RUNNING NAPARI_LATTICE WORKFLOWS ON HPC!

Setupcd

1. **Login to slurm_login node via rap**
   a. **If this is your first time, type:**
   ```
   module load anaconda3
   conda init
   ```
   b. **FYI Microscopy is found in:**
   /stornext/Img/data/prkfs1/m/Microscopy
   c. **Type:**
   ```
   cd /stornext/Img/data/prkfs1/m/Microscopy/Lachie/MILTON_STUFF/
   ```
   d. **Create an Environment using napari_milton.yml:**
   ```
   conda env create –f napari_milton.yml
   ```
   e. **Unfortunately the current release of napari_lattice is broken, you’ll need to install from the latest update:**
   ```
   conda activate napari_milton
   pip install --upgrade --force-reinstall git+https://github.com/bioimageanalysiscoreWEHI/napari_lattice.git
   ```

Optional Convenience: Create a conda env for the jupyter notebooks so we have a bit more flexibility

```
conda deactivate

conda create –n on_demand python=3.9

conda activate on_demand

conda install ipykernel

python –m ipykernel install --user --on_demand --display-name “Python (on_demand)”

pip install read_roi
```

Done. You can quit out of milton ssh session now

Open on-demand jupyter session

Go to ondemand.hpc.wehi.edu.au

Select Jupyter Notebook

Under “Additional Modules” type Xvfb

Change Runtime hours to as long as you think you’ll need (not to run the whole pipeline, just to get the job(s) submitted.

Click Launch and wait until it’s spun up, then click Connect to Jupyter

Find the “Submit to Milton.ipyb” notebook that I have provided and launch it

Change to on_demand Kernel if you defined one – this is only necessary for some of the new/upcoming features

JUPYTER NOTEBOOK. The only commands you really need are:

ctrl + enter --> run selected cell

shift + enter --> run selected cell and advance to next cell

Arrow keys to navigate

SUBMIT_TO_MILTON.ipyb

1) **Setup file paths for config and submission script**
    a. Config and submission script will be overwritten if they exist
2) **Setup the config file paths:**
    a. Input file path
    b. Output path
    c. Roi_file
        i. Number of rois (==number of jobs)
    d. Workflow file
3) **Select the pipeline parameters**
    a. Processing: “deskew”,”crop”,”workflow”, “workflow_crop”
    b. Deconvolution: “cuda_gpu” , if using
        i. Number of iterations
        ii. psf_paths (I give 4, I’m not sure that’s necessary)
    c. Time_range, channel_range
    d. Output_file_type
4) **Setup Milton Parameters**
    a. Job_name is an arbitrary label
    b. Logfiles_path is where milton output logs go, vast is good if you have access because these don’t generally need to be backed up

   c. Wall_time is the time it takes to run ONE job – after that the process will just stop even if it's not running. Balance.

5) **Create config and submission file**
6) **Run the !sbatch command**
7) **Wait a few seconds**
8) **Run the !squeue command (with your username)**

9) **Wait for the email!**

If you want to submit another job, restart the notebook kernel to clear all the variables and go through the process again. This is the most basic version of this script and I intend to make a more powerful one but I just want to see how we get on now.