Using AutoDock 4 and Vina with AutoDockTools: A Tutorial

Written by Ruth Huey, Garrett M. Morris and Stefano Forli

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

16 December 2009

Contents

USING AUTODOCK 4 AND VINA WITH AUTODOCKTOOLS: A TUTORIAL..... 1

Contents.....2

Introduction.....4

FAQ – Frequently Asked Questions.....5

Overview: What is a Docking?.....7

AUTODOCK Procedure:.....7

VINA Procedure:.....11

Exercise One: PDB Files Are Not Perfect: Editing a PDB File.....12

Procedure:.....12

Exercise Two: Preparing a Ligand for AutoDock.....15

Procedure:.....15

Exercise Three: Preparing the Macromolecule. 19

Procedure:.....19

Exercise Four: Preparing the Grid Parameter File. 20

AUTODOCK Procedure:.....20

Exercise Five: Starting AutoGrid 4.....23

Procedure:.....23

Exercise Six: Preparing the Docking Parameter File. 25

Procedure:.....25

Exercise Seven:Starting AutoDock4/AutoDockVina.....27

AUTODOCK4 Procedure:.....27

VINA Procedure:.....28

Exercise Eight: Preparing the flexible residue file (optional). 30

Procedure:.....30

Exercise Nine: More Visualizations…. 32

Procedure:.....32

Files for Exercises:.....35
Input Files:.....35
Results Files.....35
Useful Scripts in AutoDockTools/Utilities24.....35
Appendix 1: PMV Basics.....36
Appendix 2: Dashboard Widget.....38
Appendix 3: Conformation Player 40
Appendix 4: Docking Parameters.....45
Parameters common to SA, GA, GALS:.....45
Simulated Annealing Specific Parameters:.....48
Genetic Algorithm Specific Parameters:.....49
Local Search Specific Parameters:.....50
Clustering Keywords:.....51

Introduction

This tutorial will introduce you to docking using the **AutoDock** suite of programs. We will use a Graphical User Interface called **AutoDockTools**, or **ADT**, that helps a user easily set up the two molecules for docking, launches the external number crunching jobs in **AutoDock**, and when the dockings are completed also lets the user interactively visualize the docking results in 3D.

FAQ – Frequently Asked Questions

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

   You should always start **ADT** in the same directory as the macromolecule and ligand 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 both the macromolecule and the ligand, 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?*

   You need one **AutoGrid** map for every atom type in the ligand plus an electrostatics map and a desolvation map. *E.g.*: for ethanol, C_2H_5OH, you would need C, OA and HD maps plus an electrostatics 'e' map plus a desolvation 'd' map.

4. *Why should all the total charges on the residues be an integer?*

   This is because it is assumed that the residue is interchangeable with others, and that no electrons are withdrawn or received by adjacent residues. In proteins, *e.g.*, arginines should have a total charge of +1.000 if they are protonated, or 0.000 if they are neutral.

5. *How easy will it be to get good docking results?*

   In general, the more rotatable bonds in the ligand, the more difficult it will be to find good binding modes in repeated docking experiments.

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

   The grid volume should be large enough to at least allow the ligand to rotate freely, even when the ligand is in its most fully-extended conformation.

7. *Can I identify potential binding sites of a ligand on a protein with **AutoDock**?*

   Yes, if you do not know where the ligand binds, you can build a grid volume that is big enough to cover the entire surface of the protein, using a larger grid spacing than the default value of 0.375Å, and more grid points in each dimension. Then you can perform preliminary docking experiments with **AutoDock** to see if there are particular regions of the protein that are preferred by the ligand. This is sometimes referred to as *"blind docking"*.

   Then, in a second round of docking experiments, you can build smaller grids around these potential binding sites and dock in these smaller grids.

   If the protein is very large, then you can break it up into overlapping grids and dock into each of these grid sets, *e.g.* one covering the top half, one covering the lower half, and one covering the middle half. The third-party tool, BDT, automates this process; see http://autodock.scripps.edu/resources.

Overview: What is a Docking?

An **AutoDock** search for the best ways to fit two molecules together constitutes a **docking**. The docking consists of a series of independent searches or **runs** which results in a final 'pose' or **conformation** of the ligand. Each conformation is the lowest energy found during a search. An AutoDock experiment generally includes clustering of conformations based on their similarity. The two molecules to fit together and which search method to use are specified in the **docking parameter file** (DPF). The results are written to a **docking log file** (DLG). Reading a docking log or a set of docking logs into **ADT** is the first step in analyzing the results of docking experiments.

Note: the **dlg** file contains many details that are output as AutoDock parses the input files and reports what it finds. For example, when AutoDock opens each AutoGrid map, it reports opening the map file and how many data points it read in. When it parses the input ligand file, it reports building various internal data structures. After the input phase, AutoDock begins the specified number of runs. It reports which run number it is starting; it may report specifics about each generation or simulated annealing cycle. After completing the runs, AutoDock begins an analysis phase of the conformational similarity of the dockings. At the very end, it reports a summary of the time taken and outputs the words 'Successful Completion'. The level of output detail is controlled by the parameter "outlev" in the docking parameter file. For dockings using the LGA algorithm, minimal output ('outlev 0') is recommended.

In this exercise we will use the file **imatinib_1iep_receptor.dlg** from a previous AutoDock calculation. In it, the imatinib (Gleevec © Novartis), a drug mainly used for the treatment of chronic myelogenous leukemia (CML), is docked to the c-Abl kinase domain. The key results in a docking log are the docked conformations found at the end of each run, the energies of these docked structures and their similarities to each other. The similarity of docked structures is measured by computing the root-mean-square-deviation, **rmsd**, between the coordinates of the atoms and creating a **clustering** of the conformations based on these rmsd values.

A visual examination of the interactions between the docked ligand and the macromolecule is used to evaluate the **chemical reasonableness** of the docking.

AUTODOCK Procedure:

1. **Analyze** → **Dockings** → **Open…**

First, you need to choose the AutoDock log file you would like to Analyze. This command opens a file browser that lets you choose a file with the extension **.dlg** Navigate into the **Results** directory and choose **imatinib_1iep_receptor.dlg**. Improve the display of the ligand by clicking on the circle under **S&B** and the diamond under **Atom** in the **Dashboard Widget** (Appendix 2).

Note: If there were a previous Docking in the viewer, you would be asked whether you want to add this DLG to the previous Docking instance. This can be done when the same AutoGrid map files, ligand, and DPF files were used for both docking experiments. Also, WARNING messages from the docking log can be viewed in the python shell. To do so, open the python shell and type

mv.docked.warnings

Reading a docking log creates a Docking in the viewer and displays the ligand in the lowest-energy conformation. Each conformation has a docked energy value, a binding energy value, and per atom electrostatic and vdw energy values. During the docking calculation, AutoDock computes the free energy of binding and reports a detailed energy breakdown.

ADT reports how many docked conformations were read in from the DLG and tells you to how to visualize the docked conformations or 'states'.

2. **Analyze** → **Conformations** → **Load…**

This opens **ind Conformation Chooser** which gives you a concise view of the **energies** and **clusters** of the docked results.

Note: Autodock clusters by first sorting all the docked conformations from lowest energy (best docking) to highest. The best overall docked conformation is used as the 'seed' for the first cluster. Then the coordinates of the second best conformation are compared with those of the best to calculate the root-mean-square deviation between the two conformations. If the calculated rms value is smaller than the specified cutoff, which is 0.5 by default, that conformation is added to the 'bin' containing the best conformation. If not, the second becomes the reference for a second 'bin'. Then the rms between the third conformation and the 'best' is computed . If close enough, it is added to the first bin. If not it compared with the seed of the second bin and so on….

The **lower panel** lists the docked conformations for the ligand grouped according to the clustering performed at the end of the AutoDock calculation. Double clicking on an entry in this list makes that entry the **current conformation** of the ligand. This results in displaying ligand in the viewer with new coordinates. The input conformation is always the first entry in this list.

