Autodock Vina on Linux Cluster with HTCondor

Jean-Yves Sgro

April 18, 2017

Table of Contents

## Warning: package 'knitr' was built under R version 3.5.2

# Learning Objectives

* Download and install autodock and/or autodock vina binaries
* Run prepared files on the Linux cluster with HTCondor commands

The purpose of this session is to learn how to run the Autodock and the Autodock Vina software directly on the Biochemistry Computational Cluster (BCC). File preparation will be secondary.

For remote connection you can use a Macintosh Terminal.

On Windows you could use PuTTy or MobaXterm.

*Note*: BCC does not support X11 and therefore the cluster is completely text-driven.

# Docking

Autodock and the alternate version Autodock Vina are popular but the article “Beware of Docking!” (Chen 2015) provides an almost exhaustive list of current docking software in addition to presenting caveats of the process of docking.

# Introduction

**What is the difference between AutoDock Vina and AutoDock 4?**

(*Based on the Autodock Vina* [*FAQ*](http://vina.scripps.edu/manual.html))

*AutoDock 4 (and previous versions) (Morris et al. 2009) and AutoDock Vina (Trott and Olson 2010) were both developed in the* [*Molecular Graphics Lab*](http://mgl.scripps.edu/) *at The Scripps Research Institute.*

*AutoDock Vina inherits some of the ideas and approaches of AutoDock 4, such as treating docking as a stochastic global opimization of the scoring function, precalculating grid maps (Vina does that internally), and some other implementation tricks, such as precalculating the interaction between every atom type pair at every distance. It also uses the same type of structure format (PDBQT) for maximum compatibility with auxiliary software.*

*However, the source code, the scoring funcion and the actual algorithms used are brand new, so it’s more correct to think of AutoDock Vina as a new “generation” rather than “version” of AutoDock. The performance was compared in the original publication, and on average, AutoDock Vina did considerably better, both in speed and accuracy.*

*However, for any given target*, **either program may provide a better result, even though AutoDock Vina is more likely to do so**. *This is due to the fact that the scoring functions are different, and both are inexact.*

# Process:

We will do the following:

1. login to the linux Biochemistry Computational Cluster (BCC)
2. Organize folders in the /scratch directory
3. Download binaries with wget
4. Unarchive and install binaries

# Login to BCC

**TASK**

This button will invite you to act on **Open a Terminal and login**.

1. Open a Macintosh Terminal
2. connect to BCC with your ***UWNeID*** credentials: 2.1 Replace myname with your actual NetID

ssh myname@submit.biochem.wisc.edu

1. Enter your password after the greeting. Note that this step is completely silent.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
\* Welcome to the UW-Madison Biochemistry Computational Cluster \*
\* \*
\* USE /scratch FOR JOB DATA! DO NOT STORE DATA IN YOUR USER FOLDER!!! \*
\* MOVE YOUR RESULTS TO OTHER STORAGE AFTER YOUR JOB COMPLETES, ALL DATA \*
\* MAY BE REMOVED BY ADMINISTRATORS AT ANY TIME!!! \*
\* \*
\* This computer system is for authorized use only. \*
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
-----@submit.biochem.wisc.edu's password:

## useful reminders

### Linux version

It is sometimes critical to know the linux version that is installed, for example is it 32 or 64 bit?

The command uname -a will provide the answer:

uname -a

Linux submit.biochem.wisc.edu 2.6.32-642.15.1.el6.x86\_64 #1 SMP Thu Feb 23 11:19:57 CST 2017 x86\_64 x86\_64 x86\_64 GNU/Linux

Therefore we can deduct that we are running a 64 bit linux: x86\_64 which is compiled on Intel chip or compatible (x86)

Some other information is more cryptic: el6 means “Enterprise Linux version 6” which is derived from the Enterprise Linux 6 version from Red Hat Linux.

Some other aspects would need more research, but we can also find what “derived” version we are running with the following command specific to Red Hat family:

cat /etc/redhat-release

Scientific Linux release 6.8 (Carbon)

Some of this information will be necessary later when choosing binaries to download.

