**Moebatch Scripting**

**Author: Greg Szwabowski**

**Purpose:** Moebatch scripting allows certain MOE processes that would typically require a graphical user interface (GUI) to be run on the HPC using batch scripting. For example, molecular conformational searches can be performed on a remote machine using moebatch rather than being performed locally in the context of an open MOE session. Moebatch scripting is useful in two ways:

1. Each moebatch job uses only a single license, while opening the MOE program itself uses 4 licenses. If less licenses are used, more work can be done by everyone who utilizes MOE to perform their research.
2. The HPC will finish jobs quicker than a local machine, allowing for a more efficient workflow. Furthermore, additional research can be done on a local machine once a job that is typically performed locally is sent to run on the HPC.

In addition to performing processes that are built into MOE (conformational searches, pharmacophore searches, etc.) using the -exec argument, moebatch can also run custom SVL scripts that have been written using the -load argument. The following tutorial details how to use the moebatch command line, how to set up batch jobs for running pre-defined MOE functions, and how to run batch jobs with custom scripts.

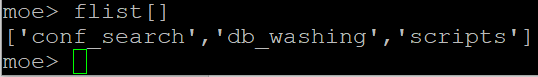
**Basic command line usage:**

* Download the ‘moebatch\_tutorial.zip’ file and upload the folder within the .zip file to your HPC directory using a FTP client like FlieZilla.
* Change your current directory to the folder you uploaded (cd moebatch\_tutorial/).
* After logging in to the HPC via SSH client, the following command will bring up the MOE SVL command line:

/public/apps/moe/moe2018/bin/moebatch

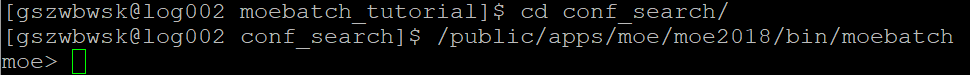


* This command line can be used like the SVL command line in an interactive MOE session, many of the same commands still apply:



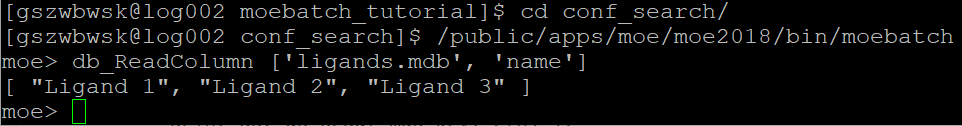


* The moebatch command line can be closed using Ctrl+C on Windows.
* Commands dealing with databases can be performed given that the database is present in the current working directory. Close the command line and set your current working directory to the conf\_search folder (cd conf\_search/), then open the moebatch command line again:



* The moebatch command line will now be open in the conf\_search folder, allowing for access to the database within.
* The following command will read the name of each entry within the ligands.mdb database contained within the conf\_search folder:

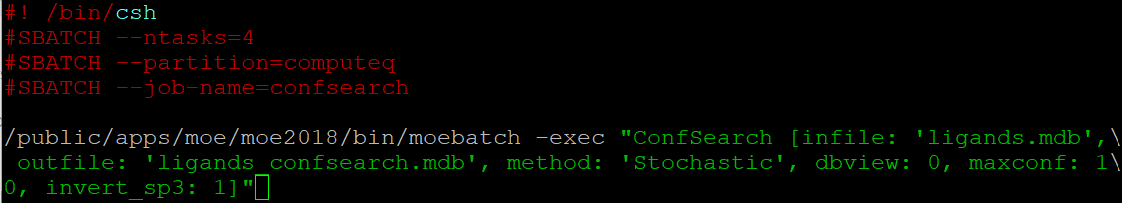
db\_ReadColumn [‘ligands.mdb’, ‘name’]



* As you can see, entering this command reads all entries in the ‘name’ field within the database. Note that this is just one example of the many commands that can be performed in the MOE SVL command line.

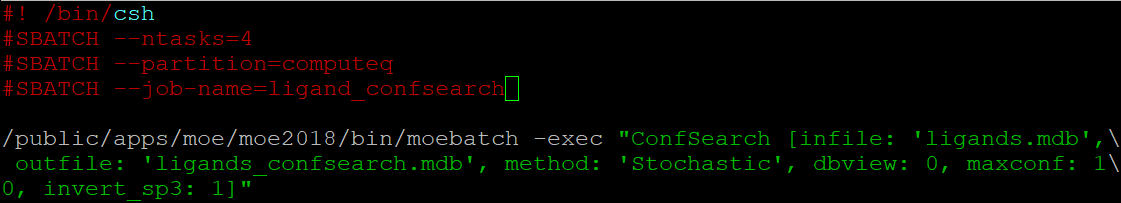
**Setting up batch jobs:**

Occasionally, one may desire to run computationally intensive/time-consuming processes such as conformational searches via the HPC as a means of saving time. To do this, .sh files are used to start jobs on the HPC. A sample job file is included in the conf\_search folder:



