# Parallel Matlab on HPC Cluster (Applied to Bresch Segmentation)

You will need to configure Matlab on the cluster. This will need to be done once. The process is based on <https://hpcc.usc.edu/support/documentation/parallel-matlab/> but I have added more material on top of it.

Login to the cluster with X forwarding using option -X in the terminal:

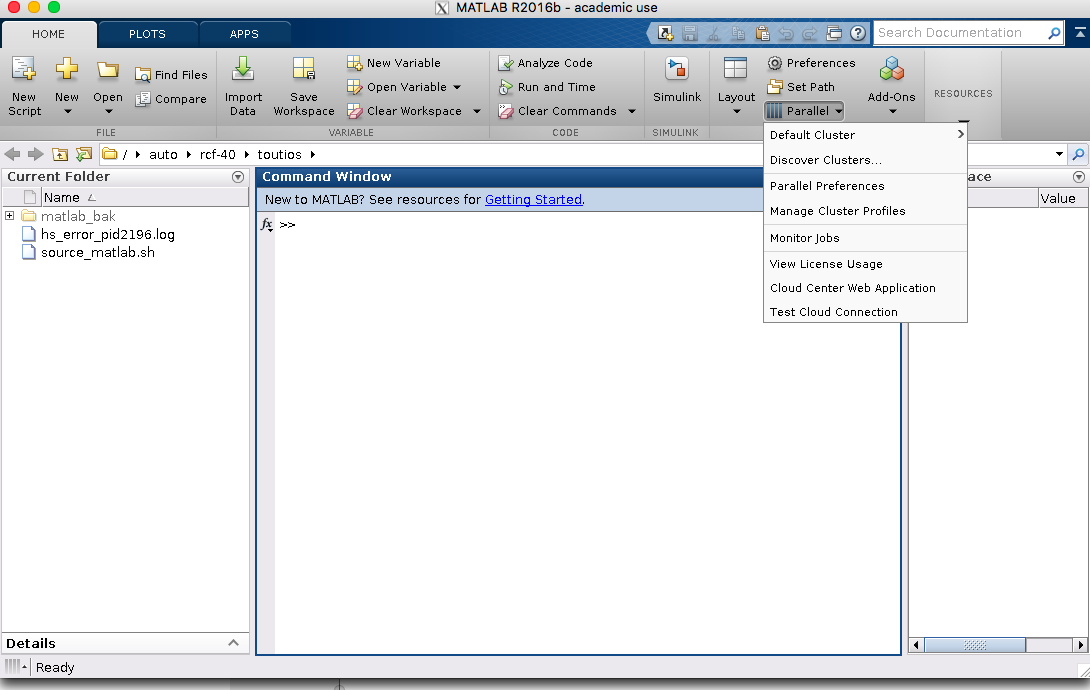
**ssh -X hpc-login2.usc.edu**

Open Matlab:

