USF Engineering Biology Core

Neurotransmitters Docking Protocol

Written by Lalith Roopesh | Updated 01/09/2023

Table of Contents

Important Note: I would take your time to understand the concepts in these instructions instead of doing everything word-for-word! Not only will the latter method fail since everyone's computers are different and you need some understanding of modifications, but you will probably be using these skills in the near future!

~ Lalith

Special thanks to Emmu Wu for testing and Amreen Naveen for contributions.

1. Installation Processes

This step consists of getting started with CIRCE and installing the necessary programs for docking. **Once you have finished this step, you will not need to do it again.**

1.a. Logging into CIRCE

Open up Command Prompt or Terminal, I will call these command line. You can search for this in your computer's search bar.

- Quick tips for command line:
  - You can highlight and copy things with Ctrl+C
  - **You can paste in command line by right clicking**
- NETID refers to your USF NetID, the text before your email. For example, if my email were lalithr@usf.edu then my NETID would be lalithr. Replace the NETID with your NetID in all following references.

Type the following into command line to access CIRCE:

```
ssh NETID@sc.rc.usf.edu
```

This code accesses CIRCE via secure shell (SSH) protocol. You should be prompted for a password, type in your normal USF password. This entire process is considered logging into CIRCE.

Sometimes your connection may reset and you will have to repeat this.

1.b. Navigating CIRCE and making a folder

To see all the files in your home path run the following:

```
file *
```

There should be nothing, as there is nothing right now. Let's make a folder for vina stuff! Run the following:

```
mkdir vina
```

Mkdir is the command for making directories, which is a fancy name for folder. Run file * again, and you should see there is a vina directory. Now, let's go into the vina directory by typing the following:

```
cd vina
```

Cd is the traveling command. By doing cd vina, you're telling your computer to go into the directory named vina. This is the path you'll be uploading your files to:

```
NETID@sc.rc.usf.edu:~/vina/
```

This says that the location you are relative to your home location (NETID@sc.rc.usf.edu) is inside the folder ~ then inside the folder vina. **Remember this path!**

1.c. Installing Autodock Vina

CIRCE SC runs on Linux like most complicated computing things, so we will have to download the Linux file. Go to https://vina.scripps.edu/downloads/ and download the Linux version, it is the link next to the cute penguin. I put it in my Downloads folder. You will probably not be able to open it, don't try. Open your file explorer, find the document, right click on it and click copy as path (Mac users can Ctrl+click and copy as pathname). This should have the following copied:

```
"C:\Users\lalit\Downloads\autodock_vina_1_1_2_linux_x86.tgz"
```

This path will be different for everyone, but make sure it's the correct path with the correct file name. **Make sure to remove the quotation marks** and copy it again. Open a NEW command line (Command Prompt or terminal. Type the following:

```
scp path1 path2
```

BUT replace path1 with your Vina file path ex. C:\Users\lalit\Downloads\autodock_vina_1_1_2_linux_x86.tgz and path2 with your CIRCE folder path ex. NETID@sc.rc.usf.edu:~/vina/ (but replace NETID with your NETID). Scp is a command that just copies a file from one path to another. For example, my command would look like:

```
scp C:\Users\lalit\Downloads\autodock_vina_1_1_2_linux_x86.tgz
lalithr@sc.rc.usf.edu:~/vina/
```

But with a space instead of an enter (I ran out of space). If this works, you will be prompted to re-enter your password and you should see your file be copied. You can close this command line window now and go back to the one logged into CIRCE.

If you are still in the vina folder, do the file * command again. This should show a compressed file. You can copy the name of the file by highlighting just the name part and clicking copy. You can use that to paste autodock_vina_1_1_2_linux_x86.tgz at any point by right clicking. Run the following in order:

```
tar xvf autodock_vina_1_1_2_linux_x86.tgz
cd autodock_vina_1_1_2_linux_x86
cd bin
file *
```

This first unzipped the compressed tar file with the tar xvf command, then went into the newly unzipped folder, went inside the bin folder inside it, then showed the contents with file *, which should be two LSB executable files labeled vina and vina_split. Time for some cleanup:

```
mv vina vina_split ../../
cd ../../
ls
```

This moved (mv) the vina executable files back into the vina directory. Ls is a simplified file * that just lists (ls) the names. **Confirm that the vina and vina_split are in the folder before you do anything else.** We will clear away unwanted stuff with the following:

```
rm -r autodock_vina_1_1_2_linux_x86
rm autodock_vina_1_1_2_linux_x86.tgz
ls
```

This removes (rm) those files. Ensure that only vina and vina_split are in the folder now.

1.d. Installing Other Tools

If you do not already have these installed, install the following:

- Discovery Studio: https://discover.3ds.com/discovery-studio-visualizer-download
    - If you cannot, PyMol should be able to do the same things: https://pymol.org/2/ and download an **educational** license here: https://pymol.org/edu/
- Autodock Tools: https://ccsb.scripps.edu/mgltools/downloads/
    - Mac users will likely not be able to use this. See **Appendix A. Using Autodock Tools within CIRCE** for instructions on how to use Tools within CIRCE.

Try running both programs to make sure they work.

2. Preparing Docking Materials

This step will consist of getting all the docking materials for the actual docking process together. Make sure you are aware of exactly which receptors you are doing, they must all have the same grid box to do them at the same time. Repeat for all sets you are doing.

2.a. Converting files to PDBQTs

Create a folder somewhere to store all of your docking materials for one run on your computer, ensuring the name has no spaces. I made mine on my Desktop so that the folder path is:

```
C:\Users\lalit\Desktop\neurodocking
```

Open AutoDock Tools click ligand > input > open and select your ligand file (it should only show .pdbqt files, so make sure to change so it shows all file types). Do not make any modifications such as adding hydrogens, calculating hydrogen bonds or charges, or adding torsions, as they break the tool. Then go to Ligand > Output > Save as .pdbqt into the desired folder with the name oxytocin, vasopressin, or dopamine only (nothing else). The final file name should only be oxytocin.pdbqt, vasopressin.pdbqt, or dopamine.pdbqt. **Since you are only using one ligand per receptor you will only do this once.**

In Discovery Studio, open the protein .pdb, double/triple click so all of the ligand/cholesterol is selected, then right-click and cut out. Save file as .pdb (should be just the protein by itself). Open AutoDock Tools and open the protein file. Do not make any modifications such as adding hydrogens, calculating hydrogen bonds or charges, or adding torsions, as they break the tool. Select Grid > Macromolecule > Choose > Select the protein > Save as .pdbqt into desired folder. **Make sure the file name is in the format "receptorcode_#_sometext.pdbqt"** where receptorcode is DRD1, DRD2, OXTR1, or AVPR1, for example "OXTR1_52_A0A6P3EIT3.pdbqt". **You will need to do this once for every single receptor**.

2.b. Creating Config Files for Docking

Open a command line and run the following:

```
py -3
```

If using a Mac, run "python3" instead. If it does not open a Python IDE, usually shown by saying something along the lines of "Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC

v.1929 64 bit (AMD64)] on win32" (your text might be different) then install the latest version of Python from here: https://www.python.org/downloads/. If it does, run "exit()" to leave the IDE.

Once that works, download the **create_configs.py** script from here. If it doesn't work, the script is also available in raw text form in **Appendix B. Scripts** to make manually. Place this script in your PDBQTs folder from step 2.a, in my case it is the "neurodocking" folder. Right click the folder and click Copy as Path. For example, I would have the following copied:

```
C:\Users\lalit\Desktop\neurodocking
```

Ensure all of your receptor PDBQTs and ligand PDBQT are in the same folder as the script. Then open a new command line and run the following commands (using the copied path from above):

```
cd C:\Users\lalit\Desktop\neurodocking
py -3 create_configs.py
```

You will be asked to confirm several things, if you agree then just type in "y" (must be lowercase) and hit enter. Go through the steps to set up all of your variables, and **remember your job name!** Get the center and size variables for your specific grid box in the template spreadsheet in our Box folder. Any other response will end the program, but you can just try again. Once run successfully, this should generate all of your config and script files needed in that folder. **Please go through and click on a few of them to verify they look correct.**

3. Using Vina in CIRCE

Docking with vina once you've used the configs creation tool is much simpler than doing a single run like before, since all the files are made already!

3.a. Docking in Vina

Stay or cd into the directory from the previous step 2.b. For me it is:

```
C:\Users\lalit\Desktop\neurodocking
```

We will be copying all of the relevant files to your vina folder by running:

```
scp *.pdbqt *.txt *.sh NETID@sc.rc.usf.edu:~/vina
```

You will be asked to enter your password and then make sure all files are copied. Once this is done, open a new command line and ssh back into CIRCE. Run the following commands:

```
cd vina
dos2unix *.pdbqt *.txt *.sh
ls
```

After the ls command, you should be able to see a file labeled "JOBNAME.sh" where JOBNAME refers to the job name you submitted to the create_configs script. Submit to the queue via:

```
sbatch JOBNAME.sh
```

Note the job number, it should show something like "Submitted batch job #" Wait about 20 seconds, and check if your code is running using:

```
squeue
```

It should show your job and NetID with status in the ST column. Make sure it shows up or is R (running). You can recheck its status whenever you want. If it is running, you can also check it's progress using:

```
cat slurm-#.out
```

Where # is your batch number. That shows the output from the program, usually with a percentage complete bar for the individual receptor it is on. Remember this file name for future reference.

