
Introduce of CryoEM Resources on BioHPC

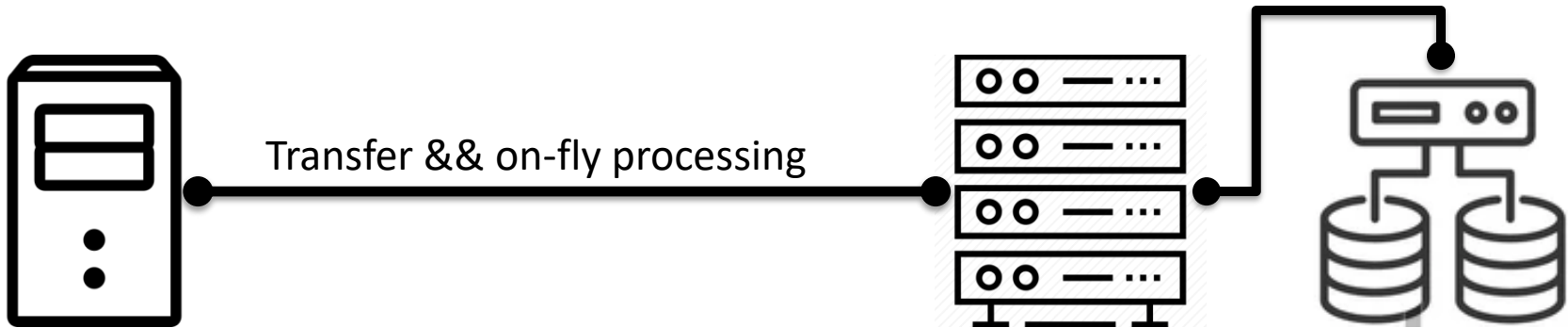
[web] portal.biohpc.swmed.edu

[email] biohpc-help@utsouthwestern.edu

Overview CryoEM resources on BioHPC

- **Raw image transfer and on-fly processing, Data Storage**
- **Web Visualization, Interactive job, SLURM job**
- **BioHPC workstation**
- **GPU partitions:**
 - >100 nodes Nvidia - Tesla K20/K40, P4, P40, P100, V100s, 4V100, A100, more will come.
- **Tools:**
 - RELION, cryoSPARC, EMAN2, MotionCorr, Coot, Cryolo, Chimera, IMOD, PEET, SBGrid, et.al.

Data transfer and Storage



Camera PCs

BioHPC cluster and storage

`:\DoseFraction\user\Images-Disc`
`:\DoseFraction\user\KEEP`

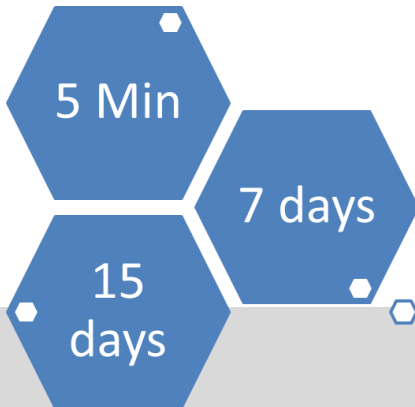
`/project/cryoem/cryoem_transfer/\krios/\user`

...

...

...

Important: Move to your own space on BioHPC or local server after your cryoEM microscope session.



Access data storage on BioHPC

As a BioHPC user, you will have several storage options:

BioHPC Standard Storage Limits



<code>/home2/<username></code>	50 GB per user
<code>/work/<department>/<username></code>	5 TB per user
<code>/project/<organization>/<group>/<username></code>	5 TB per group
<code>/archive/<organization>/<group>/<username></code>	5 TB per group



BioHPC File Exchange (accessible to outside of UTSW)

<https://cloud.biohpc.swmed.edu> 50 GB / user



BioHPC Lamella Cloud Storage (UTSW-only)

<https://lamella.biohpc.swmed.edu> 100 GB / user
(FTP, SAMBA or WebDAV*)

Project and Archive can be increased on PI request with Dept. Chair approval.
Archive usage is multiplied by 2/3 (as to encourage use of archive).
Data on home2 counts thrice and on work counts twice because of backup.

Guides about BioHPC storage:

- 1, BioHPC portal Guides/FAQ pages: [BioHPC Guides \(swmed.edu\)](#)
- 2, BioHPC training slides: [BioHPC Storage Overview \(swmed.edu\)](#)

Setting up Lamella to access project and work space

The screenshot shows the Nextcloud 'External storages' configuration page. The left sidebar contains navigation options: Personal info, Security, Activity, External storages (highlighted with a red arrow and the number 2), Mobile & desktop, Accessibility, Sharing, and Privacy. The main content area is titled 'External storages' and includes a descriptive paragraph. Below this is a table of configured external storages. The table has columns for Folder name, External storage, Authentication, and Configuration. There are four rows of storage configurations. The third row, for the 'work' folder, has a red arrow (3) pointing to the checkmark in the Configuration column. The top right corner of the interface has a user profile icon with a dropdown menu containing 'Settings', 'About', and 'Log out', with a red arrow (1) pointing to the menu. A red arrow (2) also points to the 'External storages' menu item in the sidebar.

Folder name	External storage	Authentication	Configuration
home	BioHPC/Lysosome	Log-in credentials, save in database	✓
Cloud	Cloud	Log-in credentials, save in database	✓
work	BioHPC/Lysosome	Log-in credentials, save in database	<input type="text" value="/work"/> <input type="text" value="/cellbiology/s179389"/> ... ✓
project	BioHPC/Lysosome	Log-in credentials, save in database	<input type="text" value="/project"/> <input type="text" value="/biohpcadmin/s179389"/> ... ✓
worknew	BioHPC/Lysosome	Log-in credentials, save in database	<input type="text" value="work"/> <input type="text" value="/bioinformatics/s179389"/> ... ✓