### Environment variables

Environment variables are akin to “preferences” which are set-up at login time.

*Note*: these are always written in CAPS as a programming convention.

The command printenv will type **all** variables on the screen with their current values.

The command printenv SOMEVARIABLE will print only the value of the requested variable.

Here are just a few useful variables to remember and that will be used today:

* printenv $HOME will print your working directory
* printenv $USER will print your username

### HOME directory

Upon login you will land within your **home** directory.

You can always get back there with either “commands”

cd
cd ~
cd $HOME

### Know where you are:

You can always know where you are in the system with:

pwd

# Set-up directories

It is best to create separate directories for various projects.

*Note*: the BCC “knows” about the /scratch directory which facilitates things to some level. As stated in the greetings at login: USE /scratch FOR JOB DATA! DO NOT STORE DATA IN YOUR USER FOLDER!!!

Therefore we will create everything within the scratch directory.

**TASK**

This button will invite you to act on **Move to scratch and set-up**.

cd /scratch

We now need to create a directory within /scratch with your name on it. You can either use $USER or type your actual username. *Note*: By using the variable the command will work for all!

mkdir $USER

We will now work from within this new directory:

cd $USER
pwd

Create directories, one for Autodock and one for Autodock Vina which we can simply call *Vina*

The use of uppercase can make it easier later to distinguish the folder from the software

mkdir AUTODOCK
mkdir VINA

# Download binaries

On the BCC cluster users have to either *compile* their own software or download pre-compiled binaries to be installed.

Binaries can be compiled with ***dynamic*** libraries, they are perhaps smaller but require the libraries to be pre-installed on the cluster, which is not always the case.

Therefore downloading ***static*** libraries is usually a better practise for using on BCC.

## Where to find binaries?

For open-soure software this is typically found on the “**Downlads**” page of the supporting web site.

For example, the Autodock Vina download page contains:

Download:

The current version is 1.1.2 (May 11, 2011).

Windows autodock\_vina\_1\_1\_2\_win32.msi (0.5 MB) Compatibility, installation and usage notes
Linux autodock\_vina\_1\_1\_2\_linux\_x86.tgz (1.2 MB) Compatibility, installation and usage notes
MacOSX autodock\_vina\_1\_1\_2\_mac.tgz (0.9 MB) Compatibility, installation and usage notes
Source autodock\_vina\_1\_1\_2.tgz (browse) (0.1 MB) Building from source

After exploring the web site we can “capture” the URL for the binary and download it **directly** within the cluster with help of the “web get” command wget. Of course it is best to be within the correct directory first.

**TASK**

This button will invite you to act on **Get and install binaries**.

## Install Vina

cd VINA
wget http://vina.scripps.edu/download/autodock\_vina\_1\_1\_2.tgz

When the download is done we need to un-archive and un-compress the file with the single tar command: (x = extract, z = the file is compressed, v= verbose - show what is happening, f use a file rather than a physical magnetic tape - tar was short for *Tap ARchive*.)

tar xzvf autodock\_vina\_1\_1\_2.tgz

autodock\_vina\_1\_1\_2\_linux\_x86/
autodock\_vina\_1\_1\_2\_linux\_x86/LICENSE
autodock\_vina\_1\_1\_2\_linux\_x86/bin/
autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina
autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina\_split

*Note*: The executable is called vina within the bin directory.

We will have to remember where things are later, but all should now be within /scratch/$USER/VINA/autodock\_vina\_1\_1\_2\_linux\_x86.

### Vina tutorial files

**TASK**

This button will invite you to act on **Install Vina tutorial files**.

Later we will use Vina tutorial files that we can download right now.

The [Vina tutorial web page](http://vina.scripps.edu/tutorial.html) provides the link to a .zip file that we will download. There is also a *YouTube* video detailing the creation of the files.

**TASK**

This button will invite you to act on **Get Vina tutorial files**.