The **upper panel** displays information about the current conformation. This includes its overall rank, for example the best result is always 1_1: lowest energy cluster_best individual in cluster. **Docked Energy** is the sum of the **intermolecular** and **internal energy** components. **Cluster RMS** is the root mean square difference **rms** between this individual and the **seed** for the **cluster**. **1_1** is the seed for the first cluster so its Cluster RMS is 0.0. **Ref RMS** is the rms between the specified reference structure. If no reference structure is specified in the DPF, the input structure is used as the reference. **freeEnergy** is the sum of the intermolecular energy plus the torsion entropy penalty which is a constant times the number of rotatable bonds in the ligand, **kI** calculated from the Docked Energy.

Double click on **ind_1_1** to put the ligand in the best docked conformation. Look at the information displayed in the top panel. Scroll down through the list to see how many clusters were formed with your docking results. Notice the range in energy between the 'best' docking and the seed of the last cluster.

Note: lower energies are "**better**" and in the genetic algorithm, "**fitter**."

Drag the lower right corner down to reveal the buttons at the bottom. Click **Dismiss**.

Note: If you have read in more than one docking log into the current Docking or if the results did not include clustering, you must Recluster… to create a clustering before you can show one.

Note: if clusterings were performed using several different rms tolerance values, the menu option: **Analyze→Clusterings→Show** would open a widget containing a list of the available rms values. Be sure to click only once and to click delicately on this list to open a new interactive histogram. (Otherwise, you may get several identical windows.)

Next we will examine the **convergence** of the docking result by looking at its **clustering**. The convergence of the dockings reflects the thoroughness of the search. If the independent dockings produce a small number of conformationally-similar clusters, then the docking searches have used a large-enough number of evaluations. If not, it is advisable to repeat the docking calculation with an increased number of evaluations. The number of evaluations is determined by the DPF keyword "ga_num_evals". In general, the more torsions a ligand has, the more evaluations will be needed for each docking. If the keyword **analysis** is in the DPF, AutoDock will cluster the results. By default, AutoDock clusters docked results at **2.0Å rmsd**. This process involves ordering all of the conformations by energy, from lowest to highest. The lowest energy conformation is used as the seed for the first cluster. Next, the second conformation is compared to the first. If it is within the rmsd tolerance, it is added to the first cluster. If not, it becomes the first member of a new cluster. This process is repeated with the rest of the docked results, grouping them into families of similar conformations.

3. **Analyze → Clusterings → Show…**

Opens an **interactive histogram chart** labelled **'imatinib_1iep_receptor:rms = 2.0 clustering**'. This chart has bars which represent the clusters computed at the specified rmsd. The bars are sorted by energy of the lowest-energy conformation in that cluster and initially are colored blue.

For example, the lowest energy conformation in the first bar is **1_1** while the lowest in second bar is **2_1**. The height of the bar represents how many conformations are in that cluster. Clicking on a bar makes that cluster the current sequence for the ligand's conformation player **CP,** and its color changes to red. [See Appendix 3 for more details on the CP.]

The **Conformation # Info** Widget shows you both **refRMSD**, the rmsd between the current reference and the displayed conformation and **clRMSD**, the rmsd between the displayed conformation and the lowest energy conformation in this cluster. You can set the reference coordinates to those of any of the docked conformations when it is the current conformation. When viewing clustering results, this is

especially useful because it allows you to examine the rmsd between cluster members. To do this, choose a cluster and use the arrow key to step forward to its lowest energy conformation, *e.g.* 1-1. Set the rms reference to this conformation. Now, stepping through the cluster will show you the rms difference between the lowest energy member of this cluster, *i.e.* 1-1, and the rest of the conformations in this cluster.

You can change clusters by picking a different bar in the interactive histogram chart. You can save this histogram as a PostScript file: from the **interactive histogram**'s menu select **Edit** → **Write** to open a file browser for you to enter a filename. Make sure to use ".ps" extension. Select **File** → **Exit** to close.

Note: To facilitate comparing the docked conformations, type **File→Preferences→Set Commands to be Applied on Objects** then select **colorByMolecules** When this is on, each time a new molecule is added to the viewer (up to a current limit of 20), it is colored differently.

The active site of the hiv protease has C2 symmetry. You can see evidence of this by comparing the lowest energy conformations in the different clusters:

1. Click on the lowest_energy bar in **imatinib_1iep_receptor:rms = 2.0 clustering** window to put the ligand into conformation **1_1**. Open **Set Play Options** widget by clicking on the **Ampersand** button. Click on **Build Current**
2. Build a new molecule of the conformations 2_1 by clicking on the second_lowest_energy bar in the clustering widget followed by clicking on **Build Current**

Notice how the docked conformations are symmetry related.

To recluster the results at different rms tolerances, use

Analyze → **Clusterings** → **Recluster…**

You can type in any number float values for **Tolerances** followed by clicking on the **OK** to recluster at these rms values.

4. Evaluate the **chemical reasonableness** of the best results by visually examining the interactions between the receptor and the best docked conformation(s).

First, display the ligand in its lowest energy conformation using the Conformation Player.

Next read in the receptor using:

Analyze → **Macromolecule** → **Open…**

Finally, display a close-up of the receptor atoms interacting with the ligand via:

Analyze → **Dockings** → **Show Interactions**

This radically changes the display, replacing the background color with white, displaying the ligand molecular surface, spheres around atoms in the receptor which are hydrogen-bonded or in close-contact to atoms in the ligand plus displaying secondary structure for sequences of 3 or more residues in the receptor which are interacting with the ligand. The GUI for this command lets you turn on and off different parts of this specialized display.

Examine the interactions between the ligand and nearby atoms in the receptor:

- Is the ligand bound inside a **pocket** in the receptor?
- Are **non-polar** atoms in the ligand docked near **non-polar** atoms in the receptor? Are **polar** atoms in the ligand docked near **polar** atoms in the receptor?
- If you know that a particular residue or residues in the protein interact with the ligand, is that interaction shown in the docked result?
- Do the interactions seem reasonable in the context of what you know about your ligand-receptor from other experimental results such as mutation studies?

Note: The **interpretation** of AutoDock results is open-ended. In large part it depends on **your** chemical insight and creativity. Docked poses of the ligand may suggest chemical modifications such as side-group substitutions, etc….

Now that we have shown you the results of docking, we're going to go back to the very start to show you how to set up molecules for docking and how to prepare the input files for AutoGrid and AutoDock.

Quit ADT by choosing **File**→ **Exit** and then click the **OK** button.

VINA Procedure:

Vina writes results as PDBQT files therefore they can be opened as simple structure molecules. Their energy is reported in the header of the files that can be inspected with a text editor.

1. **File** → **Read Molecule**

First, you need to choose the AutoDock log file you would like to

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

In the exercises which follow, we show you how to set up a docking experiment. Each docking requires at least four input files:

- a PDBQT file for the ligand that encodes a torsion tree;
- a PDBQT file for the receptor;
- a grid parameter file (GPF) for the AutoGrid calculation; and
- a docking parameter file (DPF) for the AutoDock calculation.

If some residues in the receptor are to be modeled as flexible, a fifth file will be necessary:

- a PDBQT file containing the flexible receptor residues.

Protein Data Bank (PDB) files may have a variety of problems that need to be corrected before they can be used in AutoDock. 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 waters from the receptor molecule. Nevertheless, any other problems in the receptor and all problems in your **ligand** must be corrected also. Remember that all hydrogens must always be added prior to using ADT specific commands to format the receptor *and* the ligand.

In this exercise, you will work on the macromolecule, the molecule we want to dock to and which will be kept fixed during the dockings. You will learn how to remove waters, how to add the hydrogens that AutoDock expects, and how to save the modified result.

Restart 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. If you are still in the **Results** directory, navigate up one level by clicking on the button on the top-right of the **Read Molecule** widget. Select **1iep_receptor.pdb** and click on **Open**.

This loads a **Molecule** named '1iep_receptor' into ADT. 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 in '1iep_receptor' are the oxygen atoms of waters that were present in the crystal structure. We will remove these waters later.

2. Color 1iep_receptor 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. **Select** → **Select From String**

Select From String lets you build a selection based on strings you enter for the Molecule, Chain, Residue and/or Atom level. These strings can be names, numbers, ranges of numbers, or lambda expressions that are evaluated to build a set. The strings can contain regular expressions including wild cards such as * which matches anything. For our example, we want

to select all atoms (*) in residues named HOH*. Type **HOH*** in the Residue entry, press the <Tab> key to move to the next entry field, and type ***** in the Atom entry. Now click **Add**. You may get a warning asking you if you want to "change selection level to Atom": if so, click **Yes**.

You will see **Selected: 96 Residue(s)** with a green background in the center of the message-bar at the bottom of the ADT window. Click **Dismiss** to close the **Select From String** widget.

Note: if there is no current selection, ADT expands the selection to include all atoms in the viewer. If the userpref, '**warnOnEmptySelection**' is set to 1, ADT will ask you if it should "expand empty selection to all molecules." The default behavior is to not ask you if you want the empty selection to be expanded to include every molecule in the viewer. For ADT, make sure to leave the **warnOnEmptySelection** set to 0.

4. **Edit** → **Delete** → **Delete AtomSet**

If there is a current selection, it is deleted by this command. You will be asked if you really want to do this, because deleting an AtomSet (or a molecule) cannot be undone. Click on **CONTINUE**. The selected oxygens will disappear from the viewer.

5. **Edit** → **Hydrogens** → **Add**

Choose to add **All Hydrogens** using Method **noBondOrder** with **yes** to renumbering. Click **OK** to add all hydrogens.

6. Hide the protein before going on by clicking on the gray **showMolecules** rectangle for **1iep_receptor** in the Dashboard.

Step 7 is optional because we are going to add charges and AutoDock types to 1iep_receptor in Exercise Seven. However, if you had no further plans to modify this molecule, you would save your modified result at this point:

Exercise Two: Preparing a Ligand for AutoDock.

Note: it is assumed that all oxygen atoms can accept hydrogen bonds and that all hydrogens are donors. The corresponding default atom types are **OA** and **HD**. Nevertheless, parameters for non-acceptor oxygens **O** and non-donor hydrogens **H** are part of default library and might be used in some highly specialized application.

AutoDock requires that ligands have partial atomic charges and AutoDock atom types for each atom; it also requires a description of the rotatable bonds in the ligand. AutoDock uses the idea of a tree in which the rigid core of the molecule is a 'root', and the flexible parts are 'branches' that emanate from the root. Ligands are written in PDBQT-formatted files with special keywords recognized by AutoDock. The keywords ROOT, ENDROOT, BRANCH, and ENDBRANCH establish this torsion tree. The keyword TORSDOF specifies the number of torsional degrees of freedom in the ligand. In the AutoDock 4 force field, the TORSDOF value for a ligand is the total number of rotatable bonds in the ligand. This number excludes bonds in rings, bonds to leaf atoms, amide bonds, guanidinium bonds, *etc*. TORSDOF is used in calculating the change in free energy caused by the loss of torsional degrees of freedom upon binding.

For more information about the PDBQT format, see http://autodock.scripps.edu/faqs-help/faq/what-is-the-format-of-a-pdbqt-file. The atom types in the PDBQT file determine atom parameters like the van der Waals radius and the energy well depth; these are described here http://autodock.scripps.edu/faqs-help/faq/where-do-i-set-the-autodock-4-force-field-parameters and here http://autodock.scripps.edu/faqs-help/faq/how-do-i-add-new-atom-types-to-autodock-4. These parameters are used in the molecular mechanics terms of the AutoDock scoring function: to see the form of these equations, see: http://autodock.scripps.edu/science/equations/.

Procedure:

1. **Ligand** → **Input** → **Open...**

In the "Ligand File for AutoDock 4:" widget, click on the small bar at the right of **PDBQT files: (*.pdbqt)** to display a list of file type choices. Click on **MOL2 files:** to show all the files in this directory and choose **imatinib.mol2**. Click on **Open**.

After the ligand is loaded in the viewer, ADT initializes it. This process involves a number of steps:

- ADT detects whether the ligand already has charges or not. If not, ADT computes **Gasteiger** charges; remember that for the Gasteiger calculation to work correctly, the ligand *must* have all hydrogen atoms added, including both polar and non-polar ones. If the charges are all zero, ADT will try to add charges. It checks whether the total charge per residue is an integer.

- ADT checks for and merges non-polar hydrogens, unless you have set a user preference '**adt_automergeNPHS**' not to do so.

Note: it is also possible to edit which planar, cyclic carbons are assigned AutoDock_type 'A' but we are not going to do that in this example. Also you can adjust the aromaticity cut-off if a ring is more warped via

Ligand → **Aromatic Carbons**□→□

Aromaticity Criterion.

- ADT assigns an 'AutoDock type' to each atom. For peptide ligands, ADT uses a look-up dictionary for planar cyclic carbons (unless you set another user preference, '**autotors_use ProteinAromaticList**', not to do so). For other ligands, ADT determines which are planar cyclic carbons by calculating the angle between adjacent carbons in the ring. If the angle is less than the cut-off of 7.5° (the default value) for all the atoms in the ring, the ring carbons' atom names will be assigned AutoDock type "A". Nitrogens that can accept hydrogen bonds are assigned the AutoDock atom type '**NA',** while those that cannot are assigned '**N**'. In indinavir, the AutoDock type of atom N5 in the heterocycle is 'NA' while the other nitrogens are assigned 'N'. All hydrogens can donate a pair of electrons to a hydrogen bonds and are assigned AutoDock type '**HD'**. Oxygens can accept hydrogen-bonds and are assigned AutoDock type '**OA'**. Likewise, sulphur atoms are assigned AutoDock type '**SA'**.

Note: the PDBQT format [see below] records the AutoDock type in the new '**T**' field. Consequently, names of aromatic carbons do not need to start with 'A' as they must for AutoDock 3.

- ADT displays a **summary** for the formatted ligand that lists what kind of charges were added, how many non-polar hydrogens, aromatic carbons and rotatable bonds were found, the number of torsional degrees of freedom detected (TORSDOF) and the amount the total charge differed from an integral value (total charge error). Click on **OK** to close the summary.

2. **Ligand** → **Torsion Tree** → **Detect Root…**

ADT determines which atom fits its idea of the best root and marks it with a green sphere.

This best root is the atom in the ligand with the smallest largest sub-tree. In the case of a tie, if either atom is in a cycle, it is picked to be root. If neither atom is in a cycle, the first found is picked. (If both are in a cycle, the first found is picked). As you might imagine, this can be a slow process for large ligands.

The rigid portion of the molecule includes this root atom *and* all atoms connected to it by non-rotatable bonds (which we will examine in the next section.) You can visualize the current root portion with **Ligand** → **Torsion Tree** → **Show Root Expansion** (and hide this with **Ligand** → **Torsion Tree** → **Show/Hide Root Marker**).

3. **Ligand** → **Torsion Tree** → **Choose Torsions…**

Opens the Torsion Count widget. The widget displays the number of currently active bonds. Bonds that cannot be rotated are colored red. Bonds that could be rotated but are currently marked as inactive are colored purple. Bonds that are currently active are colored green.

Bonds in cycles cannot be rotated. Bonds to leaf atoms cannot be meaningfully rotated. Only single bonds can be rotated (not double or aromatic etc). ADT determines which bonds could be rotated ('possibleTors'). You set which of these are to be rotatable ('activeTors') by inactivating the others.

You can toggle the activity of a bond or group of bonds by picking them in the viewer if they are displayed as **lines** only. Alternatively, buttons on this widget let you toggle the activity of a type of bonds such as 'peptide bonds', 'amide bonds', 'bonds between selected atoms' or 'all rotatable bonds'. One way to do this is to click on **Make all active bonds non-rotatable** then click on **Make all rotatable bonds rotatable**.