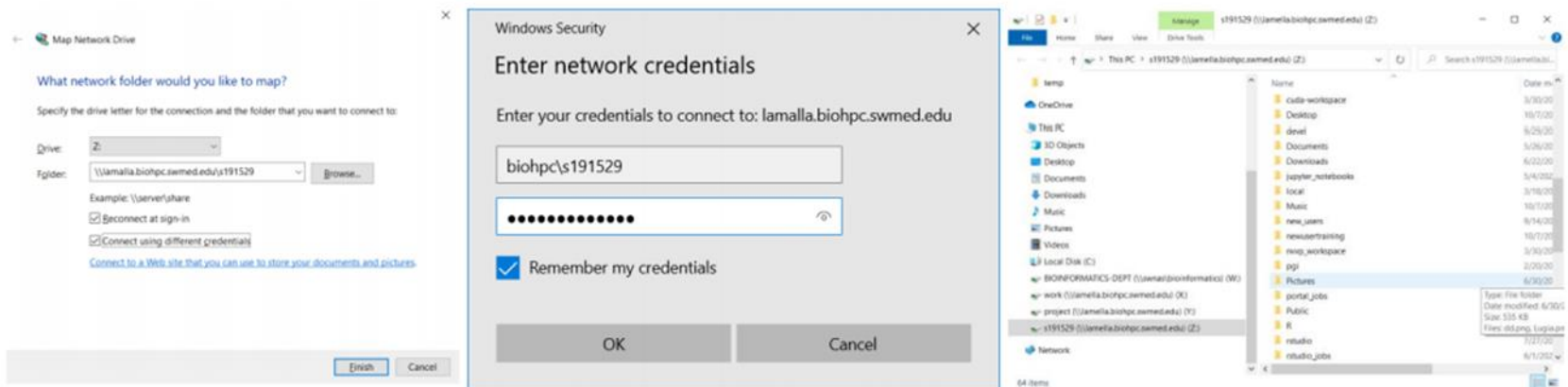
Setting up Lamella to access project and work

Computer -> Map Network Drive

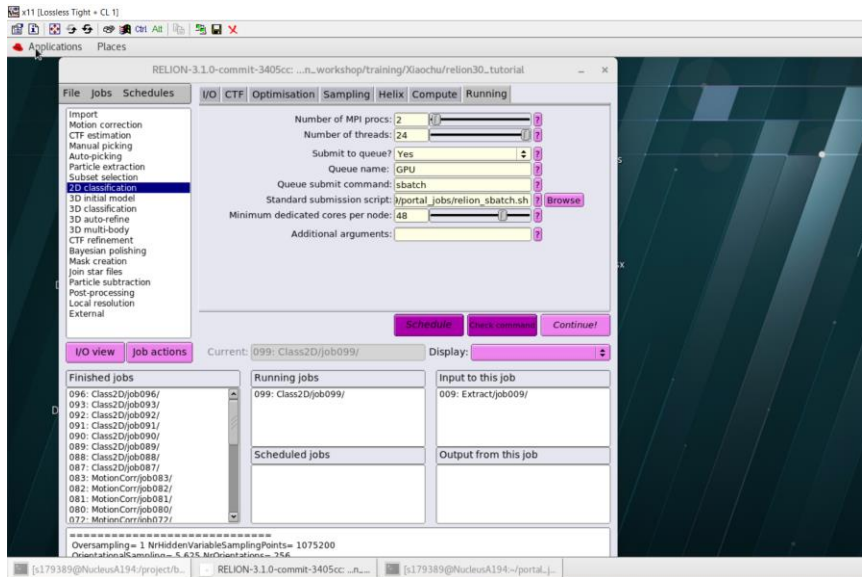
Folder is: \\lamella.biohpc.swmed.edu\username (home dir)
\\lamella.biohpc.swmed.edu\project
\\lamella.biohpc.swmed.edu\work

Check 'Connect using different credentials'

Enter your BioHPC username and password when prompted.



Web Visualization and Slurm Job



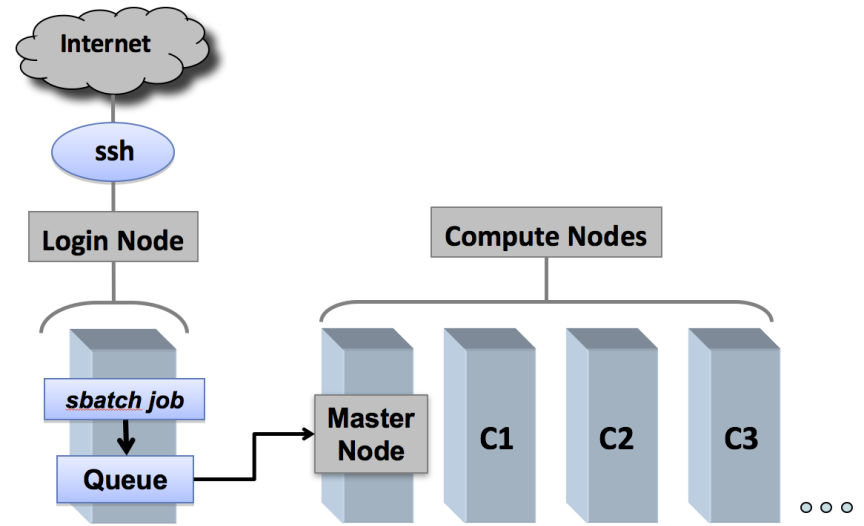
```
[s179389@Nucleus005 ~]$ cat ~/portal_jobs/relion_sbbatch.sh
#!/bin/bash

#SBATCH --job-name RelionJob
#SBATCH -p XXXqueueXXX           # partition (queue)
#SBATCH -N XXXnodesXXX
##SBATCH --mem 250000            # Memory Requirement (MB)
##SBATCH -t 1-2:0:00
#SBATCH -o XXXoutfileXXX
#SBATCH -e XXXerrfileXXX
#SBATCH --gres=gpu:1
#SBATCH --mail-type ALL
#SBATCH --mail-user xiaochu.lou@utsouthwestern.edu

module load shared relion/gcc/openmpi/cuda101/3.1

mpirun -n XXXmpinodesXXX XXXcommandXXX

[s179389@Nucleus005 ~]$
```



GPU nodes

GPU Partition	Number of CPU/Node	Memory Per Node	Number of GPU/Node	GPU Memory	Number of nodes
GPU	32	256GB	1 K20/K40	6GB/12GB	8
GPU _{p4}	72	384GB	1 P4	8GB	16
GPU _{p40}	72	384GB	1 P40	24GB	16
GPU _{p100}	56	256GB	2 P100	16GB	12
GPU _{v100s}	72	384GB	1 V100S	32GB	32
GPU _{4v100}	72	384GB	4 V100S	32GB	12
GPU _{A100}	72	1.5TB	1 A100	40GB	16
GPU _{4A100}	72	1.5TB	4 A100	80GB	10

Check node availability

```
[s179389@Nucleus005 ~]$ sinfo -p GPUp4
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
GPUp4    up      infinite    5   alloc NucleusC[002,012-013,016-017]
GPUp4    up      infinite   11   idle  NucleusC[003-011,014-015]
```


Modules

```
[s179389@Nucleus005 ~]$ module avail

----- /cm/local/modulefiles -----
cluster-tools/7.3 dot          gcc/6.1.0          module-git         null              shared
cmd                          freeipmi/1.5.2    ipmitool/1.8.17   module-info       openldap          use.own

----- /cm/shared/modulefiles -----
abra2/2.18                    intel/mkl/32/2017/6.256
acml/gcc/64/5.3.1             intel/mkl/64/2017/6.256
acml/gcc/fma4/5.3.1           intel/mkl/64/2017/current
acml/gcc/mp/64/5.3.1          intel/mkl/mic/2017/6.256
acml/gcc/mp/fma4/5.3.1        intel/mpi/32/2017/6.256
acml/gcc-int64/64/5.3.1       intel/mpi/64/2017/6.256
acml/gcc-int64/fma4/5.3.1     intel/mpi/mic/2017/6.256
acml/gcc-int64/mp/64/5.3.1    intel-cluster-checker/2.1.2
acml/gcc-int64/mp/fma4/5.3.1  intel-cluster-runtime/ia32/3.8
afni/20.0.04                  intel-cluster-runtime/intel64/3.8
afni/v17.2.17                 intel-cluster-runtime/mic/3.8
afni/v18.3.03                 intel-tbb-oss/ia32/2017_20170807oss
almost/2.1                     intel-tbb-oss/intel64/2017_20170807oss
amber/12                        io_lib/1.13.3

[s179389@Nucleus005 ~]$ module avail relion

----- /cm/shared/modulefiles -----
relion/gcc/1.2                 relion/gcc/openmpi/cuda80/2.0-beta  relion/intel/openmpi/1.4
relion/gcc/1.3                 relion/gcc/openmpi/cuda80/2.1.0     relion/intel/openmpi/cuda101/3.1-beta
relion/gcc/mvapich2/1.4        relion/gcc/openmpi/cuda91/2.1.0     relion/intel/openmpi/cuda80/2.0-beta
relion/gcc/openmpi/1.4         relion/intel/1.3                     relion/intel/openmpi/cuda91/3.0.1
relion/gcc/openmpi/cuda101/3.1 relion/intel/mvapich2/1.4

[s179389@Nucleus005 ~]$ module avail cryosparc

----- /cm/shared/modulefiles -----
cryosparc/2.15.0-6-singularity cryosparc/3.1.0-singularity

[s179389@Nucleus005 ~]$ module avail imod

----- /cm/shared/modulefiles -----
imod/4.7.5                     imod/4.8.50-beta          imod/cuda65/4.8.50-beta  imod/cuda80/4.11.0    imod/cuda80/4.9.3
imod/4.8.38                    imod/4.9.3                imod/cuda80/4.10.32     imod/cuda80/4.9.12

[s179389@Nucleus005 ~]$
```

Run RELION and submit RELION batch job for GPU intensive jobs

```
s179389@NucleusA039:~/project/biohpcadmin/s179389/Relion_workshop/training/Xiaochu/relion30_tutorial
[s179389@NucleusA039 relion30_tutorial]$ module load relion/gcc/openmpi/cuda101/3.1
[s179389@NucleusA039 relion30_tutorial]$ relion &
[1] 87305
[s179389@NucleusA039 relion30_tutorial]$ WARNING: cannot find angpix in the defined
WARNING: cannot find do_set_angpix in the defined joboptions. Ignoring it ...
```

- Module load relion;
- Start a relion GUI;
- Setting parameters: Note for the **Number of MPI procs**: checking with RELION Manual and see following slides for examples;
- Submit job to queue via the default submit script

The screenshot displays the RELION-3.1.0 GUI interface. At the top, there are two dropdown menus: "Use GPU acceleration?" set to "Yes" and "Which GPUs to use?" set to "0:1:0:1". Below these, the "Compute" tab is selected and highlighted with a red box. The "Compute" tab contains several configuration fields: "Number of MPI procs:" (6), "Number of threads:" (16), "Submit to queue?" (Yes), "Queue name:" (GPU100), "Queue submit command:" (sbatch), "Number of Nodes:" (2), "Number of tasks per node:" (3), "Number of GPUs per node:" (2), "Standard submission script:" (mpi/cuda101/3.1/bin/sbatch.sh), and "Minimum dedicated cores per node:" (56). There are also "Additional arguments:" and "Additional GPUs" fields. At the bottom of the "Compute" tab, there are three buttons: "Schedule", "Check command", and "Run!". Below the "Compute" tab, there are sections for "Finished jobs", "Running jobs", "Input to this job", "Scheduled jobs", and "Output from this job". The "Finished jobs" section lists several job IDs and names, such as "133: Class2D/job133/". At the bottom of the GUI, there are two lines of text: "stdout will go here; double-click this window to open stdout in a separate window" and "stderr will go here; double-click this window to open stderr in a separate window".

Examples of submitting RELION job to multiple nodes

Motion correction

1 GPU_p100 nodes,
2 GPU cards on each nodes

Use RELION's own implementation? No ?
MOTIONCOR2 executable: MotionCor2 ? Browse
Which GPUs to use: 0:1 ?
Other MOTIONCOR2 arguments ?

File Jobs Schedules

- Import
- Motion correction**
- CTF estimation
- Manual picking
- Auto-picking
- Particle extraction
- Subset selection
- 2D classification
- 3D initial model
- 3D classification
- 3D auto-refine
- 3D multi-body
- CTF refinement
- Bayesian polishing
- Mask creation
- Join star files
- Particle subtraction
- Post-processing
- Local resolution
- External

I/O **Motion** Running

Number of MPI procs: 2 ?
Number of threads: 16 ?
Submit to queue? Yes ?
Queue name: GPU_p100 ?
Queue submit command: sbatch ?
Number of Nodes 1 ?
Number of tasks per node 2 ?
Number of GPUs per node 2 ?
Standard submission script: ../../../../cm/shared/apps/ ? Browse
Minimum dedicated cores per node: 48 ?
Additional arguments: ?