3.b. Obtaining Vina Results

When you have been emailed about the job completing, navigate back to your Vina folder. Do ls or file * to see that your output file has been obtained. Now, you will need to obtain the output files to analyze on your computer. Open a NEW command line that is not logged into CIRCE and navigate to a NEW folder for results, not the folder that you used in Step 2. For example, the command I used to navigate was:

```
C:\Users\lalit\Desktop\neurodocking_results
```

From here, you will need to run the following:

```
scp NETID@sc.rc.usf.edu:~/vina/*_output.pdbqt ./
scp NETID@sc.rc.usf.edu:~/vina/slurm-#.out ./
```

is your job's batch number. If you forgot, do file * in your CIRCE command line to see the name of the file. Both times you should be asked for your password. You now have these files to work with on your computer, and the .out file is openable via any text editor such as Notepad (right click, click "open with"). To clean up your vina folder in CIRCE, navigate to it and run:

```
rm *.txt *.pdbqt *.out *.sh
```

Which will remove all files with those file endings, essentially everything you used besides the Vina program. You may also specify file names with it instead.

4. Using and Recording Results

TBD! Make sure to keep the out and PDBQT output files.

Appendix A. Using Autodock Tools within CIRCE

A.1. Tools Installation Process

Read through and install X2GO client https://wiki.rc.usf.edu/index.php/SC_Desktop_Environment with the exact directions on the website. This is a method of accessing CIRCE with a visual display since that is needed for Autodock Tools, but I still recommend only using command line for the actual docking procedure. If you have issues opening and running X2GO, I would recommend checking the settings listed on that website.

Outside of X2GO on your own computer, download the 1.5.7 version of MGLtools for Linux at https://ccsb.scripps.edu/mgltools/downloads/. This should be the top download link, next to a penguin that says 1.5.7 and ends with file ending .tar.gz. Once downloaded, go find the download in your file explorer, right click, and select copy as path. This may be copied in quotation marks, so paste it somewhere, delete the quotation marks, and copy it again. It will look something like:

```
C:\Users\lalit\Downloads\mgltools_x86_64Linux2_1.5.7p1.tar.gz
```

But obviously different for your own computer. Open up a command line on your computer and run the following:

```
scp path1 NETID@sc.rc.usf.edu:~/
```

BUT replace path1 with the file path, C:\Users\lalit\Downloads\mgltools_x86_64Linux2_1.5.7p1.tar.gz for me, and NETID with your own NetID in lowercase. This prompt you for your password and the file should be copied in. Next log in to CIRCE and do the following:

```
tar xvf mgltools_x86_64Linux2_1.5.7p1.tar.gz
cd mgltools_x86_64Linux2_1.5.7
./install.sh
```

This unzips the file, opens it, and runs the installation process. Once that is done, run the following twice to quit properly your command line session.

```
exit
```

Autodock Tools should now be properly installed inside your X2GO session! **You will only need to do this part of the process once**, make sure to only do the using Autodock Tools process from now on.

A.2. Using Autodock Tools Process

You will first need to send the PDB files to your CIRCE environment. I put all of mine (every model for what I'm docking) in my desktop neurodocking folder so the path is:

```
C:\Users\lalit\Desktop\neurodocking
```

But yours may be different. Anyways, in your command line do:

```
cd path
```

Where path is your folder's path, and mine is C:\Users\lalit\Desktop\neurodocking. You can check the contents with "dir" for Windows and "ls" for MacOS. From there, do

```
scp *.pdb NETID@sc.rc.usf.edu:~/
```

With your actual file names and NetID of course (the copy/paste function may be helpful here, but make sure its a space and not enter). This will copy all both files into CIRCE. You should once again be prompted for your password. Once this is done you can close this command line.

Open up your X2GO client by opening up the app (you can use your search bar to find it) and double clicking on the session created. You will be prompted to use your password. **Inside** X2GO open up a command line, it should be called Terminal. Run the following:

```
echo $DISPLAY
export DISPLAY=$(hostname)$DISPLAY
echo $DISPLAY
```

The second echo $DISPLAY should show **"sclogin0.rc.usf.edu:#"** where # is the number shown in the first echo $DISPLAY. Then run the following commands to cd to where you need to be and run:

```
ls
cd mgltools_x86_64Linux2_1.5.7
cd bin
srun --mem=16384 --time=02:00:00 --pty ./adt
```

The first ls is to allow you to copy paste the name of the folder. This should then run Autodock Tools for 2 hours, but can be adjusted. Use the instructions in **2.a. Converting files to PDBQTs**. If you run out of time, start over from the first echo $DISPLAY command. **Make sure to save all of your PDBQTs to your home directory.** Once you are done, **do not** simply close out. Open a NEW command line in X2GO, and run the following:

```
squeue
scancel #
```

Where the # is the JOBID you see next to your NETID when you run squeue. To copy all the files back to your home directory, open a command line on your actual computer and run the following:

```
cd path
scp NETID@sc.rc.usf.edu:~/*.pdbqt ./
```

Where path is the same path you used at the beginning of this section, mine being “C:\Users\lalit\Desktop\neurodocking”.

Appendix B. Scripts

B.1. create_configs.py

Note: indentation may not be preserved while copy+pasting. You may need to manually indent.

```
# Lalith Roopesh 2022

import os, fnmatch, sys

# confirmations
x = ""
while not(x.lower() == "y"):
    x = str(input("Confirm that ONLY the PDBQTs with the same grid box are in
the folder this file is in. (y/n) > "))
x = ""
while not(x.lower() == "y"):
    x = str(input("Confirm that your ligand (ex. dopamine, vasopressin, or
oxytocin) are in the folder as a PDBQT file (y/n) > "))

# some job parameters

r_possible = ["oxytocin", "dopamine", "vasopressin"]


rname = ""
while not(rname in r_possible):
    rname = str(input("""Enter your ligand file name. Your file should be named
this name with .pdbqt ONLY.
Possibilities are: oxytocin, dopamine, vasopressin
> """))

job_name = str(input("Enter the name of the job. Don't use special characters
(besides _ and -) ex. OXTR1_1-39. > "))
netid = str(input("Enter your NetID > "))

# config variables

center_x = float(input("Input center_x: "))
center_y = float(input("Input center_y: "))
center_z = float(input("Input center_z: "))
size_x = float(input("Input size_x: "))
size_y = float(input("Input size_y: "))
size_z = float(input("Input size_z: "))

# standardized config variables
```

```
exhaustiveness = 64
cpu = 24

decision = str(input("""
==========================================================================

Here are your docking settings:

JOB DETAILS
> Job name: {}
> Ligand: {}
> Your NetID: {}
> Working directory: {}

CONFIG SETTINGS
> center_x = {}
> center_y = {}
> center_z = {}
> size_x = {}
> size_y = {}
> size_z = {}

STANDARDIZED SETTINGS
> The research organizers have standardized the following values:
> exhaustiveness = {}
> cpu = {}

REVIEW
> This command will overwrite all files currently in this directory.
> Review the previous information.
> Proceed? (y/n)
> """.format(job_name, rname, netid, os.getcwd(), center_x, center_y, center_z,
size_x, size_y, size_z, exhaustiveness, cpu)))

if not(decision.lower() == "y"):
    sys.exit("Job approval authorization 'y' denied. Job terminated.")

# get files

files = os.listdir("./")

pdbqt_files = []
for f in files:
    if fnmatch.fnmatch(f, "*.pdbqt") and not(f == rname):
        pdbqt_files.append(f)

pdbqt_files.remove("{}.pdbqt".format(rname))
```

```
config_files = []
for p in pdbqt_files:
    pdata = p.split("_")
    cname = "{}_{}_config.txt".format(pdata[0], pdata[1])
    with open(cname, "w") as config:
        config.write("receptor = {}\nligand = {}.pdbqt\ncenter_x = {}\ncenter_y
= {}\ncenter_z = {}\nsize_x = {}\nsize_y = {}\nsize_z = {}\nexhaustiveness =
{}\ncpu = {}\nout = {}_{}_output.pdbqt".format(p, rname, center_x, center_y,
center_z, size_x, size_y, size_z, exhaustiveness, cpu, pdata[0], pdata[1]))
        print("Created file {}".format(cname))
        config_files.append(cname+"\n")

with open("{}_list.txt".format(job_name),"w") as configlist:
    configlist.writelines(config_files)
    print("Created file {}_list.txt".format(job_name))

with open("{}.sh".format(job_name),"w") as job:
    job.write("""#!/bin/bash\n#SBATCH --job-name={}\n#SBATCH
--time=15:00:00\n#SBATCH --mail-type=END\n#SBATCH
--mail-user={}@usf.edu\n#SBATCH --mem=60480\nreadarray -t files <
{}_list.txt\nfor file in "${{files[@]}}"; do\n./vina --config
"$file"\ndone""".format(job_name, netid, job_name))
    print("Created file {}.sh".format(job_name))
```