and adjusting the cross-hairs with slider bars, or entering coordinates in the x y z boxes.

When **AutoLigand** is run from the command line, the code will run a single fill if input coordinates are given or scan the entire grid box and output the ten best fills if no input starting point is given. By using different fill sizes and plotting the total energy per volume of the fills, optimal binding sites on the receptor can be identified.

It is important to note that since **AutoLigand** is run using a 1 Å grid and docking is normally run using a 0.375 Å grid, it is best to set up a subdirectory for the **AutoLigand** map files and results. By copying the same receptor pdb file to that subdirectory, all **AutoLigand** results can be viewed in the same coordinate frame as docking results.

Exercise One: PDB Files Are Not Perfect: Editing a PDB File

In the exercises which follow, we show you how to find and generate fill volumes for ligand binding sites. To run **AutoLigand** requires the following input files:

- a PDBQT file for the receptor;
- a grid parameter file (GPF) for the AutoGrid calculation

Protein Data Bank (PDB) files may have a variety of problems that need to be corrected before they can be used in AutoGrid. These potential problems include missing atoms, added waters, more than one molecule, chain breaks, alternate locations *etc.*

AutoDockTools (ADT) is built on the Python Molecule Viewer (PMV), and has an evolving set of tools designed to solve these kinds of problems. In particular, two modules, **editCommands** and **repairCommands**, contain many useful tools that permit you to add or remove hydrogens, modify histidine protonation, *etc.* ADT cannot do everything, though, and it is sometimes necessary to use other software to clean up and check the quality of the structures. See the list of resources at the AutoDock web site for a list of some of these tools (http://autodock.scripps.edu/resources).

Note: Because of time (and sanity) considerations, in this tutorial we only demonstrate removing hetero atoms from the receptor molecule. Nevertheless, any other problems in the receptor must be corrected also. Remember that all hydrogens must always be added prior to using ADT specific commands to format the receptor.

Note, that in some cases, you may want to leave selected waters, substrates, metal ions, etc. in the PDB file to influence the **AutoLigand** results. In this exercise we will start with a clean molecule to look for binding sites.

In this exercise, you will work on the receptor macromolecule. You will learn how to remove non-receptor molecules, how to add the hydrogens that AutoGrid expects, and how to save the modified result.

First, look at your PDB file by starting ADT.

Procedure:

1. Read in the receptor molecule:

Note: Commands available via the **Dashboard** can also be invoked using the menus at the top of the PMV GUI. Here you could use this menu sequence to open the **file browser**:
File->**Read Molecule**

See **Appendix 2** for information about the Dashboard.

Position the cursor over **PMV Molecules** in the **Dashboard** (Appendix 2) and click with the right mouse button. This will open a file browser, showing all the files in the current directory. Select **3B7E.pdb** and click on **Open**.

This loads a **Molecule** named '3B7E' into ADT. 3B7E is the N1 receptor in swine flu (more details about this molecule can be found on the PDB web-site). Repeat the procedure and load also the files 3b7e_WATERS.pdb and 3b7e_zanamivir.pdb.

The bonds between bonded atoms are represented as lines while non-bonded atoms, such as metal ions and oxygen atoms of water molecules, are shown as small squares. The non-bonded atoms you see here are the oxygen atoms of waters that were present in the crystal structure.

2. Color 3B7E by atom type

Note: Each row in the Dashboard is linked to a single entity. The entity is either **PMV Molecules** which is the group comprised of all the Molecules in the Viewer, a single **Molecule**, a single **Chain**, a single **Residue** or a single **Atom**. The diamonds on the right side of the Dashboard are used to color the displayed geometries for that row's entity by different coloring schemes.

In the **PMV Molecules** row of the Dashboard, click on the **diamond** under **Atom** to color the lines by Atom Type.

Now the lines representing the atoms are colored according to the chemical element, as follows:

- ❍ Carbons that are aliphatic (C) - white,
- ● Carbons that are aromatic (A) - green,
- ● Nitrogens (N) - blue,
- ● Oxygens (O) - red,
- ● Sulfurs (S) - yellow,
- ● Hydrogens (H) – cyan.

3. Color by Chain

In the 3B7E row of the Dashboard, click on the **diamond** under **Chain**. This colors the protein in blue. To highlight the bound zanamivir (the Relenza drug) and the water molecules, click on the **diamond** under **Atom** in the Dashboard.

