Running a Job on the Cluster

Seth and Jordan, August 27

jhoffmann@fas.harvard.edu

Setup

Step 1: Log In and Ensure that Fiji is Setup

Log in to Odyssey. Type ls and see if Fiji is there. If not, run:

```
wget http://jenkins.imagej.net/job/Stable-Fiji/lastSuccessfulBuild/artifact/fiji-linux64.tar.gz
tar -zxvf fiji-linux64.tar.gz
```

Listing 1: Download Fiji

Now, when you type ls you should see a folder containing Fiji.app. If you cd Fiji.app to get inside the directory you should see a file ImageJ-linux64. Type in pwd. Remember this information. This is the file path to your copy of Fiji.

Step 2: Get the Code

Next, we want to pull a copy of the most recent version of the code from github. Ensure that you are in your home directory by typing cd ~. Next, we want to make a directory that will contain the code. For consistency, consider: mkdir Odyssey. Next, cd into whatever folder you made and then run:

```
git clone https://github.com/JordanHoffmann/seth_and_jordan.git
```

Listing 2: Setup Github Repository

Now, you should see many files. You can always update the version of the code here with the command:

```
git pull origin master
```

Listing 3: Setup Github Repository

Okay, now we have all of the code as well as Fiji. Time to jam.

Running A Job

Okay, now we want to run our first job. To be safe, let us copied the required files to a new directory. Type:

```
cd ~
mkdir ax_ex_4
cd ax_ex_4
cp ../Odyssey/*sh ./
cp ../Odyssey/*py ./
```

Listing 4: Copy

For this task, I think that we just need to Python scripts, the main file is shown below.

```
PATH = '/n/regal/rycroft_lab/jordan/full_ax_ex_4'
iterations = str(15)
Max_T = 300

def submit(time):
        t=str(time)
        return "#!/bin/bash \n#SBATCH -J im_"+t+"\n#SBATCH -N 1\n#SBATCH -n 25\n#SBATCH -t 3-00:00\n#SBATCH -p general\n#SBATCH --mem=100000\n#SBATCH -o out_"+t+".out\n#SBATCH -e err_"+t+".err\nexport DISPLAY=:"+t+"\nXvfb $DISPLAY -auth /dev/null &\n/n/home11/jhoffmann/Fiji/Fiji.app/ImageJ-linux64 --memory=100000m -macro ./time_"+t+".ijm"
def do_tp(time):
        t = str(time)
        return 'run("Fuse/Deconvolve Dataset", "browse='+PATH+'/dataset.xml select_xml='+PATH+'/dataset.xml process_angle=[All angles] process_channel=[Single channel (Select from List)] process_illumination=[All illuminations] process_timepoint=[Single Timepoint (Select from List)] processing_channel=[channel 1] processing_timepoint=[Timepoint '+t+'] type_of_image_fusion=[Multi-view deconvolution] bounding_box=[Define manually] fused_image=[Save as TIFF stack] minimal_x=130 minimal_y=30 minimal_z=-65 maximal_x=780 maximal_y=1860 maximal_z=600 imglib2_container=[CellImg (large images)] imglib2_container_ffts=ArrayImg save_memory type_of_iteration=[Efficient Bayesian - Optimization I (fast, precise)] image_weights=[Virtual weights (less memory, slower)]
    osem_acceleration=[1 (balanced)] number_of_iterations='+iterations+'
    use_tikhonov_regularization tikhonov_parameter=0.0060 compute=[Entire image at
    once] compute_on=[CPU (Java)] psf_estimation=[Provide file with PSF]
    psf_display=[Do not show PSFs] output_file_directory='+PATH+'/decon_15/
    use_same_psf_for_all_angles/illuminations browse=['+PATH+'/psf.tif]
    transform_psfs psf_file=['+PATH+'/psf.tif]");'
11
12 if __name__ == '__main__':
13         for TIME in xrange(1,Max_T+1):
14                 text_file = open("time_"+str(TIME)+".ijm", "w")
15                 string = do_tp(TIME)
16                 text_file.write(string)
17                 text_file.close()
18                 text_file2 = open("submit_"+str(TIME), "w")
19                 string2 = submit(TIME)
20                 text_file2.write(string2)
21                 text_file2.close()
```

Listing 5: SETUP_1.py

We will need to change Line 7 to have the path to your copy of Fiji. Right now this script is a legacy code that is tried and true. Hopefully soon, it is part of a large submit script. Note that I request 25 processors and 100 GB of RAM. We also are only submitting to general. Next, run the command:

```
python Setup_1.py
```

Listing 6: Run Setup

This company might take about 5 seconds and should generate 600 different files. Now we just need to do the submission. There is a file called to_do_list.py that figures out what files still need to be done. Perhaps for the sake of this, the simplest thing to do is to type:

```
python to_do_list.py > RUN.sh
sh RUN.sh
```

Listing 7: Run Setup

Now you should JOBIDs get printed to the screen. You might need to hit one final enter to submit the last job. Now, you can type:

```
squeue -u donoughe
```

Listing 8: Check Status

This list takes some time to populate, but eventually you should see all the jobs there. At some point, they should start switching from `PD` to `R`.