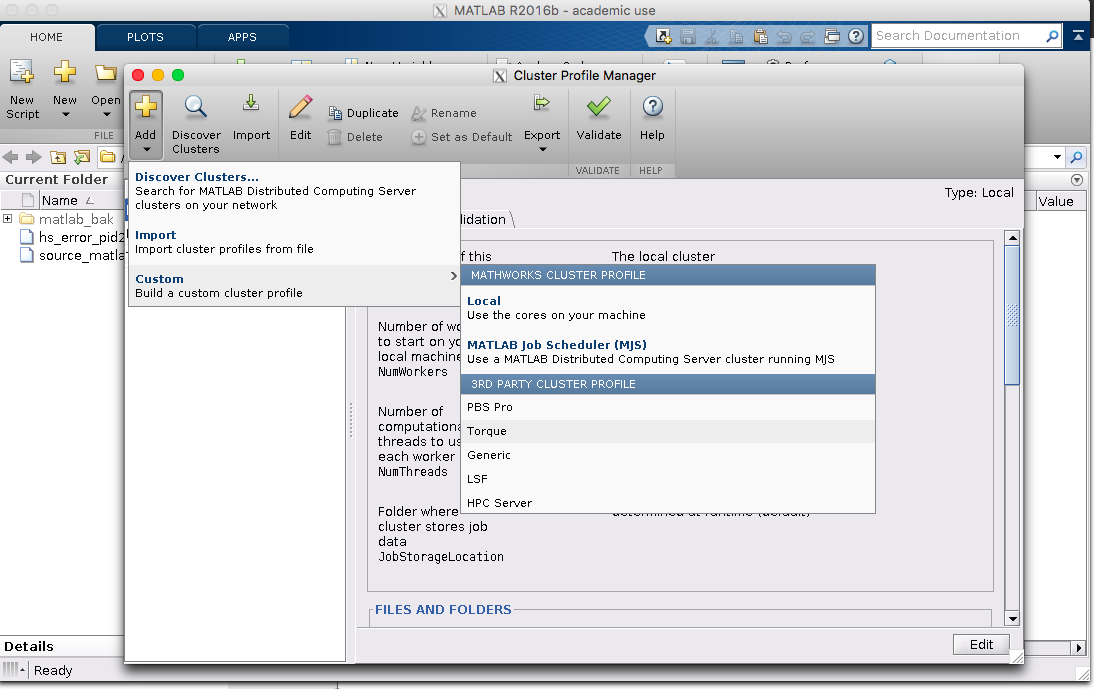
**source /usr/usc/matlab/default/setup.sh; matlab**