Once you finished analyzing the structures, hide the ligand and the waters by clicking on the **Show/Hide** boxes in the Dashboard.

Exercise Two: Preparing the Macromolecule.

Procedure:

CAUTION: The **AutoGrid** calculation must be based on the **rigid residues only**.

1. On the Dashboard, click on the **Sel.** box corresponding to the protein 3e7b.

2. **Edit** → **Hydrogens** →**Add**

   Click: **OK** to accept defaults.

3. **Grid** → **Macromolecule** →**Choose…**

   Select 3b7e

   Selecting the macromolecule in this way causes the following sequence of initialization steps to be carried out automatically:

   - ADT checks that the molecule has charges. If not, it adds Gasteiger charges to each atom. Remember that all hydrogens must be added to the macromolecule before it is chosen. If the molecule already had charges, ADT would ask if you want to preserve the input charges instead of adding Gasteiger charges.

   - ADT merges non-polar hydrogens unless the user preference **adt_automergeNPHS** is set not to do so.

   - ADT also determines the types of atoms in the macromolecule. AutoDock 4 can accommodate any number of atom types in the macromolecule.

Note: If ADT makes any modifications to the molecule you opened, a file browser will open for you to specify a file name. Type in a name + "**.pdbqt**" and click on **Save**.

Click **OK** on the WARNING dialog box. The form should be pre-filled in as **3b7e.pdbqt.** Then click on **Save**.

Exercise Three: Preparing the Grid Parameter File.

The grid parameter file tells **AutoGrid 4** which receptor to compute the potentials around, the types of maps to compute and the location and extent of those maps. In addition it may specify a custom library of pairwise potential energy parameters. For **AutoLigand** only the C, OA, and HD maps are used plus an electrostatics map and a separate desolvation map.

Procedure:

1. **Grid** → **Set Map Types** → **Directly...**

   Edit the selections in the **AutoGpf Ligand** widget to include only C, HD, and OA.

   Close this widget with the **Accept** button.

2. **Grid** → **Grid Box...**

   Opens the **Grid Options Widget**. First a brief tour of this widget:

   - This has menu buttons at the top: **File**, **Center**, **View** and **Help**.

     → **File**

     This menu lets you close the Grid Options Widget, which also causes the grid box to disappear. You can **Close saving current values** to keep your changes or **Close w/out saving** to forget your changes.

     → **Center**

     This menu lets you set the center of the grid box in four ways: → **Pick an atom**, → **Center on ligand**, → **Center on macromolecule** or → **On a named atom**.

→ **View**

This menu allows you to change the visibility of the box using **Show box**, and whether it is displayed as lines or faces, using **Show box as lines**. This menu also allows you to show or hide the center marker using **Show center marker** and to adjust its size using **Adjust marker size**.

- The "Grid Options Widget" displays the Current Total Grid Points per map. This tells you how big each grid map will be: $(n_x + 1) \times (n_y + 1) \times (n_z + 1)$, where n_x is the number of grid points in the *x*-dimension, *etc.*

Note: clicking with the right mouse button on a **thumbwheel** widget opens a box that allows you to type in the desired value directly. Like many other entry fields in ADT, this updates only when you press <**Enter**>.

- This has 3 **thumbwheel** widgets which let you change the number of points in the *x*, *y* and *z* dimensions. The default settings are 40, 40, 40, which makes the total number of grid points per map 68921 because AutoGrid always adds one in each dimension.

- You will also notice it has a **thumbwheel** that lets you adjust the spacing between the grid points.

- There are also entries and thumbwheels that let you change the location of the **center** of the grid.

The **number of points** in each dimension can be adjusted up to 126. AutoGrid requires that the input number of grid points be an even number. It then actually adds one point in each dimension, since AutoGrid and AutoDock need a central grid point.

The **spacing** between grid points can be adjusted with another thumbwheel. The default value is 0.375 Å between grid points, which is about a quarter of the length of a carbon-carbon single bond. Grid spacing values of up to 1.0 Å can be used when a large volume is to be investigated. If you were to need grid spacing values larger than this, you could edit the GPF in a text editor before running AutoGrid.

For this exercise:

The goal is to generate a grid box that encompasses the entire receptor molecule at a 1 Å spacing. First, adjust the spacing thumbwheel to 1Å. Next, adjust the number of points of the x y z wheels to cover the entire molecule: **60**, **60**, **60** should be enough in this case. Note, that the box could be adjusted to exclude parts of the molecule (such as protein-protein interfaces) where you do not want **AutoLigand** to search.