Things to note:

* Batch job files can be edited using emacs. In order to edit this job file, use the command emacs conf\_search.sh to edit the file.
* The job name can be changed by editing the 4th line of the .sh file. If I wanted my job to be named “ligand\_confsearch”, I would edit the line’s text appropriately:



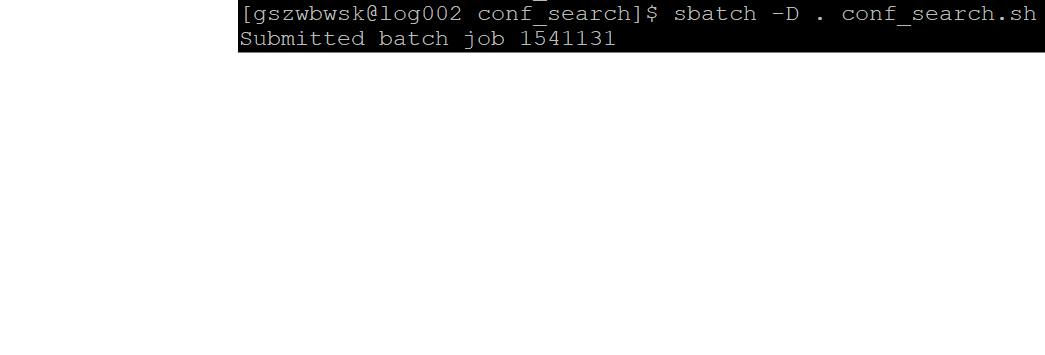
* To save any edits, hold Ctrl and press the X key followed by the S key. To close the editor, hold the Ctrl key and press the X key followed by the C key.

Based on the command given to the HPC in this job file, moebatch will be opened with /public/apps/moe/moe2018/bin/moebatch. The -exec argument instructs moebatch to execute a function, which in this case is the ConfSearch function used to perform conformational searches via command line in MOE.

* ConfSearch works without the -load argument since it is a globally defined function in MOE’s source code. If you are looking to run a custom SVL script, see the “Running SVL scripts” section.
* The options in brackets after the ConfSearch function are function-specific and are often found in the MOE SVL Function Index found in the MOE help file. These options will need to be changed based on what files are being used as input, what the desired output is, and what options are necessary. Help content can be accessed in MOE by clicking the ‘Help’ button located along the top of a MOE window.
* Commands following -exec must always be surrounded by double quotes, as this allows moebatch to read the text as a string to feed into the moebatch command line.

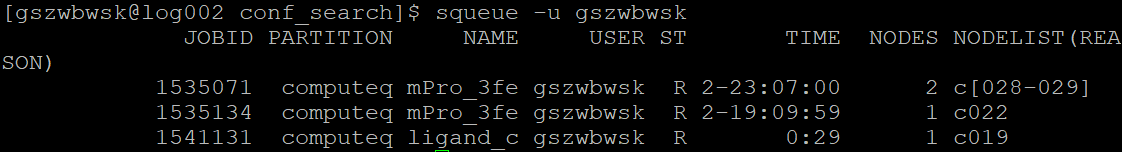
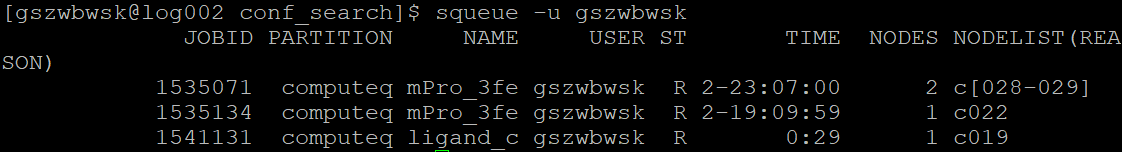
Once the .sh file is configure correctly, use the following command to run the conformational search:

sbatch -D . conf\_search.sh



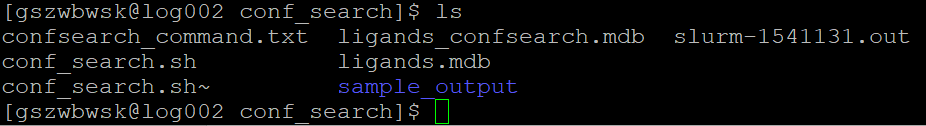
* The -D . portion of the command instructs the job to output in the current directory.

Once this command is entered, the job should start to run automatically. Job status can be checked using the command squeue -u <username>, where <username> is your HPC username.



* The R under the ST column indicates that the job is running. Once the job disappears from the squeue list (after multiple checks using the squeue -u <username> command), the job is complete.

Using the ls command when the job is done should show that multiple files have been created. Our output file (ligands\_confsearch.mdb) contains the conformational search output, and the slurm-xxxxxxx.out file can contain useful information if any output is returned while the job is being run.