Amide bonds should not be rotatable. By default, amide bonds and aromatic amines bonds are treated as non-rotatable. You can see that two bonds have been inactivated, the bond between atoms **N21** and **C22** and that between atoms **C9** and **N13**. Notice that the current total number of rotatable bonds is **6**. Reactivate the C9-N13 bond and click on **Done**. The maximum number of rotatable bonds allowed by AutoDock is 32, as shown on the widget: "7/32 indicates that 7 are currently active out of the maximum.

4. **Ligand** → **Output** → **Save as PDBQT…**

Opens a file browser allowing you to enter a name. Type in '**imatinib.pdbqt**' and click **Save**.

5. Hide the root marker and the ligand before going on:

Ligand → **Torsion Tree** → **Show/Hide Root Marker**

To hide indinavir, click on the gray **showMolecules** rectangle for **ind** in the Dashboard.

Exercise Three: Preparing the Macromolecule.

Undisplay the ligand by clicking on the gray rectangle at the left side of the **imatinib** row of the Dashboard and re-display the protein **1iep_receptor**.

Procedure:

1. **Grid** → **Macromolecule** → **Choose…**

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

Choose **1iep_receptor** and then click **Select Molecule**.

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.

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**.

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

Click **OK** on the WARNING dialog box. In the Modified AutoDock4 Macromolecule File: file browser, save the macromolecule in the tutorial directory by accepting the default filename **1iep_receptor.dbqt** and clicking on **Save**.

Exercise Four: Preparing the Grid Parameter File.

Note: to specify a custom parameter library in a grid parameter file, use the keyword **parameter_file**. See http://autodock.scripps.edu/faqs-help/faq/where-do-i-set-the-autodock-4-force-field-parameters for more information.

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. In general, one map is calculated for each atom type in the ligand plus an electrostatics map and a separate desolvation map.

AUTODOCK Procedure:

1. **Grid** → **Set Map Types**

The types of maps depend on the types of atoms in the ligand. Thus one way to specify the types of maps is by choosing a ligand. If the ligand you formatted in Exercise Six is still in the viewer, choose **Grid** → **Set Map Types** → **Choose Ligand…**, select '**imatinib**' and click on the **Select Ligand** button.

If not, use **Grid** → **Set Map Types** → **Open Ligand…**.

Note: Alternatively, if you plan to use the same macromolecule with a variety of different ligands, you might choose to calculate all the maps you would eventually need via **Set Map Types**-> **Directly**.

Choosing the ligand opens the **AutoGpf Ligand** widget that allows you to modify the types of maps to be calculated, and to choose whether to model possible hydrogen bonding.

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:

Adjust the number of points in to **45**, **65**, **55**. Notice that each map will have 170,016 points.

Important Note: ⟶
z value is **MINUS** 7.5

Type in **15.7**, **53.2** and **17.5** in the *x* center, *y* center and *z* center entries. This will center the grid box on kinase the active site where the ATP binds.

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 '**1iep_receptor.gpf'**

Important Note:
If you are setting up a docking using flexible residues, make sure the specified **receptor** file is 1iep_receptor_rigid.pdbqt**.** **Grids must be calculated using a file for the molecule without the moving residues.**

[4. **Grid** → **Edit GPF…**

If you have written a grid parameter file, it opens in an editing window. If not, you can pick one to read in and edit via the **Read** button. If you make any changes to the content of the grid parameter file, you can save the changes via the **Write** button. Edit GPF will open the file we wrote in step 3. Either **OK** or **Cancel** close this widget.]

VINA Procedure:

Vina grid definition requires the center coordinates x,y,z and the sizes in Angstrom. Those values can be obtained from the Grid widget used for the AutoGrid setup (see step 2) by changing the grid resolution to 1.00 Angstroms. Try to define a grid box the most similar to the one used by AutoDock. For this purpose, use the same center as in the step 2, and visually inspect that the values 17, 24, 21 are satisfactory.

Exercise Five: Starting AutoGrid 4

AutoGrid 4 (and **AutoDock 4)** must be run in the directory where the rigid macromolecule, ligand and parameter files are to be found.

For this reason, the current PMV session must be set to use the directory containing the input files:

- **File->Preferences->Modify Defaults->General->Startup Directory**

- Cut and paste the full path of the directory containing the inputs, that should be like:

C:\Users\%USERNAME%\Desktop\MGL\AutoDock\Input

Procedure:

1. **Run** → **Run AutoGrid…**

   Use the "Browse" button to set the full path to the Parameter Filename (**1iep_receptor.gpf**)

Here is a brief tour of this widget:

- The first two entries in the widget are used to specify which machine to use. By default the local machine is named in the Macro Name: entry and in the Host name: entry. It is possible to define macros to specify other machines with **Run** → **Host Preferences…**

- **Program Pathname**: entry specifies the location of the autogrid4 executable. If it is not in your path, you can use Browse to locate it.

- **Parameter Filename**: entry specifies the GPF file. If you have just written a GPF file, opening this widget will automatically load the GPF filename in

the **Parameter Filename:** entry. If not, you can use **Browse** to locate the GPF you want to use.

- **Log Filename**: entry specifies the log file. Selecting a GPF creates a possible related name for the GLG.

- **Nice Level**: entry used to specify a nice level for remote jobs.

- **Cmd**: entry shows you the command that will be invoked when you click on **Launch**.

2. **Launch**

Starts the AutoGrid 4 job. This opens a Process Manager that allows you to see specifics about current AutoGrid and AutoDock jobs.

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

```
% autogrid4 -p 1iep_receptor.gpf -l 1iep_receptor.glg &
```

Exercise Six: Preparing the Docking Parameter File.

The docking parameter file tells AutoDock which map files to use, the ligand molecule to move, what its center and number of torsions are, where to start the ligand, the flexible residues to move if sidechain motion in the receptor is to be modeled, which docking algorithm to use and how many runs to do. It usually has the file extension, "**.dpf**". Four different docking algorithms are currently available in AutoDock: SA, the original *Monte Carlo* simulated annealing; GA, a traditional Darwinian genetic algorithm; LS, local search; and GALS, which is a hybrid genetic algorithm with local search. The GALS is also known as a **Larmarckian** genetic algorithm, or LGA, because children are allowed to inherit the local search adaptations of their parents.

Each search method has its own set of parameters, and these must be set before running the docking experiment itself. These parameters include what kind of random number generator to use, step sizes, *etc.* The most important parameters affect how long each docking will run. In simulated annealing, the number of temperature cycles, the number of accepted moves and the number of rejected moves determine how long a docking will take. In the GA and GALS, the number of energy evaluations and the number of generations affect how long a docking will run. ADT lets you change all of these parameters, and others not mentioned here. See Appendix 4 Docking Parameters for more information about individual keywords.

Procedure:

Note: If you are setting up a docking with flexible residues , here you must select the name of the file that contains the rigid receptor, *e.g.* hsg1_rigid.pdbqt

1. **Docking** → **Macromolecule** → **Set Rigid Filename…**

   Select **1iep_receptor.pdbqt** in the file browser that opens.

Note: You can only choose a ligand if you have previously written it to an output file because AutoDock requires the filename of the formatted ligand.

2. **Docking** → **Ligand** → **Choose…**

   Choose **imatinib**. Click **Select Ligand**.

Note: If you are modeling movement in the sidechains of some residues in the receptor, you must ALSO specify the name of the file containing the formatted moving residues in the docking parameter file.

Docking → Macromolecule → Set Flexible Residues Filename...

This opens a panel that tells you the name of the current ligand, its atom types, its center, its number of active torsions and its number of torsional degrees of freedom. You can set a specific initial position of the ligand and initial relative dihedral offsets and values for its active torsions. For our exercise we will use the defaults. Click **Close** to close this widget.

3. **Docking → Search Parameters... → Genetic Algorithm...**

Note: You can change the number of **runs** here, too.

This lets you change the genetic algorithm specific parameters. It is a good idea to do a trial run with fewer energy evaluations. For our exercise, we will use the **short** setting, *i.e.* 250,000 energy evaluations. This is listed as "Maximum Number of evals". Click **Accept** to continue.