*Note*: pwd should tell you are in /scratch/$USER/VINA - If not rectify with appropriate cd command(s)

wget http://vina.scripps.edu/vina\_tutorial.zip

We then need to unzip the file:

unzip vina\_tutorial.zip

Archive: vina\_tutorial.zip
 creating: vina\_tutorial/
 inflating: vina\_tutorial/ligand.pdb
 inflating: vina\_tutorial/ligand\_experiment.pdb
 inflating: vina\_tutorial/protein.pdb

*Note*: We need 2 more files (prepared pdbqt files) that we’ll download from the Biochem web site. The files have “secutiry names” to abide by security file naming convention(s) and after download we’ll need to rename them and place them within the tutorial directory. The method to create these files is detailed in the “YouTube” [Vina tutorial web page](http://vina.scripps.edu/tutorial.html).

Get the protein PDBQT file:

wget https://biochem.wisc.edu/sites/default/files/facilities/bcrf/tutorials/Autodock/protein.pdbqt\_.txt

Get the ligand PDBQT file:

wget https://biochem.wisc.edu/sites/default/files/facilities/bcrf/tutorials/Autodock/ligand.pdbqt\_.txt

We now need to rename and move these files into the vina\_tutorial directory:

mv protein.pdbqt\_.txt ./vina\_tutorial/protein.pdbqt

mv ligand.pdbqt\_.txt ./vina\_tutorial/ligand.pdbqt

## Install Autodock

**TASK**

This button will invite you to act on **Install Autodock**.

### Which Autodock binary?

While we are in installation mode we can also install Autodock for later.

The [download page](http://autodock.scripps.edu/downloads/autodock-registration/autodock-4-2-download-page/) has downloads options for multiple platforms.

For linux there is a choice between 32 and 64 bit. This will be dependent on the hardware at hand (see above for specification of the linux version run on BCC.)

Specifically for linux the download options are:

* Linux: Intel (32-bit) (667K) md5sum e3b18a7f399525c6edbea4b05f26e850
* Linux: Intel (64-bit) based on command uname -r output:
– **2** - Linux: Intel (64-bit) (743K) md5sum 8c175d4f7b9b1529fdf8d3abf9c90772
– **3** - Linux: Intel (64-bit) (764K) md5sum 0ff500576d03abd97c8e543af6e99dd2

Which version? We already know that we need a 64 bit version.

Then there is a hint about choosing between 2 and 3:

uname -r

2.6.32-642.15.1.el6.x86\_64

The resulting output starts with a ***2*** and therefore that is the one we’ll need.

Hint: you can download 3 but if you try to run it something will be missing… (./autodock4: /lib64/libc.so.6: versionGLIBC\_2.14’ not found (required by ./autodock4)`)

### Download and install

Before download and install we need to go to the correct directory!

cd /scratch/$USER/AUTODOCK

Then we download directly from the web:

wget http://autodock.scripps.edu/downloads/autodock-registration/tars/dist426/autodocksuite-4.2.6-x86\_64Linux2.tar

The next step is to unpack:

tar xvf autodocksuite-4.2.6-x86\_64Linux2.tar

x86\_64Linux2/autodock4
x86\_64Linux2/autogrid4

Note that here there is no bin directory compared to the Vina installation.

# Vina tutorial

The purpose of this tutorial is to run Vina on the linux cluster.

The preparation of files is detailed on the [Vina tutorial web page](http://vina.scripps.edu/tutorial.html) and we downloaded most of them already.

The PDB files need to be arranged so that atoms are named properly, hydrogens are added and charges assigned. When this is done the original PDB data is saved in the PDBQT format which encodes this extra information.

The necessary files for running Vina are:

* protein structure: protein.pdbqt
* ligand structure: ligand.pdbqt
* optional: a configuration file to contain all command options: conf.txt

*Note*: Vina does not require a “grid” file as (Autodock does) as the grid is computed automatically during the run.

## Create configuration file

We already have the PDBQT files, we now need to create the configuration file. Note that this file is optional and all options could be given on the command line, but it is easier to procede in this fashion.

