

Using Relion

RELION is installed on our HPC clusters. We make use of [singularity](#) containers to make the program portable across different HPC architectures and also to facilitate compilation and redistribution.

Data Location

We ask that you keep your RELION data in the following directories:

- Proposal based: if you are working on data from an official Proposal with a designated proposal code (eg CS10, C002, etc), then you should ask for a 'group' directory under `/sdf/group/cryoem/g/<PROPOSAL>`.
- Team project: if you intend to share your data with other collaborators (either internal or external), please ask for a directory under the 'group' space.

Getting Started

Two ways to access the graphical user interface

- Using 'X Server' on your local (laptop/desktop) machine.
 - You can install [XQuartz](#) on mac. You should already be running one on Linux.
- Using FastX in a browser
 - <https://fastx3.slac.stanford.edu:3300>
 - Click "I Accept"
 - Log in with your SLAC **unix** credentials
 - Launch a new session (+ button) and go to the Command tab in the session configuration pop-up
 - Enter the following in the form fields:
Command:

```
xterm -ls -e "ssh sdf-login"
```

Window Mode (either works):

Single – Single window mode in which the session is contained to one window (Set a window size. Default 1024 x 768. Changing this doesn't seem to improve resolution)

Multiple – Multiple window mode in which the session can have multiple independent windows

Run As User:

Leave blank/default (will log-in as the user you used to log in to FastX)

Name (Optional):

Name of the session

- Click "Launch" to start the FastX session. A new tab/window will pop up (depending on your browser settings) with an open xterm window prompting you to log in to the SDF bastion hosts.
- Log in with your **SDF** credentials (this should be your SLAC Windows/Active Directory login).

To bring up the RELION GUI:

```
# Following the instructions above will land you in your 'SDF Home' directory. DO NOT use this space for your
# CryoEM work; there is insufficient disk space (quota: 25GB) and it is also not very performant. You should
# use the project directory defined in the "Data Location" section above, e.g.:
cd /sdf/group/cryoem/g/<PROPOSAL>

# if you're starting a new RELION 'project':
mkdir my_new_project
cd my_new_project

# to see what modules are installed:
module avail

# load the modules you plan to use within RELION e.g. (module versions in this example are the most current as
of
# 2022-09-28):
module load motioncor2/1.4.4
module load ctffind/4.1.13
module load relion/ver4.0

# start the relion GUI
relion
```

Using the Batch Queues



You can choose to run RELION jobs either 'locally' (ie on the login node), or submit to the batch queues. As the login node is a shared interactive resource, it is important not to run long or intensive jobs locally but to submit them to dedicated batch queue nodes where your job will run uninterrupted (and often with much higher performance)

The default settings provided by the RELION container should work.

Checking on Batch Jobs

As you click submit job to the queue on the RELION gui, in the terminal window that you launched RELION from, it should output the jobid.

You can check the queue status using

- `module load slurm`
- `squeue`

And you can see more details about your specific job via:

- `scontrol show job <jobid>`

Using Scratch

Under the Compute tab in RELION, there are options to pre-stage the data into either RAM or (say) local disk to speed up your job by increasing the performance of access to the data.

under

Copy particles to scratch directory, set it to `/lscratch/${USER}/${SLURM_JOB_ID}`

Further information at <https://hpc.nih.gov/training/handouts/relion.pdf>

GPU vs. CPU

RELION GPU vs. CPU

| Task | Implementation | GPU | CPU |
|----------------------|---|-----|-----|
| Motion correction | RELION | - | X |
| Motion correction | MotionCor | X | - |
| CTF estimation | CtfFind | - | X |
| CTF estimation | Gctf | X | - |
| Extraction | | - | X |
| Autopicking | Reference free (Laplacian of Gaussian) | - | X |
| Autopicking | Reference based | X | X |
| 2D classification | With alignments ('Perform image alignment = Yes') | X | X |
| 2D classification | Without alignments ('Perform image alignment = No') | - | X |
| 3D initial model | | X | X |
| 3D classification | With alignments ('Perform image alignment = Yes') | X | X |
| 3D classification | Without alignments ('Perform image alignment = No') | - | X |
| 3D refinement | | X | X |
| 3D multibody | | X | X |
| CTF refinement | | - | X |
| Bayesian polishing | | - | X |
| Mask creation | | - | X |
| Particle subtraction | | - | X |
| Post processing | | - | X |
| Local resolution | RELION | - | X |
| Local resolution | ResMap | X | X |

Migrating from LSF

If you have an existing project that was using the SLAC LSF job submission, you will need to update the fields under the job 'Running' tab:

| | | |
|---------------------------|--|--|
| | | |
| Queue Name | --partition=cryoem | |
| Queue Submit Command | sbatch | |
| Walltime | 1-00:00:00 | |
| Memory per CPU | 3800m | |
| GPUs (clear to remove) | --gpus 4 | |
| Additional Directives (1) | | |
| Additional Directives (2) | -N 1 | |
| Script | /sdf/group/cryoem/sw/images/relion/slurm-batch-submission.script | |
| | | |

Other notes: