
초극저온전자현미경(Cryo-EM) 데이터 분석을 위한 분석 클러스터 및 소프트웨어 도구 활용 보고서

Release 2.0

국가슈퍼컴퓨팅본부 대용량데이터허브센터

Jun 12, 2023

CONTENTS:

1	Change Logs	1
2	GSDC Computing Cluster for TEM Users and Operators	3
2.1	1. Service overview	3
2.2	2. Computing and storage resources	5
2.3	3. Cluster management softwares	6
2.4	4. Data analysis tools	8
2.5	5. Requesting user accounts and accessing GSDC TEM computing cluster	9
2.6	6. Module paths and job submission templates	9
2.6.1	Module paths for data analysis tools	9
2.6.2	Job submission templates	10
2.7	7. Batch queues	10
2.7.1	Checking batch queue names and their status	11
2.7.2	Checking all worker nodes status	11
2.8	fstat.bin : Monitoring the usage of all the worker nodes	17
2.9	dynamotd : Checking storage quota limit and usage ratio	18
3	TEM Computing Cluster Basics	21
3.1	Requesting accounts and accessing TEM cluster	21
3.1.1	For Linux/Mac users	21
3.1.2	For Windows users	21
3.2	Understanding environment modules	23
3.3	Job manager (Torque)	26
3.3.1	Resources manager and job scheduler	26
3.3.2	Directives in Torque job scripts	27
3.3.3	Torque job script examples	28
3.3.4	Job submission	29
3.3.5	Monitoring and managing your jobs	29
4	Batch Queues	31
4.1	Batch queue list	31
5	Relion	33
5.1	Executing Relion GUI tools	33
5.1.1	How to start Relion data analysis tool	33
5.2	PBS Strings used in Relion	36
5.3	Running data analysis jobs using CPU cores	37
5.3.1	Module path	37
5.3.2	Environment variables	37

5.3.3	Standard job submission script (for relion 4.0.0 CPU use)	38
5.4	Running data analysis jobs using GPGPUs	39
5.4.1	Environment variables	39
5.4.2	Standard job submission script (for relion 4.0.0 GPGPU use)	41
5.4.3	Specifying which GPGPUs to use	42
5.5	Examples	43
5.5.1	Motion Correction	43
5.5.2	CTF Estimation	47
5.5.3	2D Classification	51
6	cisTEM	57
6.1	Executing cisTEM	57
6.1.1	How to start cisTEM data analysis tool	57
6.2	Run profiles for job submission	59
6.2.1	Profile templates	59
6.2.2	Adding a new Run Profile	60
6.3	Examples of running cisTEM jobs	61
6.3.1	Importing Movies and images	61
6.3.2	Movie Alignment	62
7	CryoSPARC	65
7.1	Prerequisites	65
7.2	Getting a cryoSPARC instance	66
7.2.1	1. (Admin) Install and setup a cryoSPARC instance	66
7.2.2	2. (User) Verifying installation	67
7.3	Launching CryoSPARC instance	69
7.3.1	For Linux/Mac users	69
7.3.2	For Windows users	69
7.4	Exploring CryoSPARC web apps	71
7.4.1	CryoSPARC login	71
7.4.2	CryoSPARC dashboard	71
7.4.3	CryoSPARC project	72
7.4.4	CryoSPARC cluster(lane)	72
7.5	Tutorial on processing T20S	73
7.6	Trouble shooting	73
7.6.1	1. How to stop or start the cryoSPARC instance?	73
7.6.2	2. How to reset the password of non-admin user?	74
7.6.3	3. Job (or Workflow) failed caused by SSD caching	74
7.6.4	4. Failed to launch! 190	75
7.6.5	5. Binary locations of Gctf, MotionCor2	75
8	Topaz	77
8.1	Topaz Executables	77
8.1.1	Topaz binary executable with CUDA 9.2	78
8.1.2	Topaz binary executable with CUDA 11.2	78
8.2	Using Topaz in CryoSPARC	78
8.3	Using Topaz in Relion (v3.1.0 above)	79
9	FAQs	81
9.1	How to resolve the problems on (re)starting your own cryosparc instance?	81
9.2	How to repair cryosparc database corruption?	83
9.3	How to upgrade (or downgrade) to the specific version of cryosparc softwares?	83

9.3.1	1. Checking for updates	83
9.3.2	2. Before you update: complete or kill running jobs	84
9.3.3	3. Back-up cryosparc databases	85
9.3.4	4. Cryosparc master updates	86
9.3.5	5. Cryosparc worker updates	90
9.3.6	6. Running the newer cryosparc instance	94
10	APPENDIX : Application Form & Materials	95
10.1	GSDC TEM Application Form	95
10.2	Tutorial	95
11	Contacts	97

**CHAPTER
ONE**

CHANGE LOGS

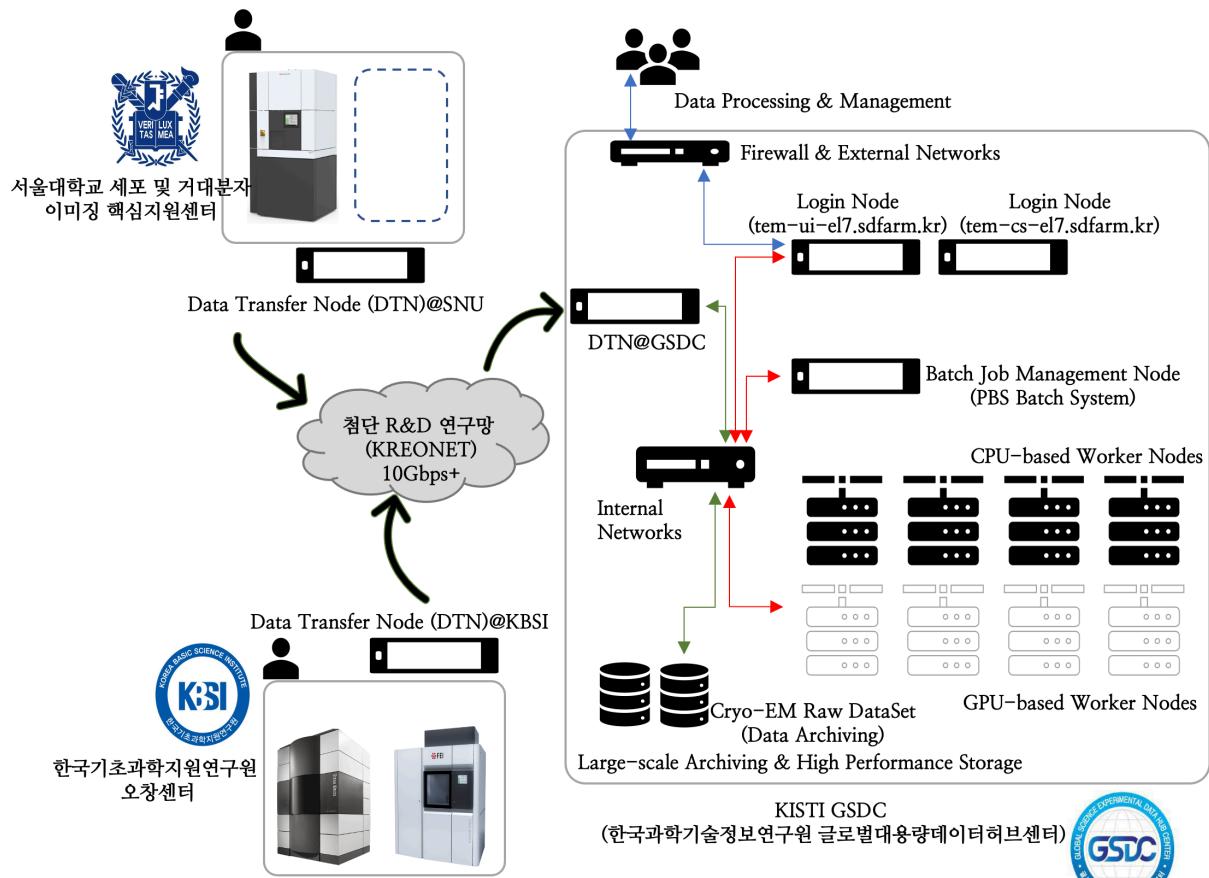
- 2023-06-30 : This document version 2.0 released.
- 2023-05-31 : Started to support Pusan National University (PNU) Cryo-EM Center.
- 2023-05-09 : CryoSPARC updated for the scratch directory per each research group.
- 2023-02-06 : Data migration from old storages (NAS and JBOD) to new high performance storage (based on lustre file system) completed.
- 2022-12-01 : Started to support Seoul National University (SNU) Cryo-EM Center.
- 2022-11-11 : Relion version 4.0.0 support added.
- 2022-11-10 : IBM aspera client tool (EMPIAR data transfer tool) added.
- 2021-11-19 : PyEM module environment added.
- 2021-08-23 : Topaz and Topaz integration (with Relion, CryoSPARC) guides added.
- 2021-07-12 : FAQ list updated.
- 2021-03-23 : Relion, cisTEM usage guide updated for EL7-based TEM farm.
- 2021-01-18 : CryoSPARC usage guide modified for v3.0.1.
- 2021-01-13 : Information about newly created analysis cluster (based on RHEL7.x) added.
- 2020-11-23 : 2020 SNU CryoEM Hands-On Workshop presentation material added (appendix).
- 2020-03-30 : CryoSPARC trouble shooting updated (Gctf, MotionCor2 locations added).
- 2020-03-25 : CryoSPARC usage guide added.
- 2020-01-22 : This document version 1.0 released.
- 2020-02-24 : Appendix: Application form and materials added.
- 2020-02-21 : Dedicated queues list removed. Shared queues (cpuQ, gpuQ) and their abstract descriptions added.
- 2019-11-01 : Examples of running cisTEM jobs are added.
- 2019-10-15 : Examples of running relion jobs are added.
- 2019-05-27 : Initial version of GSDC TEM users guide added.

GSDC COMPUTING CLUSTER FOR TEM USERS AND OPERATORS

2.1 1. Service overview

GSDC (Global Science experimental Data hub Center) provides data computing services, i.e., large-scale Cryo-EM data transfer, archiving and/or processing to Cryo-EM operators/users. Cryo-EM facilities which are operated by government-funded research institutes or academies, can be directly connected to GSDC via KREONET with 10+ Gbps dedicated/shared network links. GSDC also supports petabytes scale of high performance (and/or archiving) storages and CPU/GPU computing servers to help Cryo-EM users' scientific discoveries. Here is GSDC's computing and storage infrastructres for Cryo-EM operators/users.

Overall architecture between KBSI, SNU (Seoul National University), PNU (Pusan National University)'s Cryo-EM facilities and GSDC TEM computing cluster.