**Running SVL scripts:**

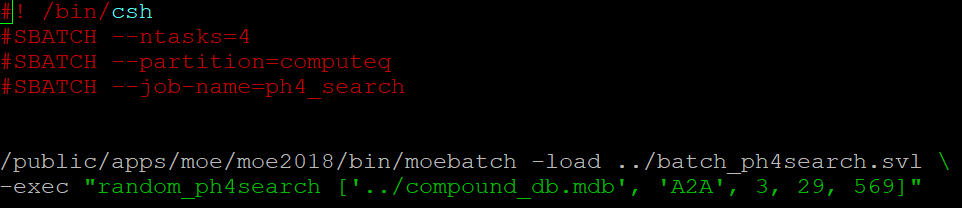
Occasionally, running custom SVL scripts requires a great amount of time and computing power. Fortunately, moebatch can load SVL scripts and run them on the HPC in a more efficient manner. To get started, set your current working directory to the scripts folder located within the moebatch\_tutorial folder:



This folder contains a subfolder called ph4\_search as well as an SVL script I’ve written that will search a compound database using all of the pharmacophore files located in a directory. Navigate to the ph4\_search directory (cd ph4\_search/).

This folder contains the batch.sh job file as well as the pharmacophore files the script acts upon. Open the the batch.sh file using the following command:

emacs batch.sh



Things to note:

* The job name can be changed by editing the third line, as described in the previous section.

Based on the command given to the HPC in this job file, moebatch will be opened with /public/apps/moe/moe2018/bin/moebatch. The -load argument instructs moebatch to first load the custom SVL script, which in this case is the batch\_ph4search.svl file. The text ../ precedes the filename since it is located in the parent folder scripts. After the -load argument, the -exec argument is used to run the function defined in the loaded SVL script.

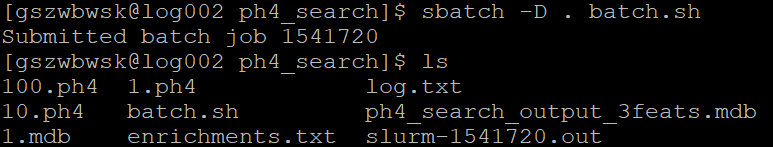
* A backslash (\) *must* be used after the -load section of the command. Without this, moebatch will load the script but fail to read any of the files present in the directory.
* Once again, the line following the -exec argument needs to be surrounded by double quotes.
* No need to worry about the changing any of the variables called for by the random\_ph4search function since this function is purely serving as an example.
  + One thing to note: if the script you are attempting to use calls for a string as an argument (ex. “Receptor”, “Beta 2”), your script will have to be changed to accept tokens as input (ex. ‘Receptor’, ‘Beta 2’) and then convert them to string format using the MOE function string.

Close emacs by holding the Ctrl key and then pressing the X key followed by the C key. Once closed, submit the job using the following command:

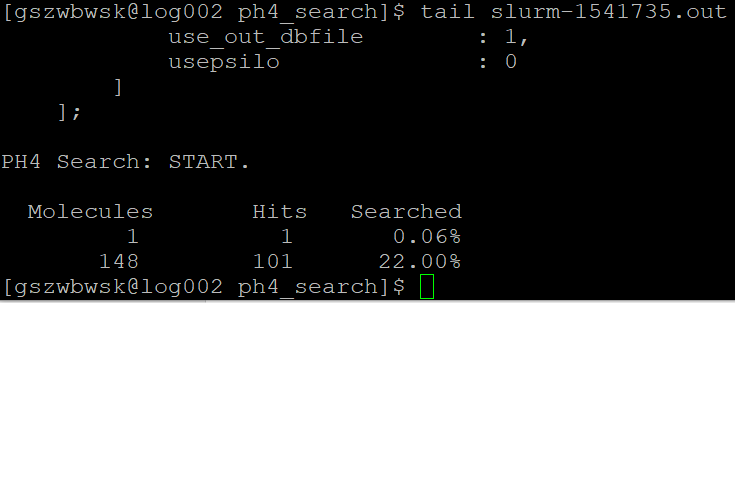
sbatch -D . conf\_search.sh

* The status of the job can once again be checked using the squeue -u <username> command.

Since SVL scripts are typically used for longer jobs, repeatedly checking the job with the squeue command is not as useful. For longer jobs, the slurm-xxxxxx.out files provide much more information. First, use the ls command to see the .out file that has been created:



As you can see, the file slurm-1541720.out has been created. The function running via the job that was just submitted writes a lot of information to the moebatch command line and will provide useful for checking our job status. View the slurm file using the tail command to confirm that the job is running/view how the job is progressing:



Since information is being written to the moebatch command line, we can estimate where the job is at and see how it is progressing. Your job will be done if the squeue command no longer lists the job and/or if the script you are using writes a line in the slurm file saying that it is finished.