Getting started with NRGDock

What you will need:

1. Connecting to a Compute Canada cluster
2. Downloading the target and ligand files
3. Get\_Cleft to specify the binding site (<https://github.com/NRGlab/Get_Cleft>)
4. NRGDock
5. Submit jobs to Compute Canada
6. Analyze the results
7. Transferring data from Compute Canada (command line)
8. Transferring data from Compute Canada (GUI; Not necessary for the course)
9. Connecting to a Compute Canada cluster  
   When copying a command, do not copy the quotation mark. To paste the command on Windows, tap the right mouse button. To paste the command on Mac, press the "Command" and "V" keys at the same time.
   * Connecting to the cluster to send commands (Windows):
     1. Press Windows key
     2. Une image contenant texte

        Description générée automatiquementType: "PowerShell"
     3. Hit enter or click on this icon:
     4. Type (replace **username** and **cluster** with the appropriate word):  
        "ssh **username**@**cluster**.computecanada.ca"
     5. If you get a message saying: "The authenticity of host…"
        1. Type: "yes" and press enter
     6. You will then be asked for your password. Type your password and press enter
   * Connecting to the cluster to send commands (macOS):
     1. While pressing the "command" key, press the "spacebar" key
     2. Type: terminal and press "enter"
     3. Type (replace **username** and **cluster** with the appropriate word):  
        "ssh **username**@**cluster**.computecanada.ca"
     4. If you get a message saying: "The authenticity of host…"
        1. Type: "yes" and press enter
     5. You will then be asked for your password. Type your password and press enter
10. Downloading the target and ligand files

For this docking exercise, we will use a target from the DUD-E dataset called akt1 or protein kinase B (Serine/threonine-protein kinase). You have access to two different types of file storage: project and scratch. Project storage is shared by your group and is linked to the professor’s account. It is better suited for smaller files that don’t change very often. Scratch storage has bigger limits for file size and file quantity. It is better suited for bigger files that get written and read often. For more info on storage: <https://docs.alliancecan.ca/wiki/Storage_and_file_management>  
The archive that we will be downloading contains all the necessary files to run a docking experiment. If you wish to try the software on your own target you will need to have the following files in the same folder:

* 1 .pdb file for the target (binding site empty)
* 1 .mol2 file containing all the ligands you want to test   
  + Navigate to your project storage with the following command (replace bold words with appropriate words; your **groupname** will be listed on this page after you login below group name (def-….): <https://ccdb.computecanada.ca/>):  
    "cd projects/**groupname**/**username**"
  + Download the akt1.tar.gz file with the following command:   
    "wget <http://dude.docking.org/targets/akt1/akt1.tar.gz>"
  + Extract all the files from the archive you just downloaded with the following command: "tar xzf akt1.tar.gz"
  + Navigate to the akt1 folder using: "cd akt1"
  + Extract all archived folders: "gzip -d \*.gz"
  + Make a folder called get\_celft: "mkdir get\_cleft"
  + To convert our receptor file from pdb to the mol2 filetype we will need openbabel.
    1. Activate the module: "module load openbabel"
    2. Convert the file from .pdb to .mol2:  
       "obabel receptor.pdb -O receptor.mol2 -d"
  + Navigate back to the main directory: "cd .. "

1. Get\_Cleft

Get\_Cleft is a software developed by the Najmanovich research group that detects cavities within a molecule. This will allow us to specify the binding site.

* Download Get\_Cleft from GitHub: "git clone <https://github.com/NRGlab/Get_Cleft>"
* Navigate to the Get\_Cleft folder: "cd Get\_Cleft"
* To compile the C program: "gcc Get\_Cleft.c -o Get\_Cleft -O3 -lm"
* Run Get\_Cleft using the following command:   
  "./Get\_Cleft -p ../akt1/receptor.pdb -o ../akt1/get\_cleft/ -s -t 3"
* Navigate back to the main directory: "cd .."

1. NRGDock  
   NRGDock was written in Python and uses Python modules and packages (NRGTEN, SciPy, NumPy, etc.). Without installing these modules, it wouldn’t be possible to run the software. The easiest way to accomplish this on a Compute Canada cluster is to create a virtual environment and load the required modules from a file called requirements.txt. This file contains a list of modules and packages as well as their version and if they already exist in the Compute Canada cluster.  
   When using your own targets and ligands, it may be interesting to explore different software parameters such as grid dot spacing and the number of ligand rotations per dot to achieve better results. This can be done by editing the config.txt file in the main software folder (N\_ORIENTATIONS = number of ligand rotations per axis; DOT\_DIVISION = grid dot spacing in angstroms).  
   * Download NRGDock:   
     "git clone <https://github.com/ThomasDesc/NRGDock>"
   * The next step will be to create a virtual environment in python with the following commands:
     1. "module load python"
     2. "virtualenv --no-download ENV"
     3. "source ENV/bin/activate"
     4. "pip install --no-index --upgrade pip"
     5. "pip install -r ./NRGDock/requirements.txt"
2. Compute Canada Jobs  
   Running software on a Compute Canada cluster is achieved by submitting jobs. A job specifies how much processing power you need (1 core, 8 cores, whole CPU, etc.) as well as the time your software is allowed to run and the commands you want to run (1 hour to 7 days; if the time runs out and a job is still in progress, it will be canceled). After your jobs are submitted, they will be placed in a queue and run based on the group’s priority. Running jobs will lower your priority and may affect how long it takes for your jobs to start.  
   * Navigate to the NRGDock folder: "cd NRGDock"
   * Generate job files. Replace **GROUP** with your group (it will be listed on this page after you login below group name: <https://ccdb.computecanada.ca/>):   
     "python3 generate\_jobs.py ../akt1/ **GROUP** False False True"
   * Navigate to the folder called jobs: "cd jobs"
   * You can submit the jobs by typing: "sh start\_jobs.sh"
   * To check the status of your jobs you can type: "sq"
   * You will want to check the status relatively often to keep track of when your jobs will be done running
3. Result analysis  
   By now, your jobs should have finished running and have produced over 2000 result files. We are going to use another software to combine all these files and sort the ligands by CF. It also generates a folder containing the top **X** ligand poses where x is a number defined in the cofig.txt file (default = 10).   
   * Navigate to the docking software folder. If you have a new Terminal/PowerShell session, follow the instructions in step one and use this command (change **bold** words):   
     "cd projects/**groupname**/**username/**NRGDock"
   * Update the software to the latest version: "git pull"  
     If you get an error message use the "git stash" command followed by "git pull"
   * Activate the virtual environment to load all the necessary modules:  
     "source ../ENV/bin/activate"
   * Use the following command to compress and analyze the results for all the jobs you submitted (could take a while): "python3 compress\_results.py akt1 True"
   * The output will be a folder in the results\_processed folder. The folders are named using 2 parameters defined in the config.txt file: number of rotations and spacing between dots in the binding site
4. Transferring data from Compute Canada (command line)  
   To view the results, you will need to download them to your computer.  
   * Copying the result files to your computer:
     1. Open PowerShell (Windows) or Terminal (macOS)
     2. Replace all the bold words with the appropriate ones (Windows):   
        "scp -r **USERNAME**@beluga.computecanada.ca:/home/**USERNAME**/projects/def-**GROUP**/**USERNAME**/NRGDock/results\_processed C:\Users\**WINDOWS\_USER**\Desktop\results\_processed"  
          
        Replace all the bold words with the appropriate ones (macOS):   
        "scp -r **USERNAME**@beluga.computecanada.ca:/home/**USERNAME**/projects/def-**GROUP**/**USERNAME**/NRGDock/results\_processed /Users/**MAC\_USERNAME**/Desktop "
     3. If you are asked for a password, type it and press enter
   * Viewing ligand poses:
     1. Pymol is free software for viewing molecules in 3d. You can download it at the following link (click "download now"): "<https://pymol.org/2/>"
     2. Follow the installation instructions at the following link:  
        "<https://pymol.org/2/support.html?#installation>"
     3. To view a molecule, click and drag the .pdb file for the molecule you wish to view anywhere on the pymol window
5. Transferring data to/from Compute Canada and editing files (GUI; Not necessary for the course)  
   * The easiest way to access the remote storage of a cluster is by using sftp and a piece of software like **FileZilla**. To start, download and install FileZilla at the following link: <https://filezilla-project.org/download.php?type=server>
   * Open FileZilla. Enter the following information:
     1. Host: "beluga.computecanada.ca"
     2. Username: your username
     3. Password: your password
     4. Port: 22
   * Click Quickconnect
   * The bottom half of the screen is divided into two parts. The right side corresponds to the files on the Compute Canada clusters and the left side corresponds to the files and folders on your computer
   * To transfer a file, you must navigate to the proper location on each side and click and drag the file or the folder to the opposite side. You can also delete files by right-clicking on them and selecting delete.

Useful commands for navigating in a command line environment  
After every command press the "**enter**" on your keyboard to submit.

* + "ls": lists all folders and files in the current directory
  + "cd **folder\_name**": replace **folder\_name** with the desired folder.
  + "cd .. ": goes back to the previous folder
  + "cd **path/to/folder**": if you know the complete path to the folder you can replace **path/to/folder** with the desired path
  + "mkdir **foldername**": makes a new folder called **foldername**
  + "sq": shows the queue of jobs you have submitted and other information like reserved time remaining per job
  + "sshare -l -A **groupname**\_cpu -a": Your group's priority will be listed bellow LevelFS
  + "scancel -u **username**": cancel all jobs in queue
  + Pressing the up arrow will show previous commands.
  + A right click on your mouse will paste whatever you copied
  + Pressing "ctrl" and "c" on your keyboard will stop a running program

Config.txt parameters

The following parameters in the config.txt file can be modified:

* N\_ORIENTATIONS: number of rotations per axis (integer, a higher number will slow down the software)
* DOT\_DIVISION: distance between each dot in the binding site (float, a lower number will slow down the software)
* OUTPUT\_PDB: if True, outputs one ".pdb" file per ligand in lowest energy position (True or False)
* KEPT\_PDB\_NUMBER: number of ligands kept after running compress\_results.py (integer)
* CLEAN: if True, compress\_results.py will delete all the individual result files (True, False)