4. **Docking → Docking Parameters...**

Here you can choose which random number generator to use, the random number generator seeds, the energy outside the grid, the maximum allowable initial energy, the maximum number of retries, the step size parameters, output format specification and whether or not to do a cluster analysis of the results. For today, use the defaults and just click **Close**.

Note: ADT allows you to change the parameters for any of the four possible docking algorithms at any time. You commit to a specific algorithm only when you write the docking parameter file.

5. **Docking → Output → Lamarckian GA...**

We specify the name of the DPF we are about to write out here. This file will contain docking parameters and instructions for a), Lamarckian Genetic Algorithm (LGA) docking also known as a Genetic Algorithm-Local Search (GA-LS).
Type in **imatinib_1iep_receptor.dpf** and click on **Save**.

[6. **Docking → Edit DPF...**

Note: If you are modelling flexibility in the sidechains of some residues in the receptor, the **flexres** keyword and a filename must appear in the docking parameter file.

Use this menu option to look at the contents of the file from step 5. Check that the output filename, "**imatinib.pdbqt**" appears after the keyword 'move' and that torsdof is set to "14". You can click either **OK** or Cancel to continue.]

Exercise Seven:Starting AutoDock4/AutoDockVina

In general you should know:

- **AutoDock and Vina** must be run in the directory where the macromolecule, ligand, GPF and DPF files or config files are to be found.

- The named files in the parameter file must not include pathnames.

AUTODOCK4 Procedure:

1. **Run** → **Run AutoDock...**

Here is a brief tour of this widget:

- The first two entries in the widget are used to specify which machine to use. By default the local machine is named in the **Macro Name**: entry and in the **Host Name**: entry. It is possible to define macros to specify other machines.

- **Program Pathname**: entry specifies the location of the **autodock4** executable. If it is not in your path, you can use **Browse** to locate it.

- **Parameter Filename**: entry specifies the DPF file. If you have just written a DPF file, opening this widget will automatically load the DPF filename in the **Parameter Filename:** entry. If not, you can use **Browse** to locate the DPF you want to use.

- **Log Filename**: entry specifies the log file. Selecting a DPF creates a possible related name for the DLG.

- **Nice Level**: entry used to specify a nice level for remote jobs.

- **Cmd**: entry shows you the command that will be invoked when you click on **Launch**.

2. **Launch**

Starts the AutoDock job. This opens a Process Manager Widget that allows you to see specifics about current AutoDock job. It is a limited process manager that you can use to terminate an AutoDock process. You are asked if you really want to kill it.

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

```
% autodock4 -p imatinib_1iep_receptor.dpf
-l imatinib_1iep_receptor.dlg
```

VINA Procedure:

1. Open a Windows terminal: **Start->Run** → **type "cmd.exe"**

cd to the Desktop directory containing the input files (HINT: use the Tab key to complete the path):

cd C:\Documents and Settings\eng\Desktop\vina\input

Run Vina executable with the following syntax and adding all the parameters in the same line:

C:\Program Files\The Scripps Research Institute\Vina\vina.exe

--center_x 15.6 --center_y 53.2 --center_z 17.5

--size_x 17 --size_y 24 --size_z 21

--ligand imatinib.pdbqt --receptor 1iep_receptor.pdbqt

Exercise Eight: Preparing the flexible residue file (optional).

If you are not modeling flexible sidechains in some residues in the receptor, skip this exercise and proceed to Exercise 10.

AutoDock 3 docks a flexible ligand to a rigid receptor. AutoDock 4 adds support for including a conformational search of flexible sidechains of designated residues in the receptor.

For an experiment which includes some flexible residues in the receptor, the AutoGrid calculation must be based only on the rigid residues. Our convention is to use '1iep_flex.pdbqt' for the flexible residue file and 1iep_rigid.pdbqt for the rigid residue file. The map filenames are then '1iep_rigid.A.map', '1iep_rigid.C.map, '1iep_rigid.N.map' etc.

The flexibility pattern in the moving sidechains of the residues is encoded with AutoDock 4 specific keywords **BEGIN_RES** and **END_RES** as well as ROOT, ENDROOT, BRANCH and ENDBRANCH. The flexible residues are written in a separate **PDBQT** file. The keyword '**flexres'** followed by a filename in a docking parameter file sets up a flexible receptor AutoDock 4 experiment.

Redisplay **1iep_receptor.pdbqt** using its **showMolecules** rectangle in the Dashboard.

Procedure:

Note: If the hsg1 is not in the viewer, use

FlexibleResidues → Input → Open Macromolecule...

1. Choose **1iep_receptor.pdbqt** as the macromolecule to have flexible residues:

Flexible Residues → Input → Choose Macromolecule...

Click on **1iep_receptor.pdbqt** in the widget that opens and on **Select Molecule**.

Click **Yes** when asked if you want to merge the non-polar hydrogens. Click on **OK** in the formatting summary widget.

2. Select the residues to be flexible.

Select → **Select From String**

Click on **Clear Form** to empty the entries.

Note: Make sure you have selected the residues **BEFORE** going on to step 3. Otherwise, ADT will attempt to detect and display rotatable bonds in ALL THE RESIDUES in hsg1. This process would be extremely time consuming…..

Type **ARG8** in the **Residue** entry and click on **Add**. Click **Dismiss** to close the **Select From String** widget.

Check that **2 Residue(s)** appears in the **Selected:** entry below the 3D Viewer.

3. Define the rotatable bonds in the selected residues.

Flexible Residues → **Choose Torsions in Currently Selected Residues…**

This hides all the non-selected residues in the macromolecule. The sidechains of the selected residues are shown with currently rotatable bonds colored green, unrotatable bonds colored red and non-rotatable bonds colored magenta. The total number of rotatable bonds is listed in the **Torsion Count** widget. Clicking on a rotatable bond makes it non-rotatable. Clicking on a non-rotatable bond makes it rotatable.

Note: Here we inactivate this bond only to demonstrate how to do so. You can choose which of the possible torsions in the flexible sidechains of residues in the receptor to model as active. You can choose to keep them all active. The limit on the number of torsions including those in the ligand is 32 in AutoDock.

Inactivating this bond is **not** required.

Click on the rotatable bond between **CA** and **CB** in each residue to inactivate it. This leaves a total of **6** rotatable bonds in the two flexible ARG8 residues.

Click on **Close**.

Clear the selection by clicking on the **pencil eraser** icon.

You must save the macromolecule in **two** files: one containing the formatted, flexible ARG8 residues and the other all the rest of the residues in the macromolecule.

4. Save the **flexible** residues:

Flexible Residues → **Output** → **Save Flexible PDBQT…**

Type **1iep_flex.pdbqt** in the **AutoFlex File**: browser and click **Save** (or press "Return" key on a Mac or "Enter" key on a PC).

5. Save the non-moving, **rigid** residues:

Flexible Residues → **Output** → **Save Rigid PDBQT…**

Type **1iep_receptor_rigid.pdbqt** in the **AutoFlex Non-Flexible Residue Output File:** browser and click **Save**

Exercise Nine: More Visualizations….

Ultimately, the goal of a docking experiment is to **illustrate** the docked result in the context of the macromolecule, explaining the docking in terms of the overall energy landscape. The interactions between the ligand and the macromolecule are driven by energy composed of van der Waals (vdW), electrostatic, hydrogen bonding and desolvation component energies.

This exercise has three parts:

First to evaluate the chemical reasonableness of our results, we will represent the macromolecule as a solvent-excluded surface using **msms** and check whether the ligand has docked in a 'pocket' on the receptor and whether the pairwise-interactions between atoms in the ligand and atoms in the receptor are reasonable.

Next we will explore the energy landscape of the binding site, representing it using **isocontours**. This view of the docking can elucidate the binding mechanism and suggest chemical modifications of the ligand.

Finally, we will introduce two other ways of visualizing all the docked structures at once, the 'overall' binding pattern.