For this purpose we need to edit a simple text file. There are various ways to go about this, one of them would be to create this plain text file on the Mac (or Windows) and then transfer it to the cluster. However, there are risks of complications in proceeding in this manner and it is much simpler to create the file on the cluster.

For this we can use the full-screen text editor nano as it is easy to use. (If you know how to use vi or vim you can certainly use that. emacs does not seem to be installed.)

The configuration file will contain:

* receptor: file name for the protein
* ligand : file name for the ligand
* out : output all configurations of the computed ligand positions in a single file
* center\_x, center\_y, center\_z: center location where binding will be computed
* size\_x,size\_y, size\_z: size of the “box” where binding is explored

*Note*: Flexibility of specific bonds is determined during the creation of the PBDQT file.

**TASK**

This button will invite you to act on **Use nano to create a configuration file**.

We will call the file simply conf.txt and we can already let nano that this will be the name:

nano conf.txt

Within the writing area fill-in the information that we’ll pass on Vina:

 GNU nano 2.0.9 File: conf.txt Modified

receptor = protein.pdbqt
ligand = ligand.pdbqt

out = all.pdbqt

center\_x = 11
center\_y = 90.5
center\_z = 57.5

size\_x = 22
size\_y = 24
size\_z = 28

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

When you are done writing, use Ctrl-X to exit

When asked Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ? answer Y for YES

Then when asked File Name to Write: conf.txt simply press return to confirm the file name.

Verify that the file contains what you expect by typing its content on the screen:

cat conf.txt

receptor = protein.pdbqt
ligand = ligand.pdbqt

out = all.pdbqt

center\_x = 11
center\_y = 90.5
center\_z = 57.5

size\_x = 22
size\_y = 24
size\_z = 28

## Create HTCondor files

HTCondor reference (Tannenbaum et al. 2001)

We now need to create HTCondor file to schedule the run.

We will need to create the following files:

* vina.sh: a short shell script that will know where to locate and run vina with the configuration file
* vina.sub: set of commands to submit to HTCondor

For simplicity we will create these files within the vina\_tutorial directory. To make sure we are in the correct location:

cd /scratch/$USER/VINA/vina\_tutorial

### vina.sh

This file is the “executable” that HTCondor will run. Within it is all the information necessary to accomplish a run.

We will need to know the following:

1. *Where* is vina ?
2. *Where* are the PDBQT files to use?
3. *Where* is conf.txt ?
4. *How* to ask for a vina run?

The answers are:

1. vina is located in /scratch/$USER/VINA/autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina - However HTCondor DOES NOT KNOW “who” $USER is so please write YOUR username instead.
2. PDBQT files should be within: /scratch/$USER/VINA/vina\_tutorial
3. conf.txt should be within: /scratch/$USER/VINA/vina\_tutorial
4. We can verify that vina is “executable” with an ls -l command: if there are x in the permission list at right then it is executable. If not, a special command can make it so (to be reviewed in class if necessary.)

We are now “almost” ready to create the file. Since HTCondor does not understand who is $USER we can “print” the complete *path* beforehand and use the iMac Copy (or command+c) to retain the expanded location within the clipboard:

ls /scratch/$USER/VINA/autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina

In ***MY CASE*** the answer will be:

/scratch/jsgro/VINA/autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina

In YOUR CASE it will reflect YOUR username.

*Important Note*: HTCondor “knows” where /scratch is located and we take advantage of this fact: we give the “absolute PATH” starting with /scratch and therefore we DO NOT NEED to transfer the vina software to run it, it is accessed on the /scratch drive.

**TASK**

This button will invite you to act on **Use nano to create a vina.sh**.

* On the first line type: #!/bin/bash (this is standard to specify the shell interpreter)
* **Copy** YOUR vina location in the clipboard as detailed above.
* Use nano to create a new file called vina.sh:
* **Paste** the vina location
* add the name of the configuration file with --config conf.txt
* Exit nano with Ctrl-X and Y to preserve the file name.