Schedule Check command Run!

Examples of submitting RELION job to multiple nodes

2D classification

1 GPU40 nodes,
1 GPU cards on each nodes

Use GPU acceleration? Yes [?] [v]
Which GPUs to use: 0 [?]

File Jobs Schedules I/O CTF Optimisation Sampling Helix **Compute** Running

Import
Motion correction
CTF estimation
Manual picking
Auto-picking
Particle extraction
Subset selection
2D classification
3D initial model
3D classification
3D auto-refine
3D multi-body
CTF refinement
Bayesian polishing
Mask creation
Join star files
Particle subtraction
Post-processing
Local resolution
External

I/O view Job actions

Number of MPI procs: 3 [?] [v]
Number of threads: 16 [?] [v]
Submit to queue? Yes [?] [v]
Queue name: GPU40 [?] [v]
Queue submit command: sbatch [?] [v]
Number of Nodes 1 [?] [v]
Number of tasks per node 3 [?] [v]
Number of GPUs per node 1 [?] [v]
Standard submission script: /usr/local/cuda-10.1/bin/nvcc.sh [?] [v] Browse
Minimum dedicated cores per node: 48 [?] [v]
Additional arguments: [?] [v]

Schedule Check command Run!

Current: Give_alias_here Display: [v]

Examples of submitting RELION job to multiple nodes

2D classification

4 GPU40 nodes,

1 GPU cards on each nodes

The screenshot shows the RELION software interface with the 'Compute' tab selected. The 'Use GPU acceleration?' field is set to 'Yes' and 'Which GPUs to use:' is set to '0:0:0:0'. The 'Number of MPI procs:' is 12, 'Number of threads:' is 16, 'Submit to queue?' is 'Yes', 'Queue name:' is 'GPU40', 'Queue submit command:' is 'sbatch', 'Number of Nodes' is 4, 'Number of tasks per node' is 3, 'Number of GPUs per node' is 1, 'Standard submission script:' is '/mpi/cuda101/3.1/bin/sbatch.sh', and 'Minimum dedicated cores per node:' is 48. The 'Additional arguments:' field is empty. The 'Schedule', 'Check command', and 'Run!' buttons are visible at the bottom. The 'Current:' field shows 'Give_alias_here' and the 'Display:' field is empty.

Use GPU acceleration? Yes ?
Which GPUs to use: 0:0:0:0 ?

File Jobs Schedules I/O CTF Optimisation Sampling Helix **Compute** Running

Import
Motion correction
CTF estimation
Manual picking
Auto-picking
Particle extraction
Subset selection
2D classification
3D initial model
3D classification
3D auto-refine
3D multi-body
CTF refinement
Bayesian polishing
Mask creation
Join star files
Particle subtraction
Post-processing
Local resolution
External

I/O view Job actions

Number of MPI procs: 12 ?
Number of threads: 16 ?
Submit to queue? Yes ?
Queue name: GPU40 ?
Queue submit command: sbatch ?
Number of Nodes 4 ?
Number of tasks per node 3 ?
Number of GPUs per node 1 ?
Standard submission script: /mpi/cuda101/3.1/bin/sbatch.sh ? Browse
Minimum dedicated cores per node: 48 ?
Additional arguments: ?

Schedule Check command Run!

Current: Give_alias_here Display:

Examples of submitting RELION job to multiple nodes

2D classification

1 GPU_p100 nodes,
2 GPU cards on each nodes

Use GPU acceleration? ?
Which GPUs to use: ?

File Jobs Schedules

I/O CTF Optimisation Sampling Helix **Compute** Running

- Import
- Motion correction
- CTF estimation
- Manual picking
- Auto-picking
- Particle extraction
- Subset selection
- 2D classification**
- 3D initial model
- 3D classification
- 3D auto-refine
- 3D multi-body
- CTF refinement
- Bayesian polishing
- Mask creation
- Join star files
- Particle subtraction
- Post-processing
- Local resolution
- External

Number of MPI procs: ?

Number of threads: ?

Submit to queue? ?

Queue name: ?

Queue submit command: ?

Number of Nodes ?

Number of tasks per node ?

Number of GPUs per node ?

Standard submission script: ? [Browse](#)

Minimum dedicated cores per node: ?

Additional arguments: ?

[Schedule](#) [Check command](#) [Run!](#)

I/O view Job actions

Current: Display:

Examples of submitting RELION job to multiple nodes

2D classification

2 GPUp100 nodes,
2 GPU cards on each nodes

Use GPU acceleration? ?
Which GPUs to use: ?

File Jobs Schedules I/O CTF Optimisation Sampling Helix **Compute** Running

Import
Motion correction
CTF estimation
Manual picking
Auto-picking
Particle extraction
Subset selection
2D classification
3D initial model
3D classification
3D auto-refine
3D multi-body
CTF refinement
Bayesian polishing
Mask creation
Join star files
Particle subtraction
Post-processing
Local resolution
External

I/O view Job actions

Number of MPI procs: ?
Number of threads: ?
Submit to queue? ?
Queue name: ?
Queue submit command: ?
Number of Nodes ?
Number of tasks per node ?
Number of GPUs per node ?
Standard submission script: ?
Minimum dedicated cores per node: ?
Additional arguments: ?

Current: Display:

Examples of submitting RELION job to multiple nodes

2D classification

3 GPUp100 nodes,
2 GPU cards on each nodes

Use GPU acceleration? ?
Which GPUs to use: ?

The screenshot shows the RELION software interface with the 'Compute' tab selected. The left sidebar contains a menu with '2D classification' highlighted. The main configuration area includes the following settings:

- Number of MPI procs: 9
- Number of threads: 16
- Submit to queue?: Yes
- Queue name: GPUp100
- Queue submit command: sbatch
- Number of Nodes: 3
- Number of tasks per node: 3
- Number of GPUs per node: 2
- Standard submission script: /usr/local/cuda-10.1/bin/nvcc.sh
- Minimum dedicated cores per node: 48
- Additional arguments: (empty)

Buttons at the bottom include 'Schedule', 'Check command', and 'Run!'. The 'Compute' tab is highlighted in red, and a red box highlights the GPU acceleration settings at the top.