Note: If your docking included flexible residues, delete **hsg1** from the Viewer, read in the rigid molecule **hsg1_rigid.pdbqt** and use it instead of hsg1 in this Exercise.

If 1iep_receptor is still in your viewer, skip Step 1. Instead use **Display** → **Show/Hide Molecule** to display it if it is not visible. Also, undisplay any docked conformations you may have already built.

Procedure:

1. **Analyze** → **Macromolecule** → **Open…**

   If **1iep_receptor.pdbqt** cannot found in the current directory, a file browser opens to ask you to specify where it can be found.

Visualize binding in a pocket by displaying molecular surfaces:

TSRI tutorials only: Improve the msms surface display by turning off **depth-cueing**. To do this, click on the **D** on the keyboard.

In the Dashboard, click on the circle under MS in the **imatinib** row and in the **1iep_receptor** row. Click on the diamond under Atom to color the ligand surface by the element of the underlying atom. Click on the diamond under DS to color 1iep_receptor by David Goodsell colors.

This view of the docking allows you to see how the docked ligand 'fits' into the macromolecule. Clicking on the circles under MS allows you to show and hide the various msms surfaces.

Alternatively, you may want to visualize the docked conformations in the context of the energy grids. This may be useful for computer-aided drug design.

In the steps that follow, you will visualize the oxygen affinity map as an isocontour, then display only two residues in the active site of the ATP and finally see atom OA of imatinib sitting in a pocket of oxygen next to the ASP381 residue backbone. This is our most complicated exercise.

2. **Analyze** → **Grids** → **Open...**

This opens a list chooser of the grids used in this docking. Select **iep_receptor.OA.map**.

Note: The grid isocontours are colored by atom type. (See Exercise One for a list of these colors.)

The AutoGrid map file is read into the viewer, creating an instance of a **Grid**. This map is visualized as an **isocontour** in 3D. This means that every point in the grid box that is equal to the isocontour level will be connected together by lines or polygons. You can change the isocontour level, which is an energy in Kcal/mol; the step between grid points for sampling the grid values; and whether to show the isocontoured regions as lines or filled (solid) polygons. You can also toggle the visibility of the Grid and its bounding box.

To illustrate the kind of information you can obtain from the atomic affinity grid maps, try this:

1. Set the IsoValue to **-0.5**; if you type into the slider entry, remember to press <**Return**> or <**Enter**>.

2. Display **1iep_receptor.pdbqt**; if it is not present in the viewer, use **Analyze** → **Macromolecule** → **Open...**.

3. Choose **Select** → **Select From String** and type in **ASP381** into the Residue field and then click **Add**. Click **Yes** to change selection level if necessary and **Dismiss** to close **Select From String** widget.

4. Choose **Display** → **Sticks And Balls**

   Opens **Display Sticks and Balls:** widget. Increase the quality to 15 and click ok.

   Choose **Color** → **By Atom Type** and select 'balls' and 'sticks' in the widget that opens and click ok.

5. Choose the lowest-energy cluster using the **Analyze->Clustering->Show** menu entries and clicking on the first bar from the left.

Now you can rotate the objects in the viewer. You will see that the oxygen atom of the inhibitor is buried in cloud of Oxygen-affinity.

Click **Display Map** and **Show Box** to undisplay the isocontour and its bounding box before you **Dismiss** this panel.

3. **Analyze** → **Dockings** → **Show as Spheres…**

This command provides a '*bird's eye view*' of all the docking results. It represents each docked conformation by a sphere placed at the average position of the coordinates of all the atoms in that conformation. Clicking on the name of a docking log in the list makes the spheres representing its results visible (if the associated ligand is visible).

Click on **ind.dlg** in the list. You can change the **radii** of the spheres, their **color** and their smoothness (or "**quality**"). [You may need to reduce the radii to .03 Å to see the overlapping dockings of our result.]

——

Files for Exercises:

Input Files:

1iep_receptor.pdb
imatinib.pdb

Results Files

Ligand

imatinib.pdbqt (7 torsions)

Macromolecule

1iep_receptor.pdbqt

AutoGrid

1iep_receptor.gpf
1iep_receptor.glg
1iep_receptor.*.map
1iep_receptor.maps.fld,1iep_receptor.maps.xyz

AutoDock

imatinib_1iep_receptor.dpf
imatinib_1iep_receptor.dlg

VINA

1iep_receptor.pdbqt

imatinib.pdbqt
imatinib_1iep_receptor.conf

imatinib_out*.pdbqt

Useful Scripts in AutoDockTools/Utilities24

prepare_ligand4.py

prepare_receptor4.py

prepare_flexreceptor4.py

prepare_gpf4.py

prepare_dpf4.py

summarize_results4.py

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/ri (X) and up/down*
Shift	*Select*	*Scale or Zoom*	*Translate in/out*

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 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*

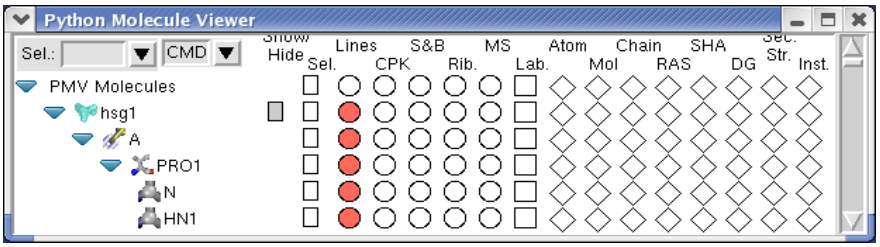
Note: By default, the Viewer's current object is root so you will not see any changes here if you toggle between *transform root* and *transform current object*. The DejaVu GUI lets you change the current object.

	the Viewer's current object.

Appendix 2: Dashboard Widget

Select entry + **Command** Buttons ⟶

Tree Widget



Note: Clicking on a s**h**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.

Appendix 3: Conformation Player

- **Type-in entry** at center for random access to any conformation by its id. Valid ids depend on which menubutton was last used to start the player.

- Click on **black arrow** buttons next to entry to change to next or previous conformation in current list.

- **White arrow** buttons start play according to current play mode parameters (see below). Clicking on an active white arrow button stops play. [While a play button is active, its icon is changed to double vertical bars.]

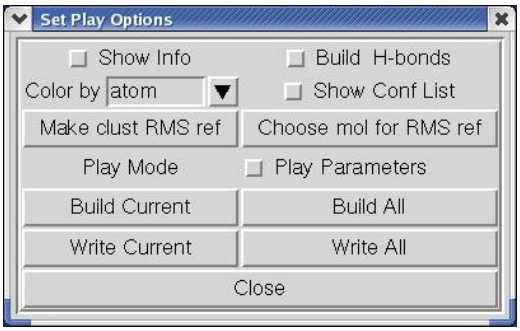
- **Double black arrow** buttons start play as fast as possible in the specified direction.

- **Double black arrow plus line** buttons advance to beginning or end of conformation list.

- **Ampersand** button opens the **Set Play Options** widget (see next).

- **Quatrefoil** button closes the player.

Next, a tour of the **Set Play Options** widget and its buttons:



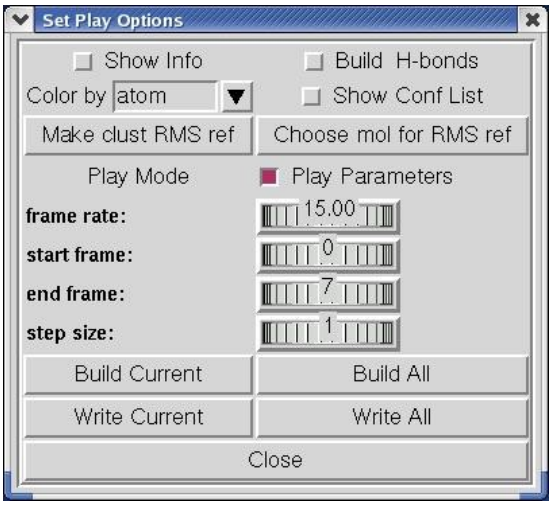
Note: building **hydrogen bonds** requires that the receptor molecule be present in the viewer and that you have either chosen it using:

Analyze→Macromolecule→Choose or read it in specifically using

Analyze→Macromolecule→Open …

- **Show Info** opens and closes a separate panel **Conformation # Info** which displays additional information about current conformation (see following).
- **Build H-bonds** turns on and off building and displaying hydrogen bonds between the macromolecule and the ligand in its current conformation.
- **Color by** allows you to choose how to color the ligand from a list of available coloring schemes.
- **Show Conf List** opens and closes a separate **Choose Conformations** widget showing current idlist (see below).
- **Make clust RMS ref** sets the reference coordinates for RMS to those of the current conformation. [This RMS value is shown in Info panel as clRMS]
- **Choose mol for RMS ref** lets you select a different molecule from list of those in Viewer to use as reference for a new RMS computation.
- **Play Mode** opens a separate **Play Mode** widget (see next).
- **Play Parameters** exposes controls for setting parameters governing how the conformations are played:

Note: To set the value of a thumbwheel click on it with the left mouse button and hold the mouse button down while you drag the mouse to the right to increase the value or to the left to decrease the value. Alternatively, you can right-click on a thumbwheel to open a separate widget which lets you type in a new value.



Note: the input conformation of the ligand is always inserted at the list of conformations . It is always conformation 0.

Four **thumbwheel** widgets are used to set **Play Parameters**.

- **frame rate:** set the relative speed of the player [absolute rate is cpu dependent]
- **start frame:** index into current conformation list. **Note** the input conformation is always inserted at index 0 in sequence list.
- **end frame:** index into current conformation list.
- **step size:** determines next conformation in list. For example, step size 1 plays every available conformation whereas step size 2 every other…

Clicking on **Play Parameters** hides the thumbwheels.

- **Build Current** adds a new molecule to the viewer with current conformation's coordinates providing that a molecule hasn't already been built with this conformation's id.

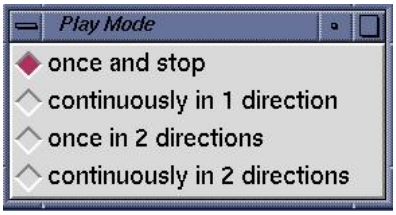
- **Build All** builds a new molecule for each conformation in the current sequence of conformations bound to the player.

- **Write Current** lets you choose a filename for writing a formatted file using the current coordinates.

- **Write All** writes a formatted file for each set of coordinates in the current sequence. This uses default filenames based on the id of each Conformation.

- **Close** button closes **Set Play Options** widget.

Next, the **Play Mode** widget and its buttons:



These 4 radiobuttons are used to set the current play mode. [Note that at any time, the current **endFrame** and the current **startFrame** depend on the direction of play.]

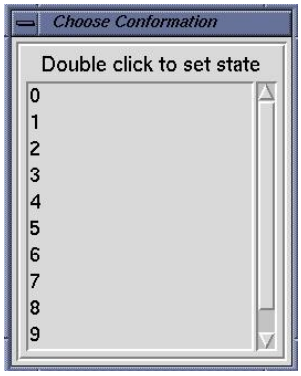
- **once and stop** plays from the current conformation in the current direction up to and including the endFrame.

- **continuously in 1 direction** plays in the current direction up to and including the endFrame and then restarts with startFrame , again and again….

- **once in 2 directions** plays from the current conformation in the current direction up to and including the endFrame and then plays back to startFrame and stops.

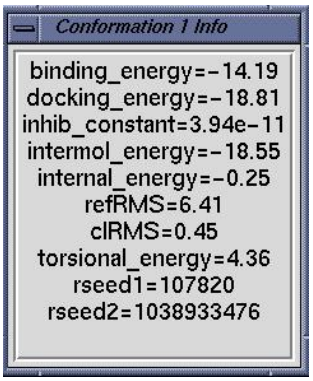
- **continuously in 2 directions** plays from current conformation in current direction up to endFrame then back to the beginning then back to the end, again and again….
-

The Choose Conformation widget:



Choose Conformation widget has list of ids for each conformation in the current sequence list. Double clicking on an entry in this list updates the ligand to the corresponding conformation. This widget is closed by clicking on the checkbutton **Show Conf List** in **Set Play Options** widget.

Conformation # Info widget shows information about a specific conformation from a docking experiment.



- **binding energy** is the sum of the intermolecular energy and the torsional free-energy penalty.

- **docking energy** is the sum of the intermolecular energy and the ligand's internal energy.

- **inhib_constant** is calculated in AutoDock as follows:

  Ki=exp((deltaG*1000.)/(Rcal*TK)

  where **deltaG** is docking energy, **Rcal** is 1.98719 and **TK** is 298.15

- **refRMS** is rms difference between current conformation coordinates and current reference structure. By default the input ligand is used as the reference.

- **clRMS** is rms difference between current conformation and the lowest energy conformation in its cluster.

- **torsional_energy** is the number of active torsions * .3113
  [.3113 is AutoDock 3 forcefield torsional free energy parameter]

- **rseed1** and **rseed2** are the specific random number seeds used for current conformation's docking run.

Appendix 4: Docking Parameters

Parameters common to SA, GA, GALS:

‘**parameter_file**’: *optional* file containing a user-defined library of customized parameters.

'seed': The number of arguments following this keyword determines which random number generator is to be used. One argument causes AutoDock to use the system’s implementation of the random number generator and a corresponding system seed call. One argument is required for the simulated annealing algorithm. Two arguments tells AutoDock to use the platform-independent library for random number generation. Two arguments are required for the genetic algorithm. The arguments themselves can be any combination of explicit long integers, the key word ‘time’ or the keyword ‘pid’. ‘time’ is the number of seconds since the epoch, referenced to 00:00:00CUT 1 Jan 1970. ‘pid’ gives the UNIX process ID of the currently executing AutoDock process.

'ligand_types': all AutoDock 4 atom types present in ligand. If flexible residues are to be docked, their atom types must be included.

'fld': grid data field file created by AutoGrid and readable by AVS.

'map': filename for the first AutoGrid affinity grid map of the 1st atom type. Repeated for all atom types specified in ‘ligand_types’.

'elecmap': filename for the required electrostatics map.

'desolvmap': filename for the required desolvation map.

'flexres': *optional* filename for the flexible residues to be docked.

'unbound <value>': *optional* reference energy ‘value’ for the unbound extended state of the ligand to be docked. If many dockings are to be done with the same input ligand, it is not necessary to compute the unbound extended energy for each. Instead, it could be included in subsequent docking parameter files using this keyword.

‘**unbound_intnbp_coeffs’**: *optional* internal pairwise non-bonded energy parameters for the unbound energy calculation for the flexible ligand: equilibrium distance and well depth followed by integer exponents n and m.

‘**include_1_4_interactions’:** *optional* flag to include the pairwise-energy for atoms separated by three bonds in the internal energy calculation. By default the interaction energy of atoms separated by 3 bonds is not included.

‘**epdb** filename’: *optional* calculate the energy of the molecule in filename in the context of the specified maps. Previously this feature was only accessible via the command-mode of AutoDock 3. The command-mode is no longer supported in AutoDock 4.

'about': x y z center of ligand about which rotations will be made. Coordinate frame of reference inside AutoDock. That is, internally the ligand’s coordinates become centered at the origin. If flexible residues are to be included in the docking, their atom positions are not included in determining the ‘about’ value: it remains the center of the ligand atoms only.

'reorient': apply an initial reorientation to the ligand either **random**, **standard** which moves the ligand such that the first three atoms lie parallel to the xy-plane and the first two atoms lie parallel to the x-axis or **<axis-x> <axis-y><axis-z>** to apply a specific rotation to the input ligand.

'tran0': initial coordinates for the center of the ligand or **random**. Each new run starts the ligand from this location. Please note: the user should enscure that the ligand, when translated to these coordinates still fits inside the volume of the grid maps. If there are some atoms which do lie outside this volume, AutoDock will automatically move the ligand until the ligand is completely inside the box.