Close this widget by clicking **File** → **Close saving current**.

3. **Grid** → **Output** → **Save GPF…**

Opens a file browser allowing the user to specify the name of the grid parameter file. The convention is to use '.gpf' as the extension.

Write the GPF as '**3b7e.gpf'**

Be sure to put in the proper path

Exercise Four: Starting AutoGrid 4

In general, you should know:

- **AutoGrid 4** (and **AutoLigand)** must be run in the directory where the macromolecule and parameter files are to be found.
- The named files in the GPF must not include pathnames.

Procedure:

1. **Run** → **Run AutoGrid…**

   In the **Run AutoGrid** widget that opens, change **autogrid3** to **autogrid4** and press the <**Return**> key.

2. **Launch**

   Starts the AutoGrid 4 job. On most platforms, this opens a Process Manager that allows you to see specifics about current AutoGrid and AutoDock jobs. It is a limited process manager that you can use to terminate an AutoGrid or AutoDock process by selecting its entry. You are asked if you really want to kill it.

   (The AutoGrid 4 calculation takes about 2 minutes on a Mac.)

Please note that you can easily start a job from the command line by cd in to the directory containing the input files and typing:

```
% autogrid4 -p 3b7e.gpf -l 3b7e.glg
```

Exercise Five: Running AutoLigand via Command Line.

For this exercise, we will open a new terminal window and change to the directory where the map files are located. To run **AutoLigand** under Windows and find the ten best binding sites, type the following:

```
c:\Python\python.exe [PATH]AutoLigand.py 3b7e 90
```

where the 3b7e is the base name of the map files and 90 is the number of fill points to use (for this exercise). [Note, you may need to adjust your PATH to the AutoLigand.py command depending on how you have set up your computer.]

The command takes 10 or more minutes to run depending on the number of fill points you have requested (500 points can take hours).

The code generates up to ten FILL files and one Results file. Some of the top ten fill results may collapse to the same answer, thus reducing the number of fills to less than ten. The results are listed in rank order from best Total Energy per Volume of each of the fills generated.

This exercise should generate nine fills for the input size of 90 points. Each fill file is output as a PDB file in the following format:

```
FILL_90out1.pdb
```

where 90 is the number of points in the fill and out# is the rank of that fill. To examine the fill volumes, go back to the ADT window and do the following:

1. **File** → **Read Molecule** → `FILL_90out1.pdb`

   This opens the first fill volume found by **AutoLigand**. The default in ADT is to open in Line mode. To make it more visible, we will change it to a modified CPK mode.

   Left click on the circle under CPK in the FILL_90out1 line on the Dashboard. Then, Right Click this circle to open a size widgit. Move the Scale Factor Thumbwheel to 0.3 and click OK. Lastly, click on the diamond under Atom in the FILL_90out1 line to color the spheres by atom type.

2. **File** → **Read Molecule** → `FILL_90out2.pdb`

This fill overlaps most of fill 1. Nearly identical fills can be generated when the fills are started from different points at opposite ends.

3. Open the rest of the fills:

   File → **Read Molecule** → **`FILL_90outX.pdb`**

   Fills 3 and 6 share some overlap, but are much different fills. Fills 4 and 5 are almost identical. And fills 7 and 8 share some overlap.

   Fills 1,2 and 4,5 are internal with a lot of H-bond donors and acceptors, which is an indication that they might be internal structural water sites. To check to see if they are structural water sites, we will open the original PDB file and highlight the waters.

4. **File** → **Read Molecule** → `3B7E.pdb`

   Note upper case letters which is the unmodified PDB file. Next, we will select the waters and highlight them.

5. **Select** → **Select From String**

   Type: HOH* in the Residue box and then click Add and then Dismiss the widgit. All of the waters have been selected and should have yellow crosses on them.
   Next, we will display the waters in CPK.

6. **Display** → **CPK**

   Change the Scale Factor on the widgit 0.5 so that they will stand out from the fill volumes. Then, shut off the selection (the pencil erraser icon button).

   Focus the view on fills 1 and 2. It can be seen that the fills overlap 9 water molecules. The grey square on the dashboard can be used to hide/show the fills, which is useful in determining overlaps.

Fills 3 and 6 overlap the Zanamivir ligand molecule from the crystal structure.

Now, focus the view on fills 4 and 5. This internal site contains 8 water molecules. Thus, the two internal sites are not likely to be ligand binding sites.