Examples of submitting RELION job to multiple nodes

2D classification

4 GPUp100 nodes,
2 GPU cards on each nodes

The screenshot displays the RELION software interface with the 'Compute' tab selected. A red box highlights the 'Use GPU acceleration?' dropdown set to 'Yes' and the 'Which GPUs to use:' field containing '0:1:0:1:0:1:0:1'. A red arrow points from this box to the 'Number of GPUs per node' field, which is set to '2'. Other configuration fields include: 'Number of MPI procs: 12', 'Number of threads: 16', 'Submit to queue? Yes', 'Queue name: GPUp100', 'Queue submit command: sbatch', 'Number of Nodes: 4', 'Number of tasks per node: 3', 'Standard submission script: /usr/local/cuda-10.1/bin/nvcc.sh', and 'Minimum dedicated cores per node: 48'. The 'Additional arguments:' field is empty. At the bottom, there are buttons for 'Schedule', 'Check command', and 'Run!', along with a 'Current:' field containing 'Give_alias_here' and a 'Display:' dropdown.

Examples of submitting RELION job to multiple nodes

RELION 3.1 Manual

Number of MPI procs 3

(Note that *when using the EM-algorithm*, 2D classification, 3D classification, 3D initial model and 3D auto-refine use one MPI process as a master, which does not do any calculations itself, but sends jobs to the other MPI processors. Therefore, we often run the EM-algorithm using a single worker MPI process on each of the available GPUs, so we specify 3 here to include the master and one workers on each of the two GPUs.)

The general rule for 2D classification, 3D classification, 3D initial model, and 3D auto-refine would be:

```
IF Num_of_GPU_per_Node == 1:
```

```
    Num_of_MPI = 3 * Num_of_Nodes;
```

```
IF Num_of_GPU_per_Node >1:
```

```
    Num_of_MPI = (Num_of_GPU_per_node + 1) * Num_of_Nodes;
```

```
Num_of_threads = minimum_num_of_cores_per_Node * Num_of_Nodes / Num_of_MPI ;
```

```
Num_of_tasks_per_Node = Num_of_MPI / Num_of_Nodes;
```

For other steps, eg, motion correction:

```
Num_of_MPI = Num_of_GPU_per_Node * Num_of_Nodes;
```

Standard RELION submit script

```
$ cat /cm/shared/apps/relion/gcc/openmpi/cuda101/3.1/bin/sbatch.sh  
#!/bin/bash
```

```
#SBATCH --job-name RELIONJob  
#SBATCH -p XXXqueueXXX      # partition (queue)  
#SBATCH --nodes=XXXextra1XXX  
#SBATCH --ntasks-per-node=XXXextra2XXX  
##SBATCH --mem 250000      # Memory Requirement (MB)  
##SBATCH -t 2-2:0:00  
#SBATCH -o XXXoutfileXXX  
#SBATCH -e XXXerrfileXXX  
#SBATCH --gres=gpu:XXXextra3XXX
```

```
module load shared relion/gcc/openmpi/cuda101/3.1
```

```
mpiexec -n XXXmpinodesXXX XXXcommandXXX
```

Create and use user defined submit script

The image shows a terminal window and a graphical interface for job submission. The terminal window displays the following commands and output:

```
s179389@NucleusA039:~/project/biohpcadmin/s179389/Relion_workshop/training/Xiaochu/relion30_tutorial
[s179389@NucleusA039 relion30_tutorial]$ module load relion/gcc/openmpi/cuda101/3.1
[s179389@NucleusA039 relion30_tutorial]$ relion &
[1] 87305
[s179389@NucleusA039 relion30_tutorial]$ WARNING: cannot find angpix in the defined job
WARNING: cannot find do_set_angpix in the defined joboptions. Ignoring it ...
```

The graphical interface shows a list of job options on the left, including '2D classification', '3D initial model', '3D classification', '3D auto-refine', '3D multi-body', 'CTF refinement', 'Bayesian polishing', 'Mask creation', 'Join star files', 'Particle subtraction', 'Post-processing', 'Local resolution', and 'External'. The '2D classification' option is selected. The configuration panel on the right shows the following settings:

- Number of MPI procs: 6
- Number of threads: 16
- Submit to queue?: Yes
- Queue name: GPU100
- Queue submit command: sbatch
- Number of Nodes: 2

A file selection dialog is open, showing a list of files and folders. The file 'relion_sbatch.sh' is selected. The filename field contains '/home2/s179389/portal_jobs/relion_sbatch.sh'. A red arrow points to the 'Browse' button in the background interface.

Customize your RELION submit script

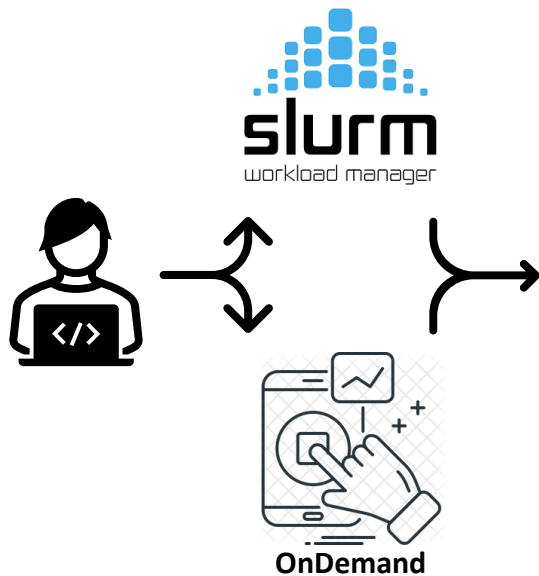
```
$ cat ~/portal_jobs/relion_sbatch.sh  
#!/bin/bash
```

```
#SBATCH --job-name RELIONJob  
#SBATCH -p XXXqueueXXX      # partition (queue)  
#SBATCH --nodes=XXXextra1XXX  
#SBATCH --ntasks-per-node=XXXextra2XXX  
##SBATCH --mem 250000      # Memory Requirement (MB)  
##SBATCH -t 1-2:0:00  
#SBATCH -o XXXoutfileXXX  
#SBATCH -e XXXerrfileXXX  
#SBATCH --gres=gpu:XXXextra3XXX  
#SBATCH --mail-type ALL  
#SBATCH --mail-user xiaochu.lou@utsouthwestern.edu
```