Check the content of your file. Except username it should look like this:

cat vina.sh

#!/bin/bash
/scratch/jsgro/VINA/autodock\_vina\_1\_1\_2\_linux\_x86/bin/vina --config conf.txt

## Submit file

We now need to create a “submit” file to tell HTCondor what we want to do, including running the vina.sh file we just created.

There are many ways to configure a submit file, we’ll keep options to minumum.

* We need to declare the HTCondor “Universe.” VANILLA is the default but printed here as some other system my have a different default
* Some files (but not the vina software - see above) need to be transferred: PDBQT files for example

**TASK**

This button will invite you to act on **Use nano to create a vina.sub**.

Enter the following information:

Universe = vanilla
Executable = vina.sh
transfer\_input\_files = conf.txt, ligand.pdbqt, protein.pdbqt
should\_transfer\_files = Yes
when\_to\_transfer\_output = ON\_EXIT

output = job.out.$(Process)
error = job.error.$(Process)
log = job.log.$(Process)
Queue 1

## Submit the job

We are now ready to submit the job:

condor\_submit vina.sub

Submitting job(s).
1 job(s) submitted to cluster 178298.

*Note*: the job number may be useful to remove unwanted jobs from the queue.

The condor\_submit command has a very large number of options detailed within [its online manual entry](http://research.cs.wisc.edu/htcondor/manual/current/condor_submit.html).

To check if the job is running:

condor\_q $USER

-- Schedd: submit.biochem.wisc.edu : <128.104.119.165:9618?... @ 04/18/17 11:47:31
OWNER BATCH\_NAME SUBMITTED DONE RUN IDLE TOTAL JOB\_IDS
jsgro CMD: vina.sh 4/18 11:47 \_ 1 \_ 1 178298.0

1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended

On the last line we can see that we have 1 running

## Results

“As is” the request may take 5 to 6 minutes to run.

List all files in the directory in time-reserve order

ls -lth

Output truncated on left:

 1.3K Apr 18 11:52 job.log.0
 27K Apr 18 11:52 all.pdbqt
 1.7K Apr 18 11:52 job.out.0
 0 Apr 18 11:47 job.error.0
 89 Apr 18 11:46 vina.sh
 258 Apr 18 11:46 vina.sub
 149 Apr 18 11:06 conf.txt
 3.8K Apr 18 10:20 ligand.pdbqt
 212K Apr 18 10:20 protein.pdbqt
 3.7K Nov 13 2008 ligand\_experiment.pdb
 3.9K Nov 13 2008 ligand.pdb
 172K Nov 13 2008 protein.pdb

The final result is in file all.pdbqt and we can detect how may conformations were calculated with the very simple grep command searching for the PDB code MODEL:

fgrep MODEL < all.pdbqt

MODEL 1
MODEL 2
MODEL 3
MODEL 4
MODEL 5
MODEL 6
MODEL 7

### Transfer result file to local computer

To transfer the file to your local computer for futher analysis we can use the sftp command method.

**TASK**

This button will invite you to act on **Copy results to local computer**.

The easiest is to open a new Terminal from the Terminal program with the menu cascade:

**Shell > New Window > Choose a color option or basic**m (I often use “Ocean”)

Before we connect it is a good idea to point the new terminal to look *e.g.* on the Desktop

pwd
cd ~/Desktop

We will use this new window to connect with sftp

sftp YOURUSERNAME@submit.biotech.wisc.edu

@submit.biochem.wisc.edu's password:
Connected to submit.biochem.wisc.edu.
sftp>

The sftp prompt mean that we can issue commands. Some commands are identical or similar to those of the bash shell. However, $USER or TAB-completion do not work.