Fills 7 and 8 overlap a bound Glycerol in the crystal structure.

Fill 9, which has the lowest total energy per volume, looks like a surface patch and may not be a real ligand binding site.

To shut off the water molucules, click on the CPK button on the 3B7E line, and click it again to shut it off.

Exercise Six: Running AutoLigand via the GUI.

From exercise Five, we found the ligand binding site. In this exercise we will use the Graphical User Interface (GUI) to run more fills starting only at the ligand binding site.

Restart ADT and open 3b7e.pdbqt.

1. **File** → **Read Molecule** → `3b7e.pdbqt`

   Select color by atom types by clicking on the diamond under Atom in the 3b7e row.

   Next, we will open the zanamivir ligand PDB file. [Note, to non-TSRI class users: this file can be generated from the original PDB file by deleting all lines but the ZMR B hetero-atoms – `HETATM 6021` to `HETATM 6043` and saving as zanamivir.pdb]

2. **File** → **Read Molecule** → `3b7e_zanamivir.pdb`

   Highlight this ligand by changing it to ball and stick. Click the circle under S&B in the zanamivir row and then color by atom type by clicking on the diamond under Atom in the zanamivir row.

   Now, we will begin making different size **AutoLigand** fills at this site.

3. **Compute** → **AutoLigand** → **Run AutoLigand...**

   This opens the widgit and shows a box around the space which can be searched. The File Base Name is pre-filled with the name of the map files (3b7e in this case) and set to 100 points. The start location is defaulted to the center of the grid box or the last start point used. There are three ways to change the starting point:

   1. Click on an atom next to where you wish to start filling.
   2. Enter the X Y Z coordinates
   3. Adjust the slider bars

We want to start making a few different size fills starting at a point next to the ligand.

Click on the oxygen atom in the center ring of the ligand (any point near this would work the same, but this atom is easy to find). The coordinates and slider bars automatically adjust. Note, that the ligand is not really there for **AutoLigand** calculations as the map files were generated without the ligand in place. Click **OK** to start the calculation. [**Important Note:** if other fills are run at the same size, the fills will be overwritten. Use different number of points, or rename the FILL file if you wish to run the GUI again and keep your first results.]

Now, run a few different size fills.

4. **Compute** → **AutoLigand** → **Run AutoLigand...**

Change the Number of Points to 150. Note, that the start point is located at the same position as the last run (as long as you are still in the same ADT session). Click **OK** to start the calculation.

Rerun again for sizes 200 and 250.

To see how the fills fit in the pocket, we will make a molecular surface about the receptor molecule. Click the circle under the MS column in the 3b7e row.

You can see as larger fill sizes are used, the fill volume is forced to push out of the binding pocket and move into regions of less affinity. A Results file is generated for each run of the GUI (note again that if the same fill size is used, these files are overwritten). By looking at the energy per volume (EPV) of fills 100 to 250, you can see that the EPV stays about the same for fills 100 and 150, but drops off after that.

Exercise Seven: Visualizations….

To visualize the results, a molecular surface can be placed around the fill volumes. Click the circle under the MS column in the FILL_150out1 row. Then click color by Atom type. The DejaVu GUI button can be used to modify the fill result as needed.

One other important visualization that can be seen is to color by affinity values per point in the fill. This demonstrates that ligand binding sites consist of "hot points" connected by lesser affinity structure.

Select FILL_150out1 (use the select square on the dashboard).

Color → by Properties

Click cpk and **OK**, then click the Atom button and use scroll bar to move down to bottom of list and select temperature Factor. Next, click the >> button in the center at the bottom of the widgit, then click OK. This affinity scale is from red (best) to blue (worst) and shows three regions of higher affinity connected by lower affinity structure in this example.

Files for Exercises:

Input Files:

```
3b7e.pdb
3b7e_zanamivir.pdb
```

Results Files

Macromolecule

```
3b7e.pdbqt
```

AutoGrid

```
3b7e.gpf
3b7e.glg
3b7e.*.map
3b7e.maps.fld,3b7e.maps.xyz
```

AutoLigand

```
3b7e_*Results.txt
FILL_*out*.pdb
```

Appendix 1: PMV Basics

You might have a three-button mouse. If so, the mouse buttons can be used alone or with a modifier key to perform different operations. To zoom the molecule (make the molecule look bigger or smaller) in the viewer window, press and hold down the <Shift> key and then click and drag with the middle mouse button. To summarize what the mouse buttons do:

On Apple Computers:.

option: Rotate

command: Translate left/right

shift+option: Scale or Zoom

shift+command: Translate in/out

Modifier	Left	Middle	Right
None	*Pick*	*Rotate*	*Translate left/right (X) and up/down (Y)*
Shift	*Select*	*Scale or Zoom*	*Translate in/out (Z)*

Note: as you translate a molecule out in the Z dimension, it will disappear into **fog** which is used for depth-cueing.