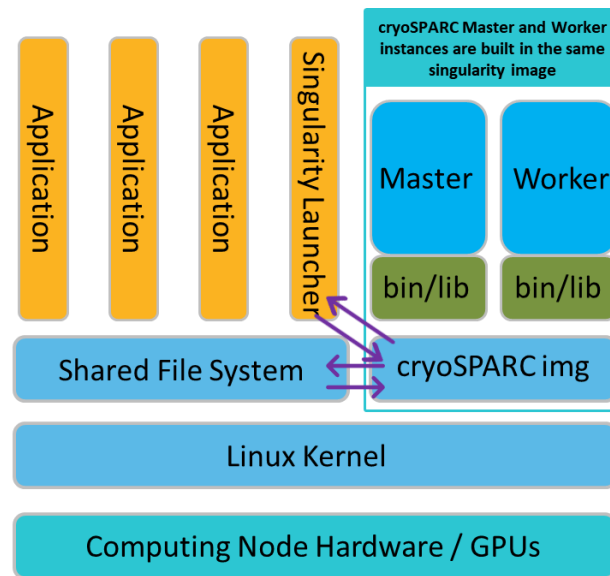
```
module load shared relion/gcc/openmpi/cuda101/3.1  
mpiexec -n XXXmpinodesXXX XXXcommandXXX
```

Containerization of cryoSPARC with Singularity

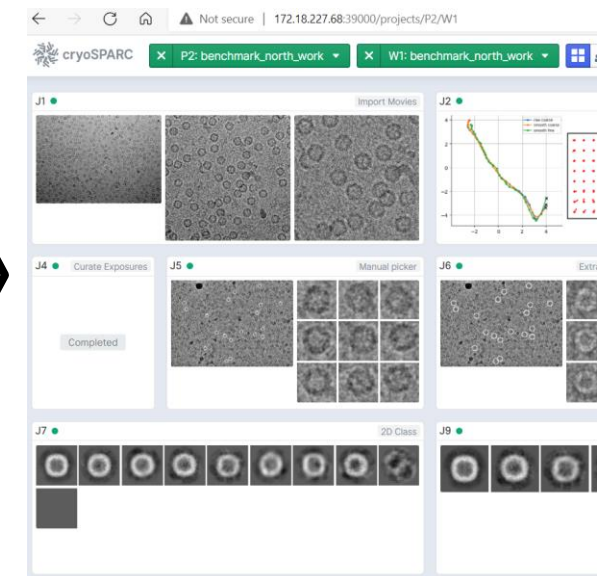
User submit job



Job queue on cluster



Access cryoSPARC via browser



Get the cryoSPARC license

CryoSPARC™ and **cryoSPARC Live™** are available free of charge for **non-profit academic use**. To obtain a License ID for cryoSPARC, go to cryosparc.com/download, fill out the form and submit it.

Get the cryoSPARC™ System

CryoSPARC™ and cryoSPARC Live™ are available free of charge for **non-profit academic use**. Please fill out the form to request a license.

Please allow up to 24 hours for us to respond to your request. Instructions for downloading and installing cryoSPARC will be emailed to you. In the meantime, please feel free to review our extensive documentation available [here](#).

Questions? Please contact us: info@structura.bio.

Commercial/for-profit licensing enquiries can be sent to sales@structura.bio.

I'm an academic user carrying out my own research at a public or private non-profit educational/research institution

Export cryoSPARC license ID to user env

Step 1: Create a hidden file (.cryopwd) at user home2 directory, which is used to save the passwd for the cryoSPARC session. **Note: Please use some random characters. DO NOT use important password of yours (eg, BioHPC account password)**

```
[s179389@Nucleus006 ~]$ cat ~/.cryopwd  
*****
```

Step 2: Add line “**export CRYOSPARC_LICENSE_ID="????????-????-????-????-????????????"**” to export your cryoSPARC license ID to user .bashrc file.

```
[s179389@Nucleus006 ~]$ cat ~/.bashrc  
# .bashrc  
  
# Source global definitions  
if [ -f /etc/bashrc ]; then  
    . /etc/bashrc  
fi  
  
# User specific aliases and functions  
module load slurm shared  
export CRYOSPARC_LICENSE_ID="????????-????-????-????-????????????"
```


OnDemand cryoSPARC and cryoSPARC Live

BioHPC OnDemand - CryoSPARC



CryoSPARC is a state of the art scientific software platform for cryo-electron microscopy (cryo-EM) used in research and drug discovery pipelines. CryoSPARC combines powerful innovations in 3D reconstruction algorithms with specially designed software to provide a streamlined end-to-end single particle cryo-EM workflow. Rapidly solve high-resolution structures of biologically important targets, with advanced tools for membrane proteins, heterogeneous samples, and flexible molecules. Process 3D refinements in minutes on GPU.

Note that each user has unique License to run CryoSparc. If you don't have a license ID you can get one by reaching UTSW CryoEM management or CryoSparc web interface at <https://guide.cryosparc.com/licensing>. The licence number must be added to the .bashrc file in the format given below:
export CRYOSPARC_LICENSE_ID="????????-????-????-????-????????????"

Jobs shown 'PENDING' are waiting for a free node on the cluster. You can connect to jobs once they are 'RUNNING'. Reload the page to refresh the status of your CryoSPARC session jobs.

Jobs run for a maximum of 20 hours. To end a job earlier than this please click the cancel link.

You have no current CryoSPARC sessions

Launch a new CryoSPARC session

Guides about cryoSPARC usage:

1, BioHPC portal Guides page: [CryoEM resource on BioHPC \(swmed.edu\)](https://swmed.edu)

OnDemand cryoSPARC and cryoSPARC Live

portal.biohpc.swmed.edu/terminal/ondemand_cryosparc/

Language

Jobs shown 'PENDING' are waiting for a free node on the cluster. You can connect to jobs once they are 'RUNNING'. Reload the page to refresh the status of your CryoSPARC session jobs.

Jobs run for a maximum of 20 hours. To end a job earlier than this please click the cancel link.

Current CryoSPARC Sessions

Type	Job ID	Status	Times	Connect to Session	Cancel
CryoSparc3GPU	2782713	SLURM: RUNNING Provisioning: NONE	Submitted: June 15, 2021, 11:48 p.m. Started: June 15, 2021, 11:48 p.m. Ends: June 16, 2021, 7:48 p.m.	Address: http://vnc.biohpc.swmed.edu:20475 Password: *****	

Launch a new CryoSPARC session

Note that a session may take time to start if there are no nodes currently free in the cluster. Jobs run for a maximum of 20 hours.

Job type*

CryoSparc3GPU - CryoSparc3 Development Environment on the Web

Your session will start immediately, nodes are available.

Launch Job

cryoSPARC and cryoSPARC Live

The screenshot displays the cryoSPARC Live web interface. The top navigation bar shows the session is paused for 'S1: biohpc_demo'. The left sidebar contains a 'Details' menu with options like Configuration, Overview, and Refinement. The main workspace is divided into several panels:

- Configuration:** Includes 'Compute Resources' with fields for Preprocessing Lane, Reconstruction Lane, and Auxiliary Lane, and checkboxes for 'Use SSD'.
- Parameters:** A 'Microscope/Camera Parameter' section with input fields for Raw pixel size (A), Accelerating voltage (kV), Spherical aberration (mm), and Total exposure dose (e/A²).
- 1 Exposure Group:** Settings for 'Exposure Group 1 (0 found)', including 'Directory to watch' and 'Minimum File Size'.
- Start Checklist:** A light blue box listing required information for starting a session, such as 'Compute Resources: Preprocessing Lane' and 'Raw pixel size (A)'.
- Workflow Grid:** A grid of job steps (J1-J10) including 'Import Movies', 'Patch motion (M)', 'Patch CTF (M)', 'Curate Exposures', 'Manual picker', 'Extract From Micrographs', '2D Class', 'Select 2D', and 'Template picker'. Each step shows a preview of the data or results.
- Job Builder:** A sidebar on the right showing 'P2 → W1 DETAILS' with a list of jobs (J1-J13) and a 'Delete' action.

Submit slurm job to start cryoSPARC and cryoSPARC Live

```
$ cat ~/portal_jobs/cryosparc/cryosparc_sbatch_v100s.sh
#!/bin/bash
#SBATCH --job-name="Cryosparc3"
#SBATCH --partition=GPUv100s
#SBATCH --nodes=1
#SBATCH --gres=gpu:1 # Number of GPUs(per node)
#SBATCH --ntasks=1
#SBATCH --time=2-02:00:00
#SBATCH --output="logs.cryosprac3.%j.%N.txt"
#SBATCH --error=errors.cryosparc3.%j.%N.txt

module load cryosparc/3.3.2-singularity
export no_proxy="localhost"
export CUDA_VISIBLE_DEVICES=0
cryosparc start
tail -f ~/cryosparc-v3/run/command_core.log
```

Submit slurm job to start cryoSPARC and cryoSPARC Live

Submit to start the job

```
[s179389@Nucleus005 ~]$ sbatch ~/portal_jobs/cryosparc/cryosparc_sbatch_v100s.sh  
Submitted batch job 2776433
```

Check the cryoSPARC job initialization log

```
[s179389@Nucleus005 ~]$ cat ~/cryosparc-v3/cryosparc.log
```

.....

From other machines on the network, access cryoSPARC at

<http://Nucleus162.cm.cluster:39000>

and access cryoSPARC Live at

<http://Nucleus162.cm.cluster:39006>

.....

Success starting cryosparc master!

.....

Success starting cryosparc worker

Checking Licence validation !

.....

FINISHED : Check ~/cryosparc-v3/run/command_core.log file for detail

◀ Allocated node with cryoSPARC job running.

Connect to cryoSPARC web interface via:

172.18.224.162:39000 (39006: cryoSPARC Live)

The number is from Nucleus162

The other example, if the allocated node is

NucleusC048; Connect to cryoSPARC web interface via:

172.18.227.48:39000 (39006: cryoSPARC Live)

The number is from NucleusC048

Submit slurm job to start cryoSPARC and cryoSPARC Live

Submit to start the job

```
[s179389@Nucleus005 ~]$ sbatch ~/portal_jobs/cryosparc/cryosparc_sbatch_v100s.sh  
Submitted batch job 2776433
```

Cancel the job after finished

```
[s179389@Nucleus005 ~]$ module load cryosparc/3.1.0-singularity
```

```
[s179389@Nucleus005 ~]$ squeue -u s179389
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
2776433	GPU4	Cryospar	s179389	R	5:16	1	NucleusC014

```
[s179389@Nucleus005 ~]$ cryosparc_canceljob 2776433
```

```
user confirmed of job: 2776433 running on node: NucleusC014
```

```
CryoSPARC is running.
```

```
Stopping cryoSPARC
```

```
app: stopped
```

```
command_core: stopped
```

```
command_rtp: stopped
```

```
command_vis: stopped
```

```
liveapp: stopped
```

```
webapp: stopped
```

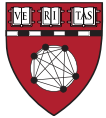
```
database: stopped
```

```
Shut down
```

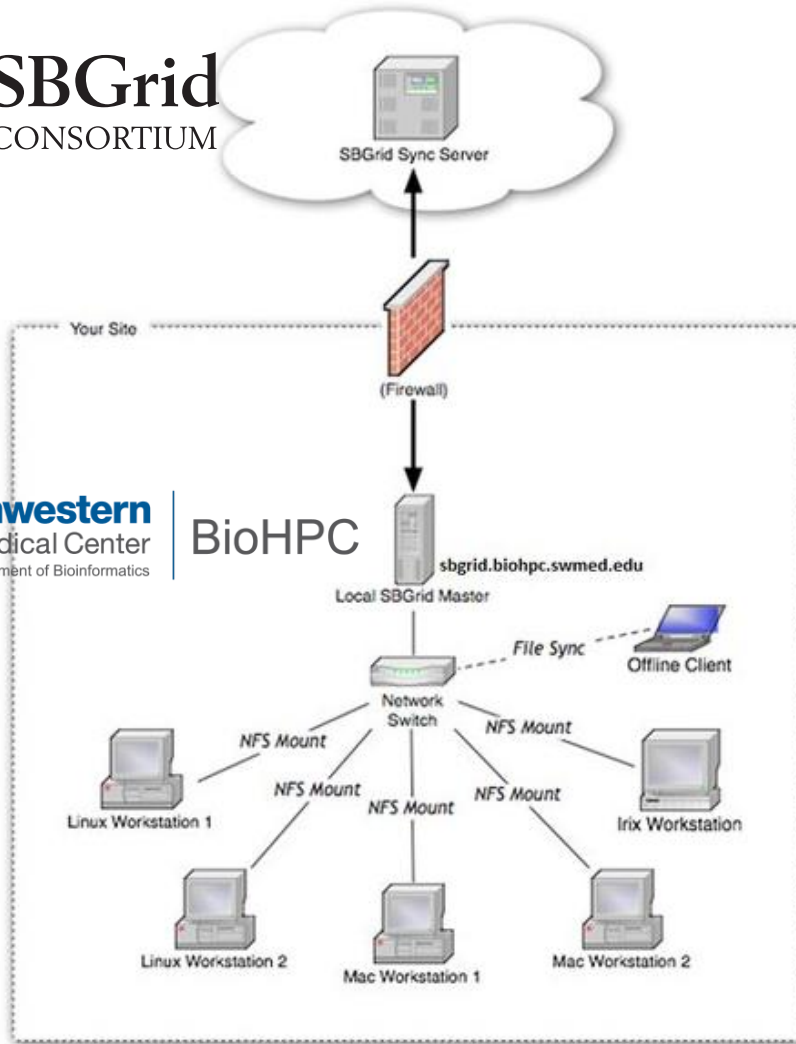
Note: before first time use:

Create a file named `~/.cryopwd` and write your Cryosparc password to the first line without any space.

Initializing and Use SBGrid on BioHPC



SBGrid
CONSORTIUM



UT Southwestern
Medical Center
Lyda Hill Department of Bioinformatics

BioHPC

License / Access

SBGRID is available to SBGRID member labs under the license below:

Your use of the applications contained in the /programs directory constitutes acceptance of the terms of the SBGrid License Agreement included in the file /programs/share/LICENSE. The applications distributed by SBGrid are licensed exclusively to member laboratories of the SBGrid Consortium.

Information about joining SBGRID can be obtained from the SBGRID consortium via: <https://sbgrid.org/join/>

Guides about SBGrid usage:

1, BioHPC portal Guides page: [SBGrid on BioHPC \(swmed.edu\)](#)

Initializing and Use SBGrid on BioHPC

```
[s179389@Nucleus005 ~]$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions
module load slurm/16.05.8 shared

alias data='cd /project/biohpcadmin/s179389'

#sbgrid initializing
. /programs/sbgrid.shrc
```


Initializing and Use SBGrid on BioHPC

```
*****
                Software Support by SBGrid (www.sbgrid.org)
*****
Your use of the applications contained in the /programs directory constitutes
acceptance of the terms of the SBGrid License Agreement included in the file
/programs/share/LICENSE. The applications distributed by SBGrid are licensed
exclusively to member laboratories of the SBGrid Consortium.
    Run sbgrid-accept-license to remove the above message.
*****
SBGrid was developed with support from its members, Harvard Medical School,
HHMI, and NSF. If use of SBGrid compiled software was an important element
in your publication, please include the following reference in your work:

Software used in the project was installed and configured by SBGrid.
cite: eLife 2013;2:e01456, Collaboration gets the most out of software.
*****
SBGrid installation last updated: 2021-03-05 (Update available)
Please submit bug reports and help requests to:      <bugs@sbgrid.org> or
                                                    <http://sbgrid.org/bugs>

    For additional information visit https://sbgrid.org/wiki
*****
                SBGrid Announcements
- There are known issues with MacOS 11.0 "Big Sur" and some SBGrid
  applications. We recommend not upgrading Apple computers to 11.0
  until these issues can be addressed.
  see https://sbgrid.org/wiki/big\_sur for more info.
*****
[s179389@Nucleus006 ~]$ which relion
/programs/x86_64-linux/system/sbgrid_bin/relion
[s179389@Nucleus006 ~]$ module load relion/gcc/openmpi/cuda101/3.1
[s179389@Nucleus006 ~]$ which relion
/cm/shared/apps/relion/gcc/openmpi/cuda101/3.1/bin/relion
```

Use IMOD and PEET on BioHPC

The screenshot shows a Linux desktop environment with a terminal window and the 'Setup Tomogram - Etomo' dialog box open. The terminal window displays the following commands and output:

```
[s179389@NucleusA194 ~]$ module load imod/cuda80/4.9.3 peet/
[s179389@NucleusA194 ~]$ etomo
starting eTomo with log in /home2/s179389/.etomologs/etomo_e
[s179389@NucleusA194 ~]$
```

The 'Setup Tomogram - Etomo' dialog box is titled 'Setup Tomogram - Etomo' and contains the following fields and options:

- Dataset name:** [Empty text field]
- Backup directory:** [Empty text field]
- Templates:**
 - Scope template: None available
 - System template: No selection (2 available)
 - User template: None available
- Data Type:**
 - Axis Type: Single axis, Dual axis
 - Frame Type: Single frame, Montage
- Scan Header:** Pixel size (nm): [Empty text field], Fiducial diameter (nm): [Empty text field], Image rotation (degrees): [Empty text field]
- Parallel Processing, Graphics card processing
- Axis A:**
 - Extract tilt angles from data
 - Specify the starting angle and step (degrees)
 - Starting angle: -60.0, Increment: 1.0
 - Tilt angles in existing rawTlt file
 - Series was bidirectional from [Empty text field] degrees
 - Exclude views: [Empty text field]
 - Focus was adjusted between montage frames
- Axis B:**
 - Extract tilt angles from data
 - Specify the starting angle and step (degrees)
 - Starting angle: -60.0, Increment: 1.0
 - Tilt angles in existing rawTlt file
 - Series was bidirectional from [Empty text field] degrees
 - Exclude views: [Empty text field]
 - Focus was adjusted between montage frames
- Buttons:** Cancel, Use Existing Coms, Create Com Scripts, Advanced
- Status:** No data set loaded

UT Southwestern
Medical Center
Lyda Hill Department of Bioinformatics

BioHPC

CryoSPARC/Relion Workshop (SBL/BioHPC)

Questions? Comments?

Email: biohpc-help@utsouthwestern.edu

Thanks!

UT Southwestern

Medical Center

Lyda Hill Department of Bioinformatics

BioHPC