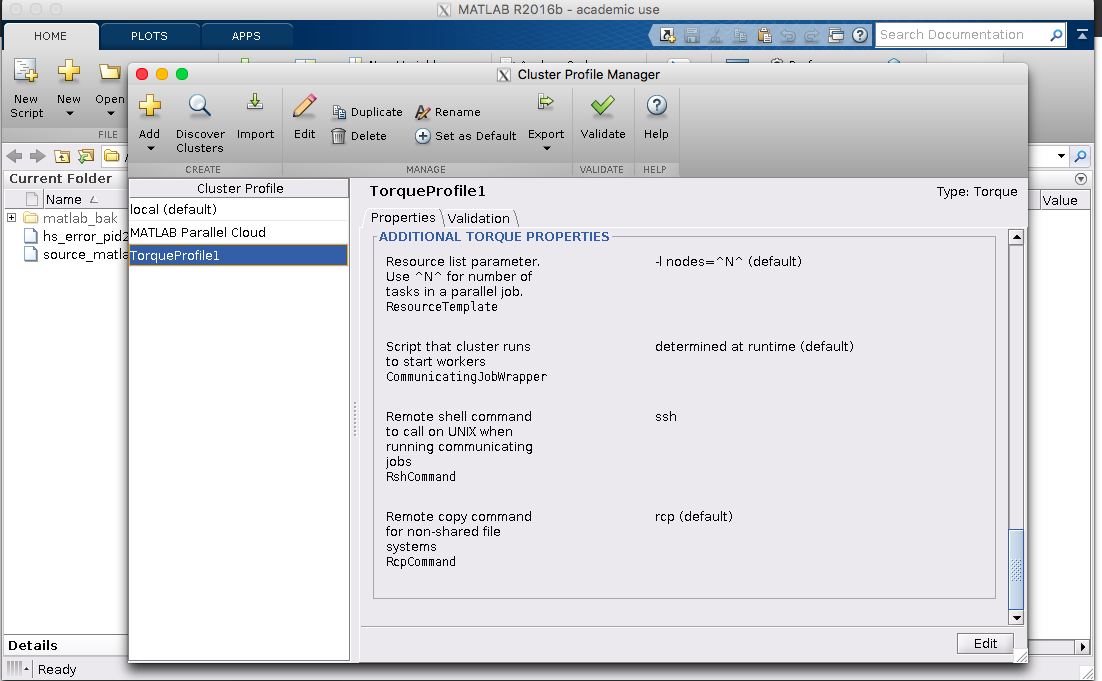
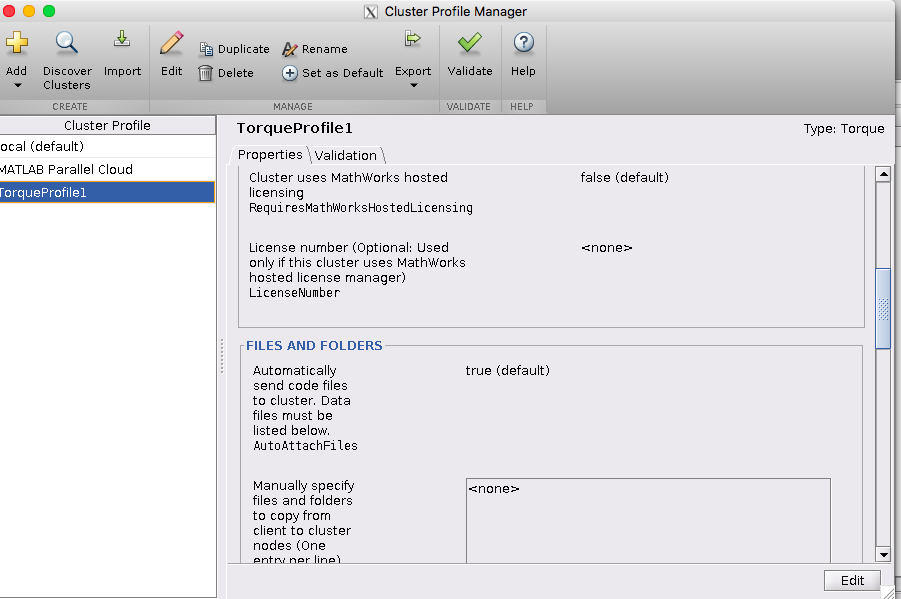
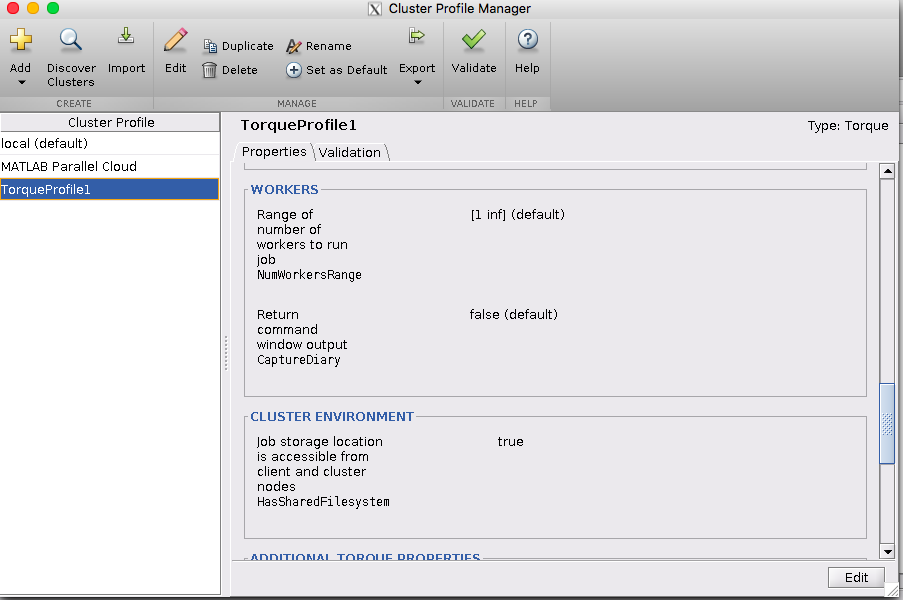
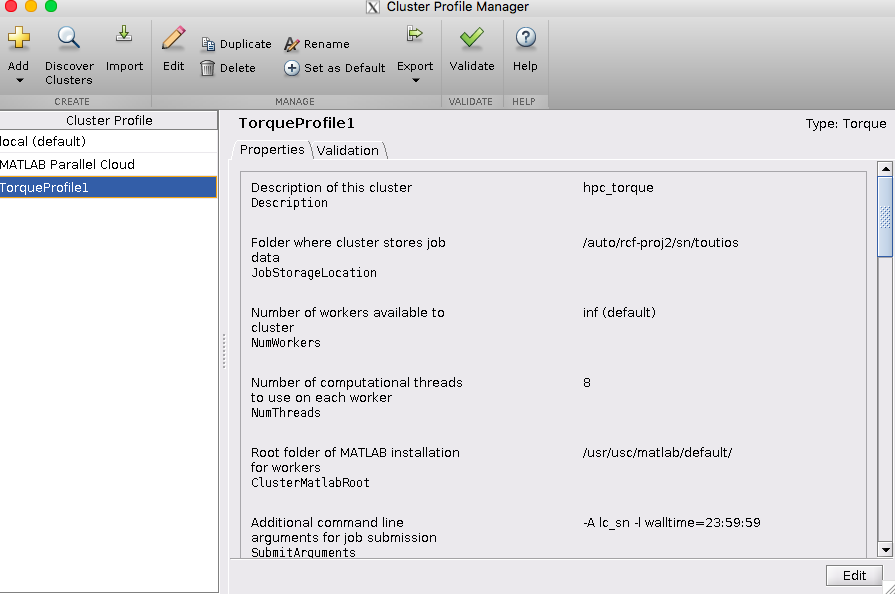
X forwarding allows Matlab to open the full GUI, rather than the terminal interface. The full GUI is required to configure Matlab. The GUI may respond slowly to mouse clicks and keyboard input, so be patient.