We first need to “go” to the appropriate folder and list content:

sftp> cd /scratch/jsgro/VINA/vina\_tutorial
sftp> ls
all.pdbqt conf.txt job.error.0
job.log.0 job.out.0 ligand.pdb
ligand.pdbqt ligand\_experiment.pdb protein.pdb
protein.pdbqt vina.sh vina.sub
sftp>

We can get any file from here, one at a time with get or multiple files at a time with mget:

With get the ***exact*** file name is required

sftp> get all.pdbqt
Fetching /scratch/jsgro/autodock\_vina/vina\_tutorial/all.pdbqt to all.pdbqt
/scratch/jsgro/autodock\_vina/vina\_tutorial/all.pdb 100% 27KB 27.0KB/s 00:00
sftp>

With mget we can use the “wild card” \* to replace most of the file names:

sftp> mget \*.pdbqt

The files are now located on the Desktop, or any othe directory decided *before* using the sftp command to connect.

## Requesting more CPUs

From the [Vina manual](http://vina.scripps.edu/manual.html) web page:

*Vina can take advantage of multiple CPUs or CPU cores to significantly shorten its running time*.

It is possible to request multiple CPUs when sumbitting the job, for example:

condor\_submit request\_cpus=6 vina.sub

In my case this finished *approximately* in less than 2 minutes rather than 6 min previously with a single processor.

There are ways to make this “requirement” part of the submit file itself.

# Files preparation tutorials

There are multiple tools available to prepare files for Autodock or Autodock Vina. There are many references to ADT (Autodock Tools) to prepare files but there are other options as well, including UCSF Chimera and VMD.

* Autodock tutorial with Chimera (PowerPoint) : <https://en-lifesci.tau.ac.il/sites/lifesci_en.tau.ac.il/files/media_server/life%20sci/bioinformatics/autodock_tutorial1.pptx>
* Vina with Chimera: <http://www.free-bit.org/course/2014-SriLanka/pdf/034-chimera_vina.pdf>
* Molecular docking tutorial with VMD ADT and Autodock: <https://sites.ualberta.ca/~pwinter/Molecular_Docking_Tutorial.pdf>
* Using AutoDock with AutoDockTools: A Tutorial - <http://autodock.scripps.edu/faqs-help/tutorial/using-autodock-with-autodocktools/UsingAutoDockWithADT_v2e.pdf>
* Molecular Docking: Tutorial - Docking with Autodock Vina: A step by step guide for Beginners or Advanced Users (with MarvinSketch and OpenBabel.) <https://cbiores.com/molecular-docking-tutorial/>
* Protein- Ligand Interaction <http://vlab.amrita.edu/?sub=3&brch=275&sim=1495&cnt=2>
* peptide docking protocol Rosetta FlexPepDock <http://aidanbudd.github.io/ppisnd/trainingMaterial/oraSchuelerFurman/FlexPepDock%20Tutorial_1.6.2016.pdf>

# Manuals:

Autodock user guide: <http://autodock.scripps.edu/downloads/faqs-help/manual/autodock-4-2-user-guide/AutoDock4.2.6_UserGuide.pdf>

Autodock Tutorials: <http://autodock.scripps.edu/faqs-help/tutorial/>

Vina manual: <http://vina.scripps.edu/manual.html>

# Acknowledgments

This tutorial is based on the following online resources:

* OSGrid [AutoDock Vina](https://support.opensciencegrid.org/support/solutions/articles/5000634379-autodock-vina#tutorial-files)
* All OSGrid files can be downloaded here: <https://github.com/OSGConnect/tutorial-AutoDockVina>
* Preparation for Autock Vina PDBQT and conf.txt files: <http://vina.scripps.edu/tutorial.html> and embedded video

# REFERENCES

Chen, Y. C. 2015. “Beware of docking!” *Trends Pharmacol. Sci.* 36 (2): 78–95. <http://dx.doi.org/10.1016/j.tips.2014.12.001>.

Morris, G. M., R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson. 2009. “AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility.” *J Comput Chem* 30 (16): 2785–91. <http://autodock.scripps.edu>.

Tannenbaum, Todd, Derek Wright, Karen Miller, and Miron Livny. 2001. “Condor – a Distributed Job Scheduler.” In *Beowulf Cluster Computing with Linux*, edited by Thomas Sterling. MIT Press.

Trott, O., and A. J. Olson. 2010. “AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading.” *J Comput Chem* 31 (2): 455–61. <http://vina.scripps.edu>.