2.2 2. Computing and storage resources

Category	Name (sd-farm.kr)	Specification	Re- sources size
Login	tem-ui-el7	<ul style="list-style-type: none"> CPU : Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz 18Core * 2 CPUs RAM : DDR4 2,666MHz 16GB * 24EA (384GB) HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1) 	72 cores (H/T)
Login (CryoSP)	tem-cs-el7	<ul style="list-style-type: none"> CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs RAM : DDR4 8GB * 24 (192GB) HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1) 	56 cores (H/T)
Master	tem-ce-el7	<ul style="list-style-type: none"> CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs RAM : DDR4 8GB * 24 (192GB) HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1) 	56 cores (H/T)
Workers	tem-wn[1001-1002]-el7	<ul style="list-style-type: none"> CPU : Intel(R) Xeon(R) Gold 6150 CPU @ 2.70GHz 18Core * 2 CPUs RAM : DDR4 2,666MHz 16GB * 24EA (384GB) HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1) 	380 cores
	tem-wn[1003-1013]-el7	<ul style="list-style-type: none"> CPU : Intel(R) Xeon(R) CPU E5-2697v3 @ 2.60GHz 14Core * 2 CPUs RAM : DDR4 8GB * 24 (192GB) HDD : 12G SAS HDD 1.2TB * 2EA (RAID-1) 	
Workers	tem-gpu[01-03]-el7	<ul style="list-style-type: none"> CPU : Intel® Xeon® CPU E5-2690v4 @ 2.60GHz 14Core * 2 CPUs RAM : DDR4 16GB * 24 (384GB) SSD : 6G SATA SSD 800GB * 2EA (RAID-1) GPU : NVIDIA P100 16GB * 2EA 	<ul style="list-style-type: none"> 204 cores 14 GPGPUs
	tem-gpu[04-05]-el7	<ul style="list-style-type: none"> CPU : Intel® Xeon® CPU E5-2690v4 @ 2.60GHz 14Core * 2 CPUs RAM : DDR4 16GB * 16 (256GB) SSD : 6G SATA SSD 800GB * 2EA (RAID-1) GPU : NVIDIA P40 24GB * 2EA 	
	tem-gpu[06-07]-el7	<ul style="list-style-type: none"> CPU : Intel® Xeon® Gold 6226R CPU @ 2.90GHz 16Core * 2 CPUs RAM : DDR4 32GB * 12 (384GB) HDD : 12G SAS HDD 1.6TB * 2EA (RAID-1) 	
Storage	User home directory (Home)	100GB per each user account (/tem/home)	5

2.3 3. Cluster management softwares

Category	Name	Description	Version (module path)
OS System M/W	Scientific Linux	Operating system	7.9
	Environment module	<ul style="list-style-type: none"> Module environment https://modules.readthedocs.io/en/latest 	v4.4.1
	OpenPBS(torque)	<ul style="list-style-type: none"> Cluster resources management http://www.adaptivecomputing.com/products/torque 	v6.1.2
	OpenMPI	<ul style="list-style-type: none"> Messaging Pass Interface(MPI) Reference implementation for MPI standard https://www.open-mpi.org 	v4.0.3 (mpi/gcc/openmpi/4.0.3) (mpi/gcc/8.3.1/openmpi/4.0.3)
	cuda	<ul style="list-style-type: none"> Compute Unified Device Architecture(CUDA) NVIDIA CUDA Runtime & Toolkit https://developer.nvidia.com/cuda-toolkit 	9.2 (cuda/9.2) 11.2 (cuda/11.2)
	Anaconda	<ul style="list-style-type: none"> Python based virtual environment https://www.anaconda.com/ 	2020.11 (conda/2020.11)
	Python	<ul style="list-style-type: none"> Python runtime 	v2.7.5

2.4 4. Data analysis tools

Category	Name	Description	Version (module path)
Tools	Relion	<p>A stand-alone computer program that employs an empirical Bayesian approach to refinement of (multiple) 3D reconstructions or 2D class averages in electron cryo-microscopy (cryo-EM).</p> <ul style="list-style-type: none"> • https://www3.mrc-lmb.cam.ac.uk/relion/index.php 	v3.0.7 (apps/relion/cpu/3.0.7) (apps/relion/gpu/3.0.7) v3.1.0 (apps/relion/cpu/3.1.0) (apps/relion/gpu/3.1.0) v4.0.0 (apps/relion/cpu/4.0.0) (apps/relion/gpu/4.0.0)
	cisTEM	<p>User-friendly software to process cryo-EM images of macromolecular complexes and obtain high-resolution 3D reconstructions.</p> <ul style="list-style-type: none"> • https://cistem.org 	v1.0.0 (apps/cistem/1.0.0)
	CryoSPARC	<p>CryoSPARC is the state-of-the-art platform used globally for obtaining 3D structural information from single particle cryo-EM data.</p> <ul style="list-style-type: none"> • https://cryosparc.com 	v3.0.1 v3.2.0 v4.0.0
	Topaz	<p>A pipeline for particle detection in cryoem images using convolutional neural networks trained from positive and unlabeled data.</p> <ul style="list-style-type: none"> • https://github.com/tbepler/topaz 	v0.2.4 (topaz/cuda-9.2/0.2.4) (topaz/cuda-11.0/0.2.4)
	PyEM	<p>A collection of Python modules and command-line utilities for</p>	v0.5 (pyem/0.5)
8	Chapter 2. GSDBC Computing Cluster for TEM Users and Operators		
	samples.		
	<ul style="list-style-type: none"> • https://github.com/asarnow/ 		

2.5 5. Requesting user accounts and accessing GSDC TEM computing cluster

Requesting accounts and accessing TEM cluster

Before proceed to next, please read the documents [Understanding environment modules](#) and [Job manager \(Torque\)](#) to better understand module environment and batch job manager.

2.6 6. Module paths and job submission templates

2.6.1 Module paths for data analysis tools

```
$> module avail
----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0
apps/relion/cpu/3.0.7
apps/relion/cpu/3.1.0
apps/relion/cpu/4.0.0
apps/relion/gpu/3.0.7
apps/relion/gpu/3.1.0
apps/relion/gpu/4.0.0

----- /tem/el7/Modules/acceleration -----
cuda/9.2 cuda/11.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/8.3.1/mpich/3.4.3
mpi/gcc/8.3.1/openmpi/4.0.3
mpi/gcc/openmpi/4.0.3

----- /tem/el7/Modules/virtualenv -----
conda/2020.11
pyem/0.5
topaz/cuda-9.2/0.2.4
topaz/cuda-11.0/0.2.4

----- /tem/el7/Modules/tools -----
tools/aspera-cli/3.9.6
tools/ctffind/4.1.14
tools/gctf/1.18_b2
tools/motioncor2/1.3.1
tools/resmap/1.1.4
tools/summovie/1.0.2
tools/unblur/1.0.2

----- /tem/el7/Modules/experiment -----
devel/python/3.7
PyRosetta/4
rosetta/mpich-3.4.3/3.13
rosetta/openmpi-4.0.3/3.13
```

2.6.2 Job submission templates

```
## output, error 로그 파일을 생성하지 않는 cisTEM 작업 템플릿
/tem/el7/qsub-cisTEM-cpu-noout.sh

## output, error 로그 파일을 생성하는 cisTEM 작업 템플릿
/tem/el7/qsub-cisTEM-cpu.sh

## Relion 3.0.7 CPU MPI 작업 템플릿
/tem/el7/qsub-relion-3.0.7-cpu.bash

## Relion 3.0.7 GPU 가속 활용하는 MPI 작업 템플릿
/tem/el7/qsub-relion-3.0.7-gpu.bash

## Relion 3.1.0 CPU MPI 작업 템플릿
/tem/el7/qsub-relion-3.1.0-cpu.bash

## Relion 3.1.0 GPU 가속 활용하는 MPI 작업 템플릿
/tem/el7/qsub-relion-3.1.0-gpu.bash

## Relion 3.1.0에서 external job 으로 topaz 소프트웨어를 사용하는 작업 템플릿
/tem/el7/qsub-relion-3.1.0-topaz.bash

## Relion 4.0.0 CPU MPI 작업 템플릿
/tem/el7/qsub-relion-4.0.0-cpu.bash

## Relion 4.0.0 GPU 가속 활용하는 MPI 작업 템플릿
/tem/el7/qsub-relion-4.0.0-gpu.bash

## Relion 4.0.0에서 external job 으로 topaz 소프트웨어를 사용하는 작업 템플릿
/tem/el7/qsub-relion-4.0.0-topaz.bash
```

2.7 7. Batch queues

Category	Queue Name	Assigned Computing Resources	Remarks
Shared	cpuQ	<ul style="list-style-type: none">tem-wn[1001-1002]-el7.sdfarm.kr (36 cores and 384GB memory per node)tem-wn[1003-1013]-el7.sdfarm.kr (28 cores and 192GB memory per node)	<ul style="list-style-type: none">380 Physical CPU cores
	gpuQ	<ul style="list-style-type: none">tem-gpu[01-03]-el7.sdfarm.kr (28 cores, 2 P100 GPUs and 384GB mem.)tem-gpu[04-05]-el7.sdfarm.kr (28 cores, 2 P40 GPUs and 256GB mem.)tem-gpu[06-07]-el7.sdfarm.kr (32 cores, 2 A100 GPUs and 256GB mem.)	<ul style="list-style-type: none">204 Physical CPU cores14 GPUsP100 16GB device memoryP40 24GB device memoryA100 40GB device memory

2.7.1 Checking batch queue names and their status

```
$> qstat -Qf
Queue: cpuQ
queue_type = Execution
total_jobs = 0
state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0 Complete:0
resources_default.neednodes = cpuQ
resources_default.nodes = 1
acl_group_enable = True
acl_groups = tem_users
acl_group_sloppy = True
mtime = 1610553300
resources_assigned.nodect = 0
enabled = True
started = True

Queue: gpuQ
queue_type = Execution
total_jobs = 0
state_count = Transit:0 Queued:0 Held:0 Waiting:0 Running:0 Exiting:0 Complete:0
resources_default.neednodes = gpuQ
resources_default.nodes = 1
acl_group_enable = True
acl_groups = tem_users
acl_group_sloppy = True
mtime = 1610553300
resources_assigned.nodect = 0
enabled = True
started = True
```

2.7.2 Checking all worker nodes status

```
$> pbsnodes -a
tem-wn1001-el7.sdfarm.kr
state = free
power_state = Running
np = 36
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1001-el7.sdfarm.kr 3.10.0-1160.6.1.el7.x86_64 #1 SMP Tue Nov □
→ 10 08:19:23 CST 2020 x86_64,sessions=2125,nsessions=1,nusers=1,idletime=3189604,
→ totmem=400927652kb,availmem=386021536kb,physmem=394636200kb,ncpus=36,loadave=0.02,gres=,
→ netload=368024574355580,state=free,varattr= ,cpuclock=Fixed,macaddr=34:80:0d:46:cc:88,version=6.1.
→ 2,rectime=1610587316,jobs=
mom_service_port = 15002
mom_manager_port = 15003

tem-wn1002-el7.sdfarm.kr
state = free
power_state = Running
np = 36
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1002-el7.sdfarm.kr 3.10.0-1160.2.2.el7.x86_64 #1 SMP Mon Oct □
(continues on next page)
```