You can also press the following keys in the viewer window to change the view of the molecule:

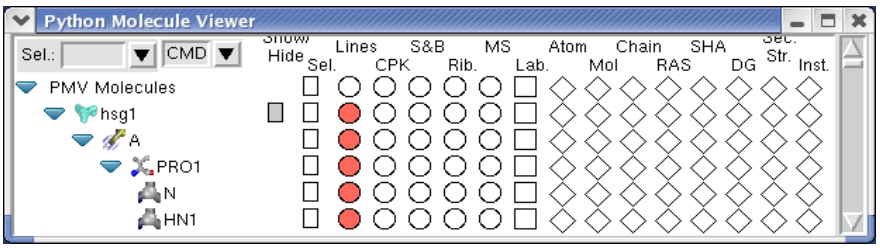
Key	Action
R	*Reset view*
N	*Normalize – scale molecule(s) so all visible molecules fit in the viewer*
C	*Center on the center of gravity of all the molecules*
D	*Toggle on/off Depth-cueing (blends molecule into background farther away)*
T	*Toggle between transform root (i.e. scene) and transform the*

	Viewer's current object.

Appendix 2: Dashboard Widget

Select entry + **Command** Buttons ⟶

Tree Widget



Note: Clicking on a **sh**ape - rectangle, circle, square or diamond - under a command causes the command linked to the shape to be applied to each node in the corresponding row. If the shape is off (colored white), the command will be applied to nodes and the shape will be colored red. If the shape is on (colored red), clicking on the command button will undo the command and the shape will be colored white. Circles are used for **display** commands, squares for **label** commands and diamonds for **color** commands. Coloring can be replaced by a different coloring scheme but cannot be undone. The gray rectangle is used for **show/hide** and the white rectangle for **select**.

The **Tree Widget** on the left lists all molecules currently loaded in PMV. Click on the arrows to navigate between molecules, chains, residues and atoms. Clicking on a **shape** in one of the columns in the right section executes the PMV command corresponding to the label at the top of the column on the group of nodes corresponding to the row. There 16 different commands that can be executed this way - gray rectangle (Show/Hide), select/unselect (**Sel**.), display lines (**Lines**), display CPK (**CPK**), display sticks and balls (**S&B**), display secondary structure (**Rib.**), display molecular surface (**MS**), display labels (**Lab**.), color by atom type (**Atom**), color by molecule (**Mol**), color by chain (**Chain**), color by residue according Rasmol (**RAS**), color by residue according Shapely (**SHA**), color according to David Goodsell colors (**DG**), color by secondary structure element type (**Sec.Str**.) and color by instance (**Inst**).

To help users see the connection between molecular fragments and PMV commands, a crosshair is drawn when cursor is inside the Dashboard widget.

Right-clicking on a shape displays an input parameter panel for the command and allows the user to customize specific input parameters for the command.

Note: A **selection in the Tree** is used to build a group of nodes to be the target for commands linked to shapes. It is not the same as the current selection in the Viewer. It can be selected using the appropriate rectangles….

The **Sel:** entry in the top left corner of the Dashboard can be used to select entries in the Tree using a Pmv compound selector. Nodes matching the specified string will be selected. Selected nodes are outlined with a yellow selection box. When a shape is clicked for a selected node, the corresponding command is applied to all currently selected nodes. Hovering over this entry shows samples of the required syntax.

The option menu on the top allows the user to specify whether commands should be applied to the backbone atoms only (**BB**), the side chain atoms only (**SC**), the sidechain atoms and **CA** atoms (**SC+CA**) or the full molecular fragment (**ALL**). This setting can be overridden for each column (**CMD**).

Click on the gray rectangle under **Show/Hide**. Notice that the molecule in the viewer disappears. Click on the same rectangle again to redisplay it. Click on the rectangle under the **Sel** level to select or deselect all. Experiment by clicking on each of the other buttons. These are short cuts to a basic set of commands for displaying and coloring various molecular representations.