Click “Parallel” as shown, and select “Manage Cluster profiles”.



In the window that opens do Add > Custom > Torque



You will have a new profile at the left column called something like “TorqueProfile1” (make a note of the name). You should select it and edit its properties. Click “Edit” at the bottom-right and fill the values as I show in the following screens (replace “toutios” by your username). 

When you are done, close the window and quit Matlab. The exit the terminal.

If the name of the profile was not “TorqueProfile1” you should go to line 28 of cluster\_contour\_tracker.m and replace that name.

Now you can run the script “wrap\_pbs\_batch.m” as before. You will get a similar folder structure as before. The difference is that instead of the shell script “<script>.sh” you will have the Matlab script “<script>.m”. This is the script which executes to start the experiment on the cluster. Upload the folder to the cluster.

Now login to the cluster without X forwarding.

**ssh hpc-login2.usc.edu**

You want to run the “<script>.m”. This is how I recommend doing it:

1. Start a screen session.

**screen**

1. In the blank screen that opens start matlab (this will now be command line, since we don’t have the –X) again with

**source /usr/usc/matlab/default/setup.sh; matlab**

1. Navigate to the script folder and execute it normally. As soon as it starts running you can press Ctrl+a and then d to “detach” the screen. This allows you to exit the terminal and return later to check on the status of the segmentation. Check back in an hour or so. You should start getting “<file name>\_track.mat” files at the usual location.

At any time, you can check the status of the jobs like before and see what is going on using the command:

**qstat -u <username>**

In case you want to return to the Matlab terminal, run t:

**screen –r**

If you are running more than one screen, you also have to specify the session identifier as the argument to the command. You detach as before with Ctrl+a then d. You can google “screen cheat sheet”, it may be interesting.

NOTE: The code uses now 64 cores. This can be changed in line 64 of “cluster\_generate\_scripts.m”. You can have more cores and it theoretically should run faster, but you will have to wait more time in the queue (I tried 128 and did not notice faster overall results). This is a value you can experiment with.