'quat0': *deprecated* now use either **quaternion0** or axis**a**ngle0.

‘quaternion0: initial quaternion Qx, Qy, Qz, Qw or **random**.

'axisangle0: initial axis and angle for rotation Qx, Qy, Qz, Qang or **random**.

'ndihe': *deprecated* AutoDock 4 determines the number of rotatable bonds from the PDBQT file of the ligand and flexible residues file (if there is one).

'dihe0': *optional* initial relative dihedral angles or random. There must be a value specified for each rotatable bond in the ligand (and optional flexres file) if the user decides to explicitly set the initial relative dihedrals.

'tstep': if one argument, the maximum translation jump per step. If single argument and less than one, the reduction factor is multiplied with the tstep at the end of each cycle to get next value. Alternatively, if there are two arguments, the user specifies the value for the first and last cycle and AutoDock calculates the reduction factor that satisfies these constraints. Default is 2.0 Angstrom

'qstep': maximum orientation step size for the angular component w of quaternion. Default is 50.0 degrees.

'dstep': maximum dihedral step size. Default is 50.0 degrees.

'torsdof': number of the torsional degrees of freedom for the estimation of the change of free energy upon binding. For AutoDock 4 this is,the number of **possible rotatable bonds in the ligand**. This differs from torsdof for AutoDock 3 which excluded any torsions that only rotated hydrogens: eg hydroxyls, amines.

'intelec': always calculate internal ligand electrostatic energies for AutoDock 4.

'outlev': diagnostic output level. For simulated annealing 0= no output, 1=minimal output, 2 = full state output at end of each cycle, 3=detailed output for each step. For GA and GA-LS: 0=minimal output, 1= write minimum, mean and maximum of each state variable at the end of every generation. Use outlev 1 for SA and outlev 0 for GA and GA-LS.

'rmstol': the rms deviation tolerance for cluster analysis, carried out after multiple docking runs. If two conformations have an rms less than this tolerance, they will be placed in the same cluster. The structures are ranked by energy, as are the clusters.

'rmsatoms': *optional* the base for cluster analysis by default is '**all'** atoms. In a docking involving flexible residues, it is possible to cluster using only the ligand atoms. To do so, include this keyword with the value '**ligand_only**'

'rmsref': *optional* the root mean square deviation of the docked conformations calculated with respect to the coordinates in the

PDBQT file or PDB file specified here. Particularly useful for comparing a docked result to a known crystal structure.

'extnrg': *optional* external grid energy assigned to any atoms that stray outside the volume of the grid during a docking. Default is 1000. kcal/mole.

'**compute_unbound_extended'**: use a specialized force field with no internal electrostatics to drive the ligand into an extended conformation and compute the energy of this extended state. This is used in the AutoDock 4 force field as the unbound energy of the input ligand. If the unbound extended energy of the ligand is specified using the keyword **unbound**, compute_unbound_extended becomes *optional*.

'analysis': perform a cluster analysis on results of a docking and output results to the log file. The docked conformations are sorted in order of increasing energy, then compared by root mean square deviation. If the docked result is within the 'rmstol' threshold, it is placed into the same cluster.

Simulated Annealing Specific Parameters:

'rt0': initial annealing temperature-actually absolute temperature multiplied by the gas constant. Default is 500. cal/mole.

'e0max': two floats-used only in SA. Keyword stipulates that the ligand's initial state cannot have an energy greater than the first value, nor can there be more than the second value's number of retries. Default is 0, 10000.

'linear_schedule': instructs AutoDock to use a linear temperature reduction schedule during Monte Carlo simulated annealing. Unless given, a geometric reduction schedule is used according to rtrf described below. Default is to use linear_schedule.

'rtrf': annealing temperature reduction factor. At the end of each cycle the annealing temperature is multiplied by this factor to give that of the next cycle. Must be positive and less than one. Default is 0.95

'trnrf': per cycle reduction factor for translations. Default is 1.0.

'quarf': per cycle reduction factor for quaternions. Default is 1.0.

'dihrf': per cycle reduction factor for dihedrals. Default is 1.0

'runs': number of automated docking runs to carry out. Default is 10.

'cycles': number of temperature reduction cycles. Default is 50.

'accs': Maximum number of accepted steps per cycle. Default is 100.

'rejs': Maximum number of rejected steps per cycle. Default is 100.

'select': State selection flag. 'm' minimum state is selected or 'l' last state. Default is m.

'simanneal': instructs AutoDock to do specified number of docking runs using the simulated annealing SA search engine.

Genetic Algorithm Specific Parameters:

'ga_pop_size': number of individuals in the population. Each individual is a coupling of a genotype and its associated phenotype. Typical values range from 50 to 200. Default is 50.

'**output_pop_file**' optional filename to write populations.

'ga_num_evals':Maximum number of energy evaluations that a GA run should make. Default is 250000.

'ga_num_generations': Maximum number of generations that a GA or LGA run should last. Default is 27000.

'ga_elitism': Number of top individuals that are guaranteed to survive into the next generation. Default is 1.

'ga_mutation_rate': The probability that a particular gene is mutated. Default is 0.02

'ga_crossover_rate': Crossover rate is the expected number of pairs in the population that will exchange genetic material. Setting this value to 0 turns the GA into the evolutionary programming method (EP) but that requires a concomitant increase in the ga_mutation_rate in order to be effective. Default is 0.80.

'ga_window_size': Number of preceding generations to take into consideration when deciding the threshold for the worst individual in the current population. Default is 10.

'ga_cauchy_alpha': Alpha parameter in a Cauchy distribution which corresponds roughly to the mean of the distribution. Default is 0.

'ga_cauchy_beta': Beta parameter in a Cauchy distribution which corresponds roughly to something like the variance of the distribution. However, the Cauchy distribution doesn't have finite variance. Default is 1.

'set_ga': sets the global optimizer to be a genetic algorithm. PLEASE note all ga_ parameters must be specified before this line in order to be used in the docking.

'do_global_only': instructs AutoDock to carry out docking using only a global search. Local search parameters in the DPF are ignored when this is present.

Local Search Specific Parameters:

'sw_max_its': Maximum number of iterations that the local search procedure applies to the phenotype of any given individual. Default is 50.

'sw_max_succ': Number of successes in a row before a change is made to the rho parameter in Solis & Wets algorithm. Default is 4.

'sw_max_fail': Number of failures in a row before Solis & Wets algorithms adjust rho. Default is 4.

'sw_rho': Parameter defining the initial variance and specifying the size of the local space to sample. Default is 1.0.

'sw_lb_rho': Lower bound on rho, the variance for making changes to genes. Rho can never be modified to a value smaller than sw_lb_rho. Default is 0.01.

'ls_search_freq': The probability of any particular phenotype being subjected to local search. Default is 0.06.

'set_psw1': Instructs AutoDock to use the pseudo-Solis and Wets local searcher. This method maintains the relative proportions of variances for the translations in Angstrom and the rotations in radians. These are typically 0.2 Angstrom and 0.087 radians to start with so the variance for translations will always be 2.3 times larger than that for the rotations (i.e. the orientation and torsions).

'do_local_only': Instructs AutoDock to carry out only the local search of a global-local search. The genetic algorithm parameters are ignored, except for the population size. This is an ideal way of carrying out a minimization using the same force field as is used during the dockings. The ga_run keyword should not be given. The integer following the keyword determines how many dockings will be performed. Default is 50.

GALS parameters include all Genetic Algorithm and Local Search parameters listed above except **do_global_only** and **do_local_only**.

Clustering Keywords:

'cluster': keyword specific to reclustering jobs. Parameter which follows is filename for concatenated docking logs. Script 'recluster.py' produces this kind of file.

'rmstol': root mean square tolerance for reclustering.

'ligand_types': types of atoms present in ligand

'write_all': option causes printed output of all docked structures in each cluster. Default is to write the first docked structure in each cluster only. (The first docked structure has the lowest energy of all the docked structures in the cluster.)