(continued from previous page)

```
↪19 10:20:12 CDT 2020 x86_64,sessions=1980,nsessions=1,nusers=1,idletime=3189585,  
↪totmem=400927812kb,availmem=386052592kb,physmem=394636360kb,ncpus=36,loadave=0.00,gres=,  
↪netload=467274352677137,state=free,varattr=.cpuclock=Fixed,macaddr=f4:e9:d4:67:a5:0c,version=6.1.2,  
↪rectime=1610587321,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003

tem-wn1003-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1003-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec 3 10:20:12 CDT 2020
↪15 08:51:23 CST 2020 x86_64,sessions=16988,30464,nsessions=2,nusers=2,idletime=77442,  
↪totmem=204113112kb,availmem=197470212kb,physmem=197821660kb,ncpus=28,loadave=0.00,gres=,  
↪netload=7771760205,state=free,varattr=.cpuclock=Fixed,macaddr=24:6e:96:01:df:d0,version=6.1.2,  
↪rectime=1610587306,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003

tem-wn1004-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1004-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec 3 10:20:12 CDT 2020
↪15 08:51:23 CST 2020 x86_64,sessions=21911,nsessions=1,nusers=1,idletime=84377,totmem=204113112kb,  
↪availmem=197460724kb,physmem=197821660kb,ncpus=28,loadave=0.19,gres=,netload=9209594231,  
↪state=free,varattr=.cpuclock=Fixed,macaddr=24:6e:96:01:df:c0,version=6.1.2,rectime=1610587297,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003

tem-wn1005-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1005-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec 3 10:20:12 CDT 2020
↪15 08:51:23 CST 2020 x86_64,sessions=2032,nsessions=1,nusers=1,idletime=84135,totmem=204113112kb,  
↪availmem=197566008kb,physmem=197821660kb,ncpus=28,loadave=0.00,gres=,netload=9652090409,  
↪state=free,varattr=.cpuclock=Fixed,macaddr=24:6e:96:02:de:b0,version=6.1.2,rectime=1610587295,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003

tem-wn1006-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-wn1006-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec 3 10:20:12 CDT 2020
↪15 08:51:23 CST 2020 x86_64,sessions=22262,nsessions=1,nusers=1,idletime=84367,  
↪totmem=204113112kb,availmem=197470252kb,physmem=197821660kb,ncpus=28,loadave=0.00,gres=,
```

(continues on next page)

(continued from previous page)

```
↳netload=9653528113,state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:01:e1:70,version=6.1.2,  
↳rectime=1610587303,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003  
  
tem-wn1007-el7.sdfarm.kr  
state = free  
power_state = Running  
np = 28  
properties = cpuQ  
ntype = cluster  
status = opsys=linux,uname=Linux tem-wn1007-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec □  
↳15 08:51:23 CST 2020 x86_64,sessions=15172,nsessions=1,nusers=1,idletime=84349,  
↳totmem=204113112kb,availmem=197490356kb,physmem=197821660kb,ncpus=28,loadave=0.08,gres=,  
↳netload=7246363991,state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:02:e3:80,version=6.1.2,  
↳rectime=1610587301,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003  
  
tem-wn1008-el7.sdfarm.kr  
state = free  
power_state = Running  
np = 28  
properties = cpuQ  
ntype = cluster  
status = opsys=linux,uname=Linux tem-wn1008-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec □  
↳15 08:51:23 CST 2020 x86_64,sessions=22147,nsessions=1,nusers=1,idletime=84323,  
↳totmem=204113112kb,availmem=197470664kb,physmem=197821660kb,ncpus=28,loadave=0.00,gres=,  
↳netload=6170249241,state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:02:df:50,version=6.1.2,  
↳rectime=1610587299,jobs=  
mom_service_port = 15002  
mom_manager_port = 15003  
  
tem-wn1009-el7.sdfarm.kr  
state = job-exclusive  
power_state = Running  
np = 28  
properties = cpuQ  
ntype = cluster  
jobs = 0-13/307.tem-ce-el7.sdfarm.kr,14-27/308.tem-ce-el7.sdfarm.kr  
status = opsys=linux,uname=Linux tem-wn1009-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue □  
↳Dec 15 08:51:23 CST 2020 x86_64,sessions=1637 21403 21462,nsessions=3,nusers=2,idletime=124523,  
↳totmem=204113112kb,availmem=82190600kb,physmem=197821660kb,ncpus=28,loadave=28.02,gres=,  
↳netload=5715573075825,state=free,varattr=,cpuclock=Fixed,macaddr=ec:f4:bb:e9:cd:28,version=6.1.2,  
↳rectime=1611712971,jobs=307.tem-ce-el7.sdfarm.kr 308.tem-ce-el7.sdfarm.kr  
mom_service_port = 15002  
mom_manager_port = 15003  
  
tem-wn1010-el7.sdfarm.kr  
state = job-exclusive  
power_state = Running  
np = 28  
properties = cpuQ  
ntype = cluster  
jobs = 0-13/307.tem-ce-el7.sdfarm.kr,14-27/308.tem-ce-el7.sdfarm.kr  
status = opsys=linux,uname=Linux tem-wn1010-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue □
```

(continues on next page)

(continued from previous page)

```
↳ Dec 15 08:51:23 CST 2020 x86_64,sessions=10683 10742 21656,nsessions=3,nusers=2,idletime=125228,  
↳ totmem=204113112kb,availmem=82076700kb,physmem=197821660kb,ncpus=28,loadave=28.41,gres=,  
↳ netload=10000812494662,state=free,varattr=,cpuclock=Fixed,macaddr=ec:f4:bb:e9:c8:e0,version=6.1.2,  
↳ rectime=1611712972,jobs=307.tem-ce-el7.sdfarm.kr 308.tem-ce-el7.sdfarm.kr  
mom_service_port = 15002  
mom_manager_port = 15003

tem-wn1011-el7.sdfarm.kr
state = job-exclusive
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
jobs = 0-13/307.tem-ce-el7.sdfarm.kr,14-27/308.tem-ce-el7.sdfarm.kr
status = opsys=linux,uname=Linux tem-wn1011-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue □
↳ Dec 15 08:51:23 CST 2020 x86_64,sessions=10368 10428 21655,nsessions=3,nusers=2,idletime=128086,  
↳ totmem=204113112kb,availmem=81587604kb,physmem=197821660kb,ncpus=28,loadave=28.16,gres=,  
↳ netload=5807235665327,state=free,varattr=,cpuclock=Fixed,macaddr=ec:f4:bb:e9:bf:28,version=6.1.2,  
↳ rectime=1611712972,jobs=307.tem-ce-el7.sdfarm.kr 308.tem-ce-el7.sdfarm.kr
mom_service_port = 15002
mom_manager_port = 15003

tem-wn1012-el7.sdfarm.kr
state = job-exclusive
power_state = Running
np = 28
properties = cpuQ
ntype = cluster
jobs = 0-13/307.tem-ce-el7.sdfarm.kr,14-27/308.tem-ce-el7.sdfarm.kr
status = opsys=linux,uname=Linux tem-wn1012-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue □
↳ Dec 15 08:51:23 CST 2020 x86_64,sessions=10379 10475 21655,nsessions=3,nusers=2,idletime=127792,  
↳ totmem=204113112kb,availmem=84717576kb,physmem=197821660kb,ncpus=28,loadave=28.27,gres=,  
↳ netload=10075699597211,state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:02:de:d0,version=6.1.2,  
↳ rectime=1611712971,jobs=307.tem-ce-el7.sdfarm.kr 308.tem-ce-el7.sdfarm.kr
mom_service_port = 15002
mom_manager_port = 15003

tem-gpu01-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = gpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu01-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec □
↳ 15 08:51:23 CST 2020 x86_64,sessions=1823 4268,nsessions=2,nusers=2,idletime=36086,  
↳ totmem=402281596kb,availmem=390304804kb,physmem=395990144kb,ncpus=28,loadave=0.05,gres=,  
↳ netload=2091843090,state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:77:a0:80,version=6.1.2,  
↳ rectime=1610587294,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=00000000:82:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
↳ id=00000000:82:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↳ memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↳ utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↳ temperature=28 C, gpu[0]=gpu_id=00000000:03:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
```

(continues on next page)

(continued from previous page)

```
↪id=00000000:03:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=29 C;gpu_display=Enabled,driver_ver=460.27.04,timestamp=Thu Jan □
↪14 10:21:33 2021
```

```
tem-gpu02-e17.sdfarm.kr
state = free
power_state = Running
np = 28
properties = gpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu02-e17.sdfarm.kr 3.10.0-1160.11.1.e17.x86_64 #1 SMP Tue Dec □
↪15 08:51:23 CST 2020 x86_64,sessions=2142,nsessions=1,nusers=1,idletime=35378,totmem=402277340kb,
↪availmem=390086436kb,physmem=395985888kb,ncpus=56,loadave=0.09,gres=,netload=2464164051,
↪state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:77:9b:30,version=6.1.2,rectime=1610587314,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=00000000:03:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
↪id=00000000:03:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=27 C, gpu[0]=gpu_id=00000000:03:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
↪id=00000000:03:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=33 C;gpu_display=Enabled,driver_ver=460.27.04,timestamp=Thu Jan □
↪14 10:21:52 2021
```

```
tem-gpu03-e17.sdfarm.kr
state = free
power_state = Running
np = 28
properties = gpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu03-e17.sdfarm.kr 3.10.0-1160.11.1.e17.x86_64 #1 SMP Tue Dec □
↪15 08:51:23 CST 2020 x86_64,sessions=1816,nsessions=1,nusers=1,idletime=34739,totmem=402281596kb,
↪availmem=390290980kb,physmem=395990144kb,ncpus=28,loadave=0.10,gres=,netload=1338950655,
↪state=free,varattr=,cpuclock=Fixed,macaddr=24:6e:96:77:9b:10,version=6.1.2,rectime=1610587315,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=00000000:03:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
↪id=00000000:03:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=29 C, gpu[0]=gpu_id=00000000:03:00.0;gpu_pci_device_id=368578782;gpu_pci_location_
↪id=00000000:03:00.0;gpu_product_name=Tesla P100-PCIE-16GB;gpu_memory_total=16280 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=28 C;gpu_display=Enabled,driver_ver=460.27.04,timestamp=Thu Jan □
↪14 10:21:53 2021
```

```
tem-gpu04-e17.sdfarm.kr
state = free
```

(continues on next page)

(continued from previous page)

```
power_state = Running
np = 28
properties = gpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu04-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec □
→ 15 08:51:23 CST 2020 x86_64,sessions=2041,nsessions=1,nusers=1,idletime=63469,totmem=137732192kb,
→ availmem=132548340kb,physmem=131440740kb,ncpus=48,loadave=0.10,gres=,netload=790032261080,
→ state=free,varattr= ,cpuclock=Fixed,macaddr=e4:43:4b:07:8c:f0,version=6.1.2,rectime=1611712958,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=00000000:AF:00.0;gpu_pci_device_id=456659166;gpu_pci_location_
→ id=00000000:AF:00.0;gpu_product_name=Tesla P40;gpu_memory_total=22919 MB;gpu_memory_used=0 □
→ MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_
→ mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_temperature=28 C,
→ gpu[0]=gpu_id=00000000:3B:00.0;gpu_pci_device_id=456659166;gpu_pci_location_id=00000000:3B:00.0;
→ gpu_product_name=Tesla P40;gpu_memory_total=22919 MB;gpu_memory_used=0 MB;gpu_mode=Default;
→ gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Enabled;gpu_
→ single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_temperature=25 C;gpu_display=Enabled,gpu_
→ display=Enabled,driver_ver=460.32.03,timestamp=Wed Jan 27 11:02:37 2021

tem-gpu05-el7.sdfarm.kr
state = free
power_state = Running
np = 28
properties = gpuQ
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu05-el7.sdfarm.kr 3.10.0-1160.11.1.el7.x86_64 #1 SMP Tue Dec □
→ 15 08:51:23 CST 2020 x86_64,sessions=2352,nsessions=1,nusers=1,idletime=63492,totmem=269906392kb,
→ availmem=261305348kb,physmem=263614940kb,ncpus=72,loadave=0.13,gres=,netload=808539072,
→ state=free,varattr= ,cpuclock=Fixed,macaddr=e4:43:4b:03:78:38,version=6.1.2,rectime=1611712989,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=00000000:AF:00.0;gpu_pci_device_id=456659166;gpu_pci_location_
→ id=00000000:AF:00.0;gpu_product_name=Tesla P40;gpu_memory_total=22919 MB;gpu_memory_used=0 □
→ MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_
→ mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_temperature=30 C,
→ gpu[0]=gpu_id=00000000:3B:00.0;gpu_pci_device_id=456659166;gpu_pci_location_id=00000000:3B:00.0;
→ gpu_product_name=Tesla P40;gpu_memory_total=22919 MB;gpu_memory_used=0 MB;gpu_mode=Default;
→ gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_utilization=0%;gpu_ecc_mode=Enabled;gpu_
→ single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_temperature=27 C;gpu_display=Enabled,gpu_
→ display=Enabled,driver_ver=460.32.03,timestamp=Wed Jan 27 11:03:08 2021

tem-gpu06-el7.sdfarm.kr
state = free
power_state = Running
np = 32
properties = gpuQ,gpuQA100
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu06-el7.sdfarm.kr 3.10.0-1160.49.1.el7.x86_64 #1 SMP Tue Nov □
→ 23 21:51:54 CST 2021 x86_64,sessions=1853,nsessions=1,nusers=1,idletime=78369,totmem=402049028kb,
→ availmem=396843552kb,physmem=395757576kb,ncpus=32,loadave=0.34,gres=,netload=2752372686,
→ state=free,varattr= ,cpuclock=Fixed,macaddr=f4:03:43:e5:19:40,version=6.1.2,rectime=1639028497,jobs=
mom_service_port = 15002
mom_manager_port = 15003
```

(continues on next page)

(continued from previous page)

```

gpus = 2
gpu_status = gpu[1]=gpu_id=0000000:D8:00.0;gpu_pci_device_id=552669406;gpu_pci_location_
↪id=0000000:D8:00.0;gpu_product_name=NVIDIA A100-PCIE-40GB;gpu_memory_total=40536 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=30%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=72 C;gpu[0]=gpu_id=0000000:86:00.0;gpu_pci_device_id=552669406;gpu_pci_location_
↪id=0000000:86:00.0;gpu_product_name=NVIDIA A100-PCIE-40GB;gpu_memory_total=40536 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=73 C;gpu_display=Enabled,driver_ver=495.29.05,timestamp=Thu Dec □
↪9 14:41:35 2021

tem-gpu07-el7.sdfarm.kr
state = free
power_state = Running
np = 32
properties = gpuQ,gpuQA100
ntype = cluster
status = opsys=linux,uname=Linux tem-gpu07-el7.sdfarm.kr 3.10.0-1160.49.1.el7.x86_64 #1 SMP Tue Nov □
↪23 21:51:54 CST 2021 x86_64,sessions=1855 2925,nsessions=2,nusers=2,idletime=77023,
↪totmem=402049028kb,availmem=396857460kb,physmem=395757576kb,ncpus=32,loadave=0.05,gres=,
↪netload=2832872237,state=free,varattr=,cpuclock=Fixed,macaddr=f4:03:43:e5:19:20,version=6.1.2,
↪rectime=1639028495,jobs=
mom_service_port = 15002
mom_manager_port = 15003
gpus = 2
gpu_status = gpu[1]=gpu_id=0000000:D8:00.0;gpu_pci_device_id=552669406;gpu_pci_location_
↪id=0000000:D8:00.0;gpu_product_name=NVIDIA A100-PCIE-40GB;gpu_memory_total=40536 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=31%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=74 C;gpu[0]=gpu_id=0000000:86:00.0;gpu_pci_device_id=552669406;gpu_pci_location_
↪id=0000000:86:00.0;gpu_product_name=NVIDIA A100-PCIE-40GB;gpu_memory_total=40536 MB;gpu_
↪memory_used=0 MB;gpu_mode=Default;gpu_state=Unallocated;gpu_utilization=0%;gpu_memory_
↪utilization=0%;gpu_ecc_mode=Enabled;gpu_single_bit_ecc_errors=0;gpu_double_bit_ecc_errors=0;gpu_
↪temperature=73 C;gpu_display=Enabled,driver_ver=495.29.05,timestamp=Thu Dec □
↪9 14:41:33 2021

```

2.8 8. fstat.bin : Monitoring the usage of all the worker nodes

##fstat.bin tool is available on tem-ui-el7.sdfarm.kr and tem-cs-el7.sdfarm.kr login nodes

```
$> which fstat.bin
/usr/bin/fstat.bin
```

```
$> fstat.bin
```

NODE	QUEUE	STATUS(F/S/E)	[GPU] T/U/F	[CPU] T/U/F	USAGE RATIO
tem-gpu01-el7.sdfarm.kr	gpuQ	Shared	2/1/1 [.]	28/2/26 [##.....]	
tem-gpu02-el7.sdfarm.kr	gpuQ	Shared	2/2/0 [##]	28/4/24 [####.....]	
tem-gpu03-el7.sdfarm.kr	gpuQ	Free	2/0/2 [..]	28/0/28 [.....]	
tem-gpu04-el7.sdfarm.kr	gpuQ	Free	2/0/2 [..]	28/0/28 [.....]	
tem-gpu05-el7.sdfarm.kr	gpuQ	Free	2/0/2 [..]	28/0/28 [.....]	
tem-gpu06-el7.sdfarm.kr	gpuQ	Free	2/0/2 [..]	32/0/32 [.....]	

(continues on next page)

(continued from previous page)

tem-gpu07-el7.sdfarm.kr	gpuQ	Free	2/0/2 [..]	32/0/32 [.....]
tem-wn1001-el7.sdfarm.kr	cpuQ	Shared	n/a	36/28/8 [#####.....]
↳...]				
tem-wn1002-el7.sdfarm.kr	cpuQ	Shared	n/a	36/28/8 [#####.....]
↳...]				
tem-wn1003-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1004-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1005-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1006-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1007-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1008-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1009-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1010-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1011-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1012-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]
tem-wn1013-el7.sdfarm.kr	cpuQ	Exclusive	n/a	28/28/0 [#####.....]

7 running jobs

1 queued(waiting) jobs

Total 584 cores / Used 342 cores (utilization 58.5 percent)

(f) Enter f to display farm (nodes) status.

(j) Enter j to display jobs.

(g) Enter g to display GPUs status.

(q) Quit.

Select? (f/j/g/q) __

* NODE : CPU 또는 GPU 장치를 가진 계산서버 이름

* QUEUE : 각 서버가 속한 큐 이름

* STATUS(F/S/E)

- F (Free) : 계산서버에 어떤 데이터 분석 작업도 할당되어 있지 않음

- S (Shared) : 계산서버에 CPU 또는 GPU 작업이 할당되어 실행중이나, 해당 서버의 모든 자원을 할당받은 □

↳상태는 아님

- E (Exclusive) : 계산서버에 작업들이 할당되어 실행중이고, 작업들이 모든 자원을 할당받아 busy 한 상태

* [GPU] T/U/F : GPU 계산서버에 설치된 GPU 카드 총 개수, 사용중인 개수(#), 유류 카드 개수(.)

* [CPU] T/U/F : CPU 계산서버의 총 코어 개수(#), 유류 코어 개수(.)

2.9 9. dynmotd : Checking storage quota limit and usage ratio

dynmotd tool is available on tem-ui-el7.sdfarm.kr, tem-cs-el7.sdfarm.kr and tem-dtn-el7.sdfarm.kr nodes

```
$> which dynmotd
/usr/local/bin/dynmotd
```

```
$> dynmotd
```



* Official GSDC TEM users guide : <https://tem-docs.readthedocs.io>

(continues on next page)

(continued from previous page)

```
=====
* Hostname.....: tem-ui-el7.sdfarm.kr
* OS Release....: Scientific Linux release 7.9 (Nitrogen)
* System uptime.: 5 days 2 hours 2 minutes 39 seconds
* Users.........: Currently 5 user(s) logged on
* Processes.....: 920 running
* CPU usage.....: 0.07, 0.85, 1.30 (1, 5, 15 min)
* Memory (used/total)...: 13445 MB / 386699 MB
* Swap in use....: 0 MB
-----
* TEM Storage (used/total)....: 383 TB / 5,836.8 TB (7%)
* Current User.....: <UserID>
* User Home Directory.....: /tem/home/<UserID>
  ** Disk Quota Limit.....: 0k
  ** Disk Usage.....: 250.8 TB
  ** Number of Files.....: 21,785,501
* Group Scratch Directory.....: /tem/scratch/<GroupDir>
  ** Disk Quota Limit.....: 40 TB
  ** Disk Usage.....: 13.01 GB
  ** Number of Files.....: 269,991
=====
```


TEM COMPUTING CLUSTER BASICS

3.1 Requesting accounts and accessing TEM cluster

Before you use GSDC' s TEM cluster, you should send an application form to TEM service manager by e-mail and get an user account to access the computing cluster (please see [APPENDIX : Application Form & Materials](#) section). If you already have valid user accounts, you can make a connection to UI (user interface) nodes to use various kind of cluster computing/storage resources and software environments (including data analysis tools, e.g., relion, cisTEM, CryoSPARC, etc.).

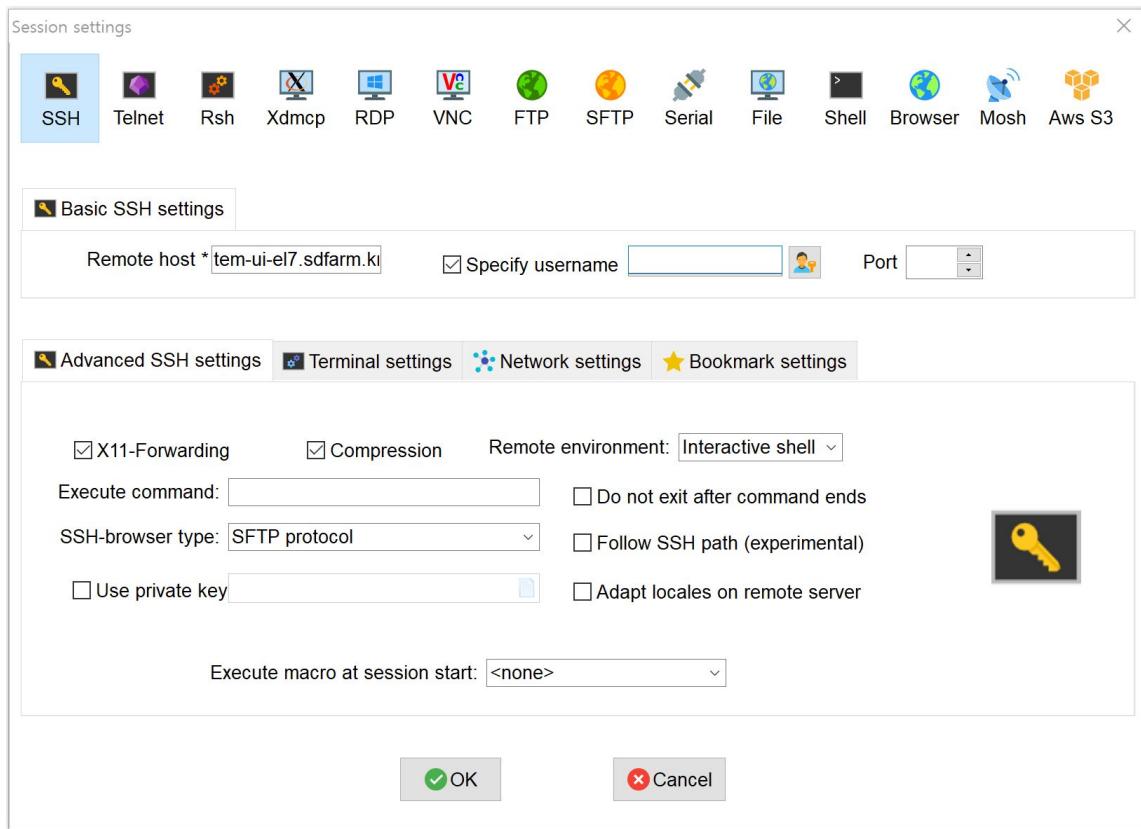
3.1.1 For Linux/Mac users

```
$> ssh -Y -o Port=<port> <userID>@tem-ui-el7.sdfarm.kr
```

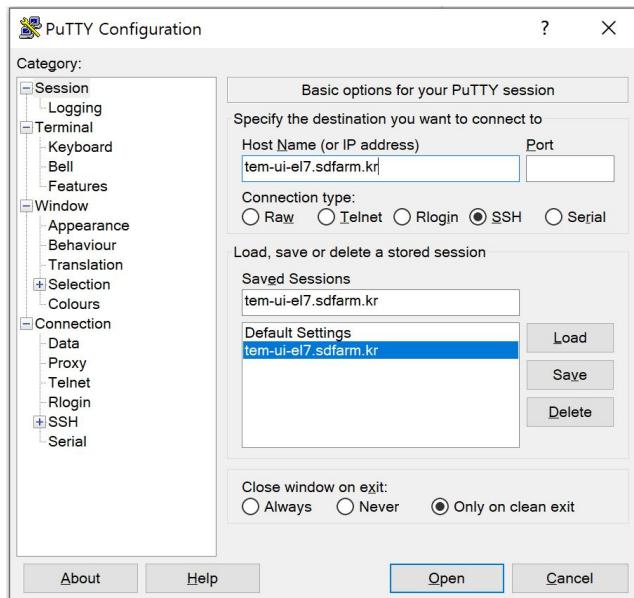
-Y (or -X) options : enable trusted X11 (or untrusted X11) forwarding

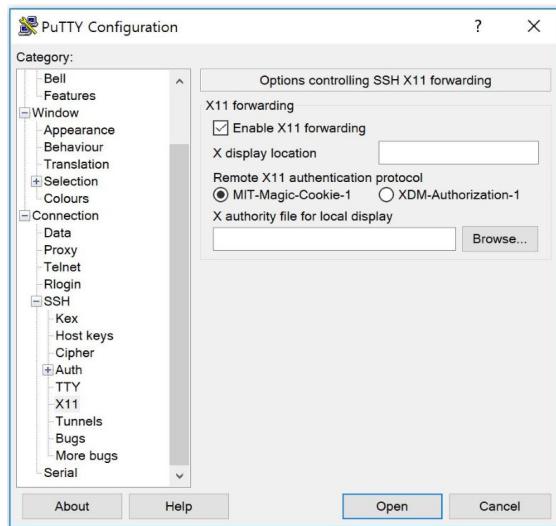
3.1.2 For Windows users

- Using MobaXterm (<https://mobaxterm.mobatek.net>) : MobaXterm is an enhanced terminal for Windows with self-contained X11 server, tabbed SSH client, network tools and much more.



- Using Putty with Xwindows manager (e.g., Xming, Xmanager, etc.) (<https://www.putty.org>) : If you use putty terminal application, you must install a 3rd-party Xwindows manager in advance.





3.2 Understanding environment modules

The Environment Modules system is a tool to help users manage their Unix or Linux shell environment, by allowing groups of related environment-variable settings to be made or removed dynamically.

- Listing available modules

```
$> module avail
----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0
apps/relion/cpu/3.0.7
apps/relion/cpu/3.1.0
apps/relion/cpu/4.0.0
apps/relion/gpu/3.0.7
apps/relion/gpu/3.1.0
apps/relion/gpu/4.0.0

----- /tem/el7/Modules/acceleration -----
cuda/9.2 cuda/11.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/4.8.5/openmpi/4.0.3
mpi/gcc/8.3.1/mpich/3.4.3
mpi/gcc/8.3.1/openmpi/4.0.3
mpi/gcc/openmpi/4.0.3

----- /tem/el7/Modules/virtualenv -----
conda/2020.11 topaz/cuda-9.2/0.2.4
pyem/0.5 topaz/cuda-11.0/0.2.4

----- /tem/el7/Modules/tools -----
tools/aspera-cli/3.9.6
tools/ctffind/4.1.14
tools/gctf/1.18_b2
tools/motioncor2/1.3.1
tools/resmap/1.1.4
tools/summovie/1.0.2
tools/unblur/1.0.2
```

(continues on next page)

(continued from previous page)

```
----- /tem/el7/Modules/experiment -----  
PyRosetta/4  
python/3.7  
rosetta/mpich-3.4.3/3.13  
rosetta/openmpi-4.0.3/3.13
```

- Show module details

```
$> module show apps/relion/gpu/4.0.0  
  
-----  
/tem/el7/Modules/apps/apps/relion/gpu/4.0.0:  
  
module-whatis {Setups relion 4.0.0 environment variables}  
module      load mpi/gcc/openmpi/4.0.3  
module      load cuda/11.2  
setenv     relion_version 4.0.0  
prepend-path PATH /tem/el7/relion-4.0.0/gpu/bin  
prepend-path LD_LIBRARY_PATH /tem/el7/relion-4.0.0/gpu/lib  
setenv     LANG en_US.UTF-8  
setenv     RELION_QUEUE_USE yes  
setenv     RELION_QUEUE_NAME gpuQ  
setenv     RELION_QSUB_COMMAND qsub  
setenv     RELION_QSUB_EXTRA_COUNT 3  
setenv     RELION_QSUB_EXTRA1 {Number of Nodes}  
setenv     RELION_QSUB_EXTRA2 {Number of processes per each node}  
setenv     RELION_QSUB_EXTRA3 {Number of GPUs per node}  
setenv     RELION_QSUB_EXTRA1_DEFAULT 1  
setenv     RELION_QSUB_EXTRA2_DEFAULT 3  
setenv     RELION_QSUB_EXTRA3_DEFAULT 2  
setenv     RELION_CTFFIND_EXECUTABLE /tem/el7/ctffind-4.1.14/bin/ctffind  
setenv     RELION_GCTF_EXECUTABLE /tem/el7/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.2  
setenv     RELION_RESMAP_EXECUTABLE /tem/el7/ResMap-1.1.4/ResMap-1.1.4-linux64  
setenv     RELION_MOTIONCOR2_EXECUTABLE /tem/el7/MotionCor2_v1.3.1/MotionCor2_v1.3.1-  
→ Cuda92  
setenv     RELION_UNBLUR_EXECUTABLE /tem/el7/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe  
setenv     RELION_SUMMOVIE_EXECUTABLE /tem/el7/summovie_1.0.2/bin/sum_movie_openmp_7_  
→ 17_15.exe  
conflict    apps/relion  
-----
```

- Loading modules

```
$> module load <module_path>  
or  
$> module add <module_path>  
e.g., $> module load apps/relion/gpu/4.0.0
```

- Listing loaded modules

```
$> module load apps/relion/gpu/4.0.0  
$> module list  
Currently Loaded Modulefiles:  
1) mpi/gcc/openmpi/4.0.3 2) cuda/11.2 3) apps/relion/gpu/4.0.0
```

- **Unloading modules**

```
$> module unload <module_path>  
or  
$> module rm <module_path>  
e.g., $> module unload apps/relion/gpu/4.0.0
```

- **Unloading all the modules**

```
$> module purge
```

- **Module environment help**

```
$> module --help  
Modules Release 4.4.1 (2020-01-03)  
Usage: module [options] [command] [args ...]  
  
Loading / Unloading commands:  
add | load    modulefile [...] Load modulefile(s)  
rm | unload   modulefile [...] Remove modulefile(s)  
purge          Unload all loaded modulefiles  
reload | refresh      Unload then load all loaded modulefiles  
switch | swap  [mod1] mod2    Unload mod1 and load mod2  
  
Listing / Searching commands:  
list      [-t|-l]      List loaded modules  
avail  [-d|-L] [-t|-l] [-S|-C] [--indepth|--no-indepth] [mod ...]  
           List all or matching available modules  
aliases          List all module aliases  
whatis     [modulefile ...] Print whatis information of modulefile(s)  
apropos | keyword | search str  Search all name and whatis containing str  
is-loaded   [modulefile ...] Test if any of the modulefile(s) are loaded  
is-avail    modulefile [...] Is any of the modulefile(s) available  
info-loaded  modulefile    Get full name of matching loaded module(s)
```

Collection of modules handling commands:

```
save      [collection|file] Save current module list to collection  
restore   [collection|file] Restore module list from collection or file  
saverm    [collection]    Remove saved collection  
saveshow  [collection|file] Display information about collection  
savelist   [-t|-l]        List all saved collections  
is-saved   [collection ...] Test if any of the collection(s) exists
```

Shell's initialization files handling commands:

```
initlist      List all modules loaded from init file  
initadd      modulefile [...] Add modulefile to shell init file  
initrm      modulefile [...] Remove modulefile from shell init file  
initprepend  modulefile [...] Add to beginning of list in init file  
initswitch   mod1 mod2    Switch mod1 with mod2 from init file  
initclear    Clear all modulefiles from init file
```

Environment direct handling commands:

```
prepend-path [-d c] var val [...] Prepend value to environment variable  
append-path [-d c] var val [...] Append value to environment variable  
remove-path [-d c] var val [...] Remove value from environment variable
```

Other commands:

(continues on next page)

(continued from previous page)

```
help      [modulefile ...] Print this or modulefile(s) help info
display | show modulefile [...] Display information about modulefile(s)
test      [modulefile ...] Test modulefile(s)
use      [-a|-p] dir [...] Add dir(s) to MODULEPATH variable
unuse     dir [...] Remove dir(s) from MODULEPATH variable
is-used   [dir ...] Is any of the dir(s) enabled in MODULEPATH
path      modulefile Print modulefile path
paths     modulefile Print path of matching available modules
clear     [-f]      Reset Modules-specific runtime information
source    scriptfile [...] Execute scriptfile(s)
config   [--dump-state|name [val]] Display or set Modules configuration
```

Switches:

```
-t | --terse  Display output in terse format
-l | --long   Display output in long format
-d | --default Only show default versions available
-L | --latest  Only show latest versions available
-S | --starts-with
               Search modules whose name begins with query string
-C | --contains Search modules whose name contains query string
-i | --icase   Case insensitive match
-a | --append  Append directory to MODULEPATH
-p | --prepend Prepend directory to MODULEPATH
--auto      Enable automated module handling mode
--no-auto   Disable automated module handling mode
-f | --force   By-pass dependency consistency or confirmation dialog
```

Options:

```
-h | --help   This usage info
-V | --version Module version
-D | --debug  Enable debug messages
-v | --verbose Enable verbose messages
-s | --silent Turn off error, warning and informational messages
--paginate  Pipe mesg output into a pager if stream attached to terminal
--no-pager   Do not pipe message output into a pager
--color[=WHEN] Colorize the output; WHEN can be 'always' (default if
               omitted), 'auto' or 'never'
```

3.3 Job manager (Torque)

3.3.1 Resources manager and job scheduler

- Resource manager : Torque(OpenPBS) v6.1.2
- Job scheduler : Torque default FIFO job scheduler

3.3.2 Directives in Torque job scripts

Torque defines some useful directives (starting with '#PBS') which can be used to describe job's resources requirements. Users must include those directives in job scripts to submit and execute jobs. The order of directives is not important, but the directives must be written prior to job execution commands.

Resource limits

The “-l” option is used to request resources, including nodes, memory, time, etc.

- Nodes and PPN (Processor Per Node)

To request a single core on the farm:

```
#PBS -l nodes=1:ppn=1
```

To request one whole node on the farm:

```
#PBS -l nodes=1:ppn=28
```

To request 4 whole nodes on the farm:

```
#PBS -l nodes=4:ppn=28
```

To request 3 whole nodes with 2 GPUs on the farm:

```
#PBS -l nodes=3:ppn=28:gpus=2
```

To request 1 node with use of 6 cores and 1 GPU:

```
#PBS -l nodes=1:ppn=6:gpus=1
```

- Wall clock time

To request 20 hours of wall clock time:

```
#PBS -l walltime=20:00:00
```

If a computational job will have not finished yet until the specified wall clock time, Torque (or maui scheduler) will release the resources that are allocated to the job and stop the job's running. If you don't define walltime, the default value is "infinite".

- Memory

To request 4GB memory:

```
#PBS -l mem=4GB
```

or

```
#PBS -l mem=4000MB
```

To request 24GB memory:

```
#PBS -l mem=24000MB
```

Job name

You can define a job name using “-N” option. If you omit this directive, the default job name is the same as the file name of job script.

```
#PBS -N jobName
```

Queue name

In general, a “queue” can be thought of a mapped set of computing resources. You can specify a queue name (using “-q” option) which the job is enqueued to.

```
#PBS -q batch
```

Job log files

When Torque executes an user’s job, Torque creates 2 different types of log files (standard output stream and standard error stream) by default. If the job’s name is “*jobName*” and the submitted job ID is “123456”, you can find 2 files (*jobName.o123456* and *jobName.e123456*) that are created in the job execution base directory. You can also merge the two streams into one file using “-j oe” option. In that case, *jobName.o1234567* file contains the standard error stream.

```
#PBS -j oe
```

3.3.3 Torque job script examples

Simple sequential job

```
#PBS -N jobName
#PBS -l walltime=40:00:00
#PBS -l nodes=1:ppn=1
#PBS -q batch

cd $PBS_O_WORKDIR
/usr/bin/time ./mysci > mysci.hist
```

Serial job with OpenMP multithreading

```
#PBS -N jobName
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28
#PBS -q batch

export OMP_NUM_THREADS=28
cd $PBS_O_WORKDIR
./a.out > my_results
```

Simple MPI parallel job

Here is an example of an MPI job that uses 4 nodes with 4 cores each, running one process per core (16 processes total).

```
#PBS -N jobName
#PBS -l walltime=10:00:00
#PBS -l nodes=4:ppn=4
#PBS -q batch

module load mpi/gcc/openmpi/4.0.3
cd $PBS_O_WORKDIR
mpirun -machinefile $PBS_NODEFILE ./a.out
```

Parallel job with MPI and OpenMP

This example is a hybrid MPI/OpenMP job. It runs one MPI process per node with 28 threads per process. The assumption here is that the code was written to support multi-level parallelism.

```
#PBS -N jobName
#PBS -l walltime=20:00:00
#PBS -l nodes=4:ppn=28
#PBS -q batch

module load mpi/gcc/openmpi/4.0.3
export OMP_NUM_THREADS=28
cd $PBS_O_WORKDIR
mpirun --bynode -machinefile $PBS_NODEFILE ./a.out
```

3.3.4 Job submission

myscript.job : the script file name of a PBS batch job

```
$> qsub myscript.job
```

In response to this command you'll see a line with your job ID:

```
123456.tem-ce.sdfarm.kr
```

3.3.5 Monitoring and managing your jobs

Status of queued jobs

- qstat

Use the qstat command to check the status of your jobs. You can see whether your job is queued or running, along with information about requested resources. If the job is running you can see elapsed time and resources used.

By itself, qstat lists all jobs in the system in standard or alternate format:

```
$> qstat
```

or

```
$> qstat -a
```

qstat with -ns option lists all jobs with showing the assigned nodes for each job:

```
$> qstat -ns
```

To list all the jobs belonging to a particular user:

```
$> qstat -u tem_user
```

To list the status of a particular job, in standard or alternate format:

```
$> qstat 123456
```

```
$> qstat -a 123456
```

To get all the details about a particular job (full status):

```
$> qstat -f 123456
```

To list the status of all the queues

```
$> qstat -Qf
```

Managing your jobs

- Deleting (canceling) a job

Situations may arise in which you want to delete one of your jobs from the PBS queue. Perhaps you set the resource limits incorrectly, neglected to copy an input file, or had incorrect or missing commands in the batch file. Or maybe the program is taking too long to run (infinite loop). The PBS command to delete a batch job is qdel. It applies to both queued and running jobs.

```
$> qdel 123456
```

- Altering a queued job

You can alter certain attributes of your job while it's in the queue using the qalter command. This can be useful if you want to make a change without losing your place in the queue. You cannot make any alterations to the executable portion of the script, nor can you make any changes after the job starts running. The options argument consists of one or more PBS directives in the form of command-line options. For example, to change the walltime limit on job 123456 to 5 hours and have email sent when the job ends (only):

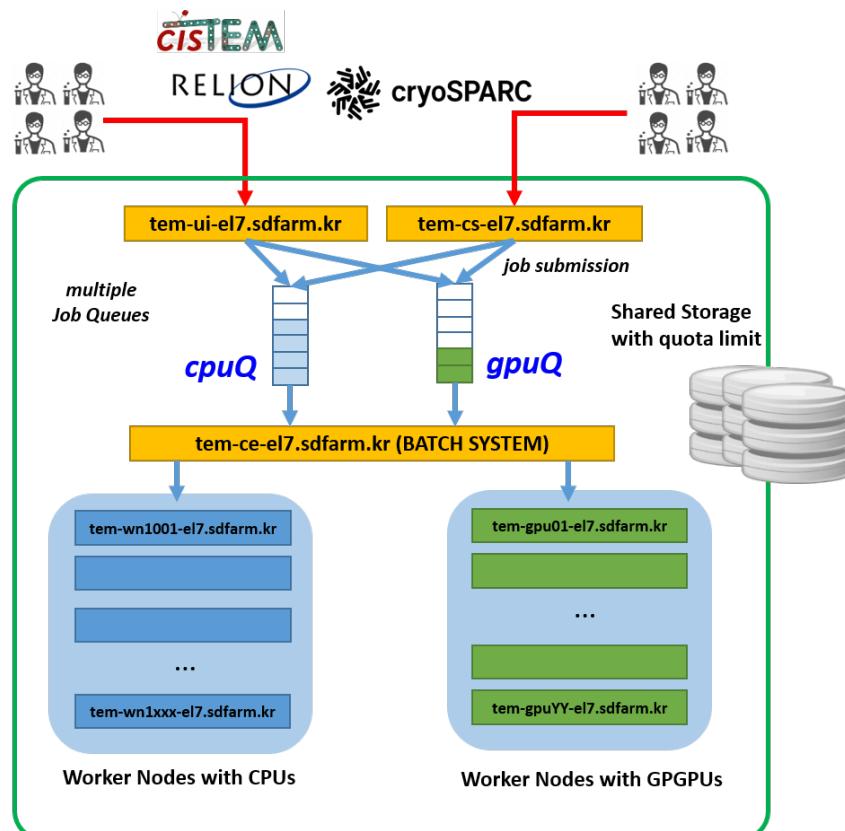
The syntax is: qalter [options ...] jobid

```
$> qalter -l walltime=5:00:00 -m e 123456
```

BATCH QUEUES

4.1 Batch queue list

TEM farm provides multiple batch queues with different characteristics to users who submit jobs to analyze large-scale Cryo-EM data. A batch queue means a logical set of CPU and GPU computing resources. Users interact with a specific queue to manage their own jobs. Within each queue, submitted jobs are executed in order (First-in-first-out). Note that multiple jobs requiring CPU and/or GPU resources can be executed concurrently if there are enough available resources in the queue.



Category	Queue Name	Assigned Computing Resources	Remarks
Shared	cpuQ	<ul style="list-style-type: none">tem-wn[1001-1002]-el7.sdfarm.kr (36 cores and 384GB memory per node)tem-wn[1003-1013]-el7.sdfarm.kr (28 cores and 192GB memory per node)	<ul style="list-style-type: none">380 Physical CPU cores
	gpuQ	<ul style="list-style-type: none">tem-gpu[01-03]-el7.sdfarm.kr (28 cores, 2 P100 GPUs and 384GB mem.)tem-gpu[04-05]-el7.sdfarm.kr (28 cores, 2 P40 GPUs and 256GB mem.)tem-gpu[06-07]-el7.sdfarm.kr (32 cores, 2 A100 GPUs and 256GB mem.)	<ul style="list-style-type: none">204 Physical CPU cores14 GPUsP100 16GB device memoryP40 24GB device memoryA100 40GB device memory

**CHAPTER
FIVE**

RELION

RELION (for REregularised LIkelihood OptimisatioN, pronounce rely-on) is a stand-alone computer program that employs an empirical Bayesian approach to refinement of (multiple) 3D reconstructions or 2D class averages in electron cryo-microscopy (cryo-EM). (from Relion official site https://www3.mrc-lmb.cam.ac.uk/relion/index.php?title=Main_Page)

5.1 Executing Relion GUI tools

5.1.1 How to start Relion data analysis tool

1. You can find out relion applications' environment module path by listing all the module available on TEM service farm

```
$> module avail

----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0
apps/relion/cpu/3.0.7
apps/relion/cpu/3.1.0
apps/relion/cpu/4.0.0
apps/relion/gpu/3.0.7
apps/relion/gpu/3.1.0
apps/relion/gpu/4.0.0

----- /tem/el7/Modules/acceleration -----
cuda/9.2 cuda/11.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/4.8.5/openmpi/4.0.3
mpi/gcc/8.3.1/mpich/3.4.3
mpi/gcc/8.3.1/openmpi/4.0.3
mpi/gcc/openmpi/4.0.3

----- /tem/el7/Modules/virtualenv -----
conda/2020.11 topaz/cuda-9.2/0.2.4
pyem/0.5 topaz/cuda-11.0/0.2.4

----- /tem/el7/Modules/tools -----
tools/aspera-cli/3.9.6
tools/ctffind/4.1.14
tools/gctf/1.18_b2
```

(continues on next page)

(continued from previous page)

```
tools/motioncor2/1.3.1
tools/resmap/1.1.4
tools/summovie/1.0.2
tools/unblur/1.0.2

----- /tem/el7/Modules/experiment -----
PyRosetta/4
python/3.7
rosetta/mpich-3.4.3/3.13
rosetta/openmpi-4.0.3/3.13
```

2. Check the module details for the specific relion version (e.g., Relion v4.0.0 with GPGPU support or Relion v4.0.0 with CPU cores support only)

```
$> module show apps/relion/gpu/4.0.0
```

```
-----  
/tem/el7/Modules/apps/apps/relion/gpu/4.0.0:  
  
module-whatis {Setups relion 4.0.0 environment variables}  
module      load mpi/gcc/openmpi/4.0.3  
module      load cuda/11.2  
setenv     relion_version 4.0.0  
prepend-path PATH /tem/el7/relion-4.0.0/gpu/bin  
prepend-path LD_LIBRARY_PATH /tem/el7/relion-4.0.0/gpu/lib  
setenv     LANG en_US.UTF-8  
setenv     RELION_QUEUE_USE yes  
setenv     RELION_QUEUE_NAME gpuQ  
setenv     RELION_QSUB_COMMAND qsub  
setenv     RELION_QSUB_EXTRA_COUNT 3  
setenv     RELION_QSUB_EXTRA1 {Number of Nodes}  
setenv     RELION_QSUB_EXTRA2 {Number of processes per each node}  
setenv     RELION_QSUB_EXTRA3 {Number of GPUs per node}  
setenv     RELION_QSUB_EXTRA1_DEFAULT 1  
setenv     RELION_QSUB_EXTRA2_DEFAULT 3  
setenv     RELION_QSUB_EXTRA3_DEFAULT 2  
setenv     RELION_CTFFIND_EXECUTABLE /tem/el7/ctffind-4.1.14/bin/ctffind  
setenv     RELION_GCTF_EXECUTABLE /tem/el7/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.2  
setenv     RELION_RESMAP_EXECUTABLE /tem/el7/ResMap-1.1.4/ResMap-1.1.4-linux64  
setenv     RELION_MOTIONCOR2_EXECUTABLE /tem/el7/MotionCor2_v1.3.1/MotionCor2_v1.3.1-  
→Cuda92  
setenv     RELION_UNBLUR_EXECUTABLE /tem/el7/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe  
setenv     RELION_SUMMOVIE_EXECUTABLE /tem/el7/summovie_1.0.2/bin/sum_movie_openmp_7_  
→17_15.exe  
conflict   apps/relion  
-----
```

or

```
$> module show apps/relion/cpu/4.0.0
```

```
-----  
/tem/el7/Modules/apps/apps/relion/cpu/4.0.0:  
  
module-whatis {Setups relion 4.0.0 environment variables}
```

(continues on next page)

(continued from previous page)

```
module      load mpi/gcc/openmpi/4.0.3
setenv     relion_version 4.0.0
prepend-path PATH /tem/el7/relion-4.0.0/cpu/bin
prepend-path LD_LIBRARY_PATH /tem/el7/relion-4.0.0/cpu/lib
setenv     LANG en_US.UTF-8
setenv     RELION_QUEUE_USE yes
setenv     RELION_QUEUE_NAME cpuQ
setenv     RELION_QSUB_COMMAND qsub
setenv     RELION_QSUB_EXTRA_COUNT 2
setenv     RELION_QSUB_EXTRA1 {Number of Nodes}
setenv     RELION_QSUB_EXTRA2 {Number of processes per each node}
setenv     RELION_QSUB_EXTRA1_DEFAULT 2
setenv     RELION_QSUB_EXTRA2_DEFAULT 16
setenv     RELION_CTFFIND_EXECUTABLE /tem/el7/ctffind-4.1.14/bin/ctffind
setenv     RELION_GCTF_EXECUTABLE /tem/el7/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.2
setenv     RELION_RESMAP_EXECUTABLE /tem/el7/ResMap-1.1.4/ResMap-1.1.4-linux64
setenv     RELION_MOTIONCOR2_EXECUTABLE /tem/el7/MotionCor2_v1.3.1/MotionCor2_v1.3.1-
˓→Cuda92
setenv     RELION_UNBLUR_EXECUTABLE /tem/el7/unblur_1.0.2/bin/unblur_openmp_7_17_15.exe
setenv     RELION_SUMMOVIE_EXECUTABLE /tem/el7/summovie_1.0.2/bin/sum_movie_openmp_7_-
˓→17_15.exe
conflict    apps/relion
```

3. Load the environment module for the version of relion application which you want to execute. As the module specified is loaded, all the modules with dependency are also loaded (you can check these modules with “module list” command)

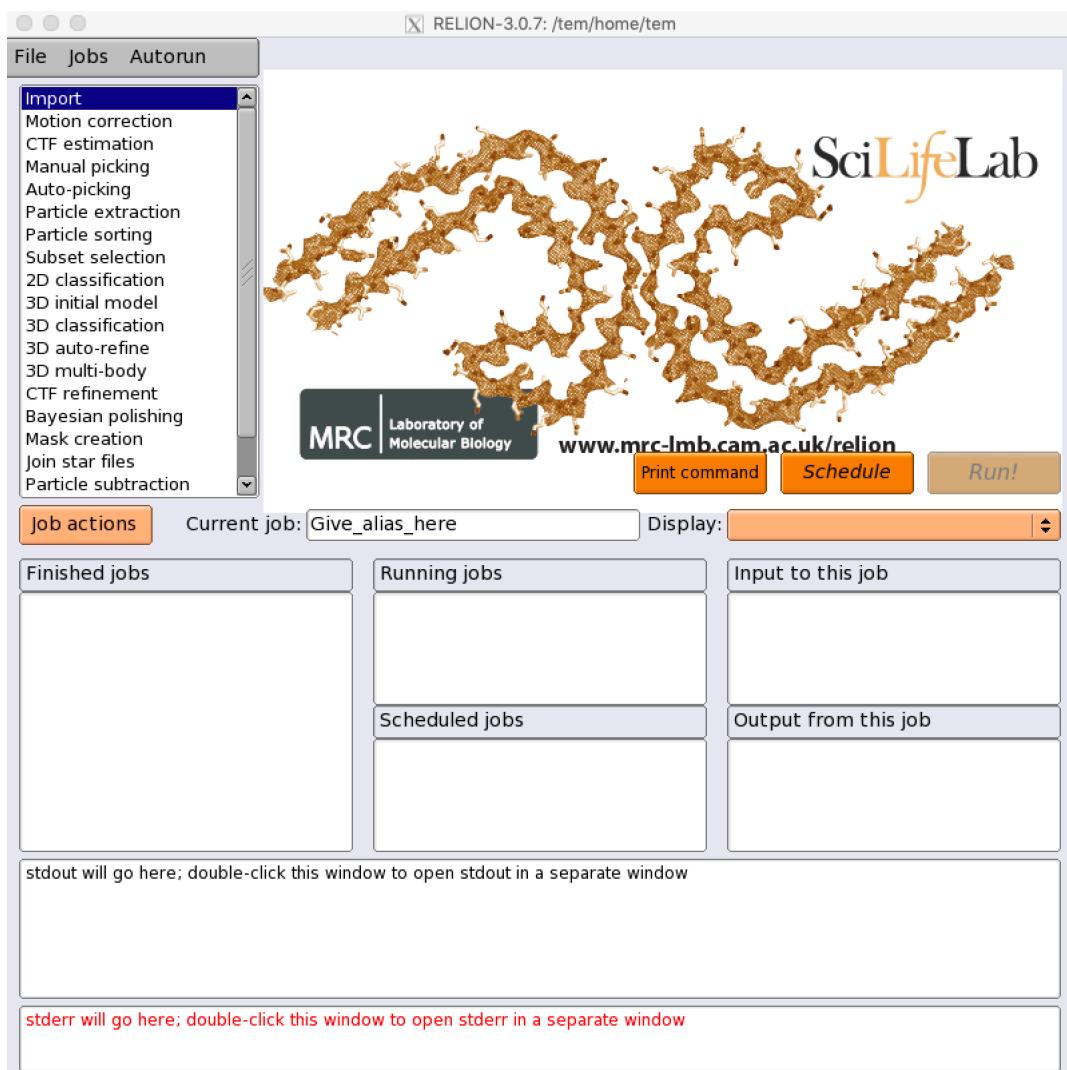
```
$> module load apps/relion/gpu/4.0.0
$> module list
Currently Loaded Modulefiles:
 1) mpi/gcc/openmpi/4.0.3  2) cuda/11.2  3) apps/relion/gpu/4.0.0
```

4. Check the relion application binary path

```
$> which relion
/tem/el7/relion-4.0.0/gpu/bin/relion
```

5. Execute the relion application (we assume that X11 forwarding is enabled)

```
$> relion
```



5.2 PBS Strings used in Relion

Table 1: torque_strings_of_relion

String	Variable type	Description
XXXcom-mandXXX	string	relion command + arguments
XXXqueueXXX	string	Name of the queue to submit job to
XXXmpin-odesXXX	integer	The number of MPI processes to use
XXXthread-sXXX	integer	The number of threads to use on each MPI process
XXXcoresXXX	integer	The number of MPI processes times the number of threads
XXXdedicatedXXX	integer	The minimum number of cores on each node (use this to fill entire nodes)
XXXnodesXXX	integer	The total number of nodes to be requested
XXXextra1XXX	string	Installation-specific
XXXextra2XXX	string	Installation-specific

Relion, by default, does not use the XXXextra1XXX, XXXextra2XXX, ... variables. They provide additional flexibility for queueing systems (like Torque) that require additional variables. They may be activated by first setting RELION_QSUB_EXTRA_COUNT to the number of fields you need (e.g. 3) and then setting the RELION_QSUB_EXTRA1, RELION_QSUB_EXTRA2, RELION_QSUB_EXTRA3 ... environment variables, respectively. This will result in extra input fields in the GUI, with the label text being equal to the value of the environment variable. Likewise, their default values (upon starting the GUI) can be set through environment variables RELION_QSUB_EXTRA1_DEFAULT, RELION_QSUB_EXTRA2_DEFAULT, etc and their help messages can be set through environmental variables RELION_QSUB_EXTRA1_HELP, RELION_QSUB_EXTRA2_HELP and so on.

5.3 Running data analysis jobs using CPU cores

5.3.1 Module path

Users should load an environment module, whose path is **apps/relion/cpu/X.X.X** (i.e., moulde load apps/relion/cpu/4.0.0) to execute data analysis jobs using CPU cores in relion GUI.

- apps/relion/cpu/3.0.7
- apps/relion/cpu/3.1.0
- apps/relion/cpu/4.0.0

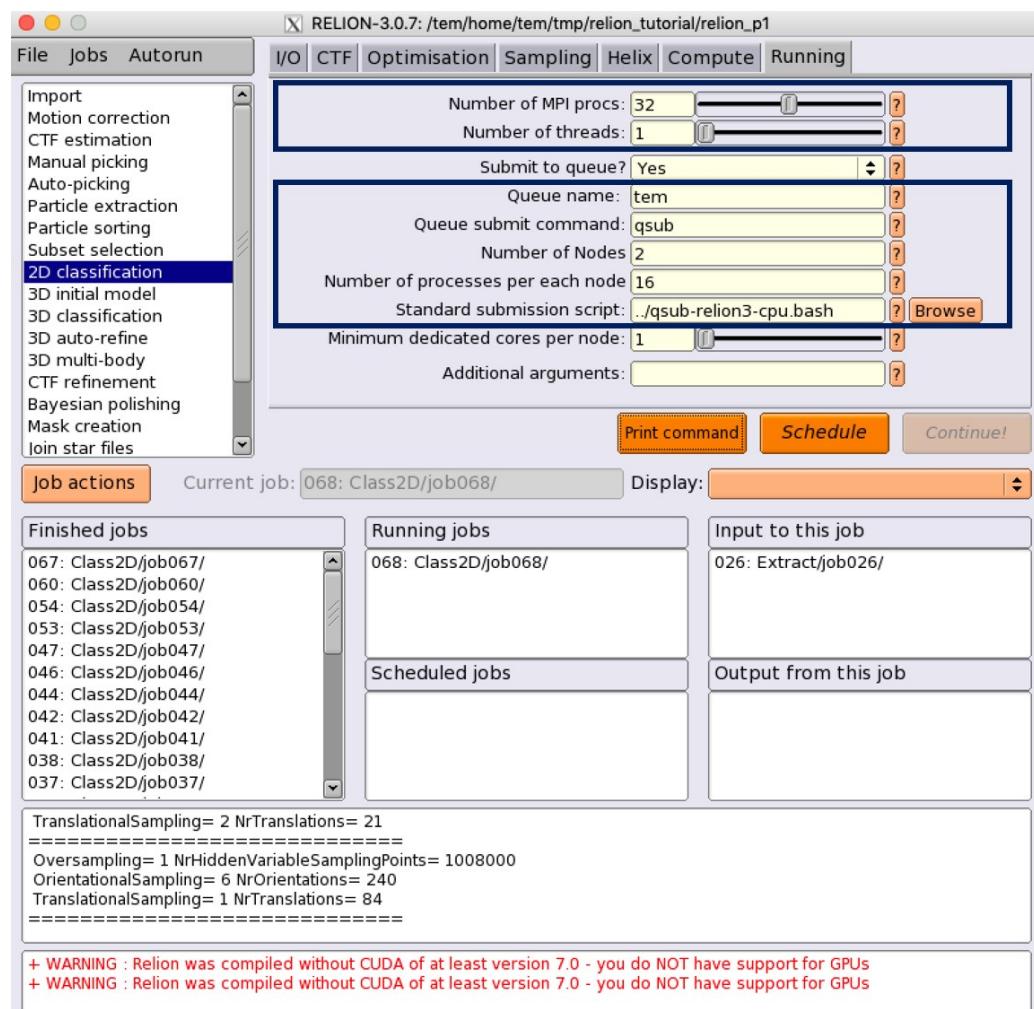
5.3.2 Environment variables

Relion defines a lot of environment variables that can be used to execute different types of subtasks in the analysis workflows. Among these, “RELION_QSUB_TEMPLATE” describes the location of a proper batch job script template (usually called standard job submission script) to submit jobs to the farm.

```
(for relion 3.0.7 standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-relion-3.0.7-cpu.  
↳bash  
(for relion 3.1.0 standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-relion-3.1.0-cpu.  
↳bash  
(for relion 4.0.0 standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-relion-4.0.0-cpu.  
↳bash
```

For the use of CPU cluster nodes, we have set the RELION_QSUB_EXTRA_COUNT to 2. Two extra options describe “Number of Nodes” and “Number of processes per each node”, respectively. These values can be referred by XXXextra1, XXXextra2XXX in the following batch job script template.

```
setenv RELION_QSUB_EXTRA_COUNT 2  
setenv RELION_QSUB_EXTRA1 "Number of Nodes"  
setenv RELION_QSUB_EXTRA2 "Number of processes per each node"  
setenv RELION_QSUB_EXTRA1_DEFAULT 2  
setenv RELION_QSUB_EXTRA2_DEFAULT 16
```



As shown in above figure, you can browse and select “**standard submission script**” as the location of RELION_QSUB_TEMPLATE for relion X.X.X (i.e., /tem/el7/qsub-relion-4.0.0-cpu.bash or its own your copy), and give “**Number of Nodes**” and “**Number of processes per each node**” values instead of default ones to submit a job to Torque based TEM farm.

Note: For CPU jobs, note that you **MUST** use **cpuQ** for the “Queue name” field and render correct “number of MPI procs” which is generally total number of processes (# of nodes * # of processes per each node)

5.3.3 Standard job submission script (for relion 4.0.0 CPU use)

```
#!/bin/bash

### Inherit all current environment variables
#PBS -V

### Job name
#PBS -N XXXnameXXX

### Queue name
```

(continues on next page)

(continued from previous page)

```
#PBS -q XXXqueueXXX

### CPU cluster use : Specify the number of nodes (XXXextra1XXX) and the number of processes per each node (XXXextra2XXX)
#PBS -l nodes=XXXextra1XXX:ppn=XXXextra2XXX:XXXqueueXXX

#PBS -o ${PBS_JOBNAME}/run.out
#PBS -e ${PBS_JOBNAME}/run.err

#####
### Print Environment Variables
#####

echo -----
echo -n 'Job is running on node ' ; cat $PBS_NODEFILE
echo -----
echo PBS: qsub is running on $PBS_O_HOST
echo PBS: originating queue is $PBS_O_QUEUE
echo PBS: executing queue is $PBS_QUEUE
echo PBS: working directory is $PBS_O_WORKDIR
echo PBS: execution mode is $PBS_ENVIRONMENT
echo PBS: job identifier is $PBS_JOBID
echo PBS: job name is $PBS_JOBNAME
echo PBS: node file is $PBS_NODEFILE
echo PBS: current home directory is $PBS_O_HOME
echo PBS: PATH = $PBS_O_PATH
echo -----


#####
# Switch to the working directory;
cd ${PBS_O_WORKDIR}/${PBS_JOBNAME}
touch run.out
touch run.err
cd $PBS_O_WORKDIR
#####


### Run:
module load apps/relion/cpu/4.0.0
mpirun --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --prefix /tem/el7/openmpi-4.0.3 -machinefile
$PBS_NODEFILE XXXcommandXXX

echo "Done!"
```

5.4 Running data analysis jobs using GPGPUs

5.4.1 Environment variables

Relion defines a lot of environment variables that can be used to execute different types of subtasks in the analysis workflows. Among these, “RELION_QSUB_TEMPLATE” describes the location of a proper batch job script to submit jobs to the farm.

(**for** relion 3.0.7 w/ GPU support standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-relion-3.0.7-gpu.bash

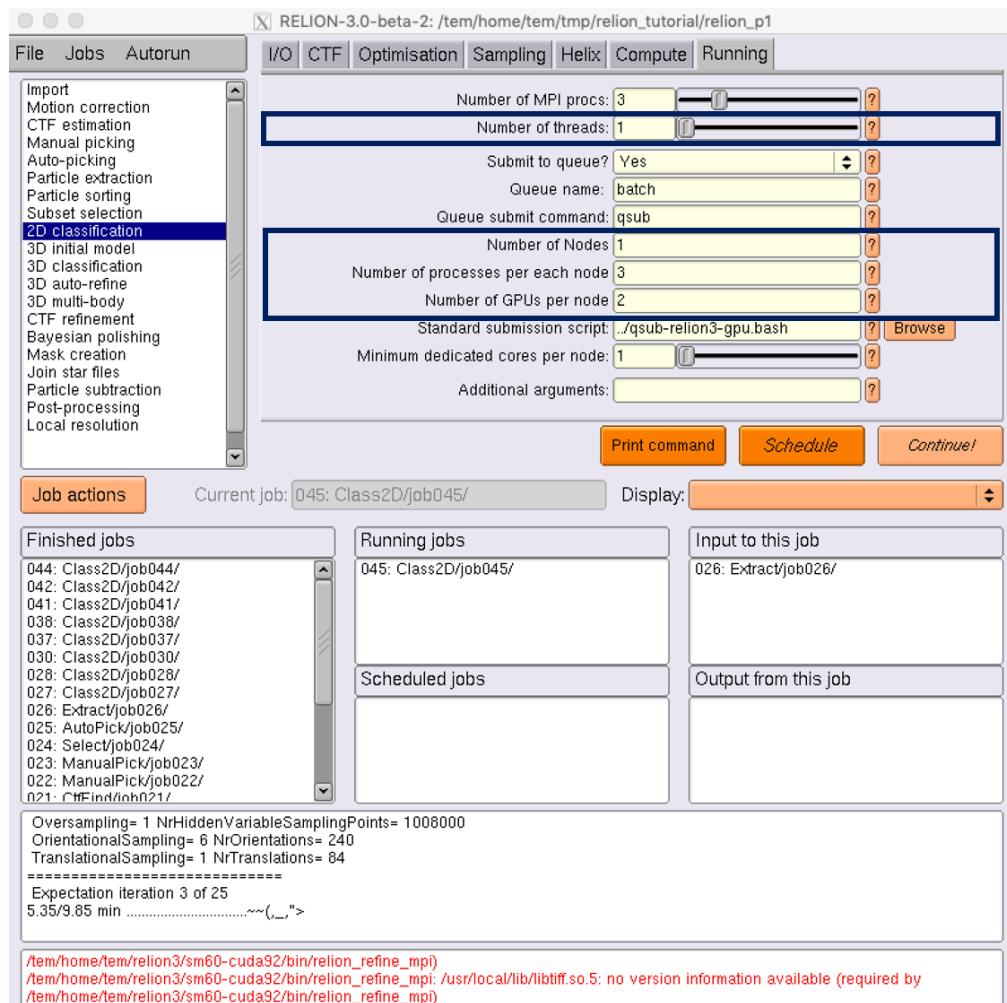
(continues on next page)

(continued from previous page)

(for relion 3.1.0 w/ GPU support standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-
→relion-3.1.0-gpu.bash
(for relion 4.0.0 w/ GPU support standard job submission script) RELION_QSUB_TEMPLATE /tem/el7/qsub-
→relion-4.0.0-gpu.bash

Unlike CPU cluster use case, we have set the RELION_QSUB_EXTRA_COUNT to 3 for the use of GPGPU cluster, where each extra option describes “Number of Nodes”, “Number of processes per each node”, and “Number of GPUs per node”, respectively. All these values can be accessed by XXXextra1, XXXextra2XXX, XXXextra3XXX in the batch job script template.

```
setenv RELION_QSUB_EXTRA_COUNT 3
setenv RELION_QSUB_EXTRA1 "Number of Nodes"
setenv RELION_QSUB_EXTRA2 "Number of processes per each node"
setenv RELION_QSUB_EXTRA3 "Number of GPUs per node"
setenv RELION_QSUB_EXTRA1_DEFAULT 1
setenv RELION_QSUB_EXTRA2_DEFAULT 3
setenv RELION_QSUB_EXTRA3_DEFAULT 2
```



Note: For GPU jobs, note that you **MUST** use **gpuQ** for the “Queue name” field and render correct “number of MPI procs” which is generally total number of processes (# of nodes * # of processes per each node)

5.4.2 Standard job submission script (for relion 4.0.0 GPGPU use)

```
#!/bin/bash

### Inherit all current environment variables
#PBS -V

### Job name
#PBS -N XXXnameXXX

### Queue name
#PBS -q XXXqueueXXX

### GPU use : Specify the number of nodes (XXXextra1XXX), the number of processes per each node (XXXextra2XXX), and the number of GPGPUs per node (XXXextra3XXX)
#PBS -l nodes=XXXextra1XXX:ppn=XXXextra2XXX:gpus=XXXextra3XXX:XXXqueueXXX

#PBS -o ${PBS_JOBNAME}/run.out
#PBS -e ${PBS_JOBNAME}/run.err

#####
### Print Environment Variables
#####

echo -----
echo -n 'Job is running on node '; cat $PBS_NODEFILE
echo -----
echo PBS: qsub is running on $PBS_O_HOST
echo PBS: originating queue is $PBS_O_QUEUE
echo PBS: executing queue is $PBS_QUEUE
echo PBS: working directory is $PBS_O_WORKDIR
echo PBS: execution mode is $PBS_ENVIRONMENT
echo PBS: job identifier is $PBS_JOBID
echo PBS: job name is $PBS_JOBNAME
echo PBS: node file is $PBS_NODEFILE
echo PBS: current home directory is $PBS_O_HOME
echo PBS: PATH = $PBS_O_PATH
echo PBS: PBS_GPUFILE=$PBS_GPUFILE
echo PBS: CUDA_VISIBLE_DEVICES=$CUDA_VISIBLE_DEVICES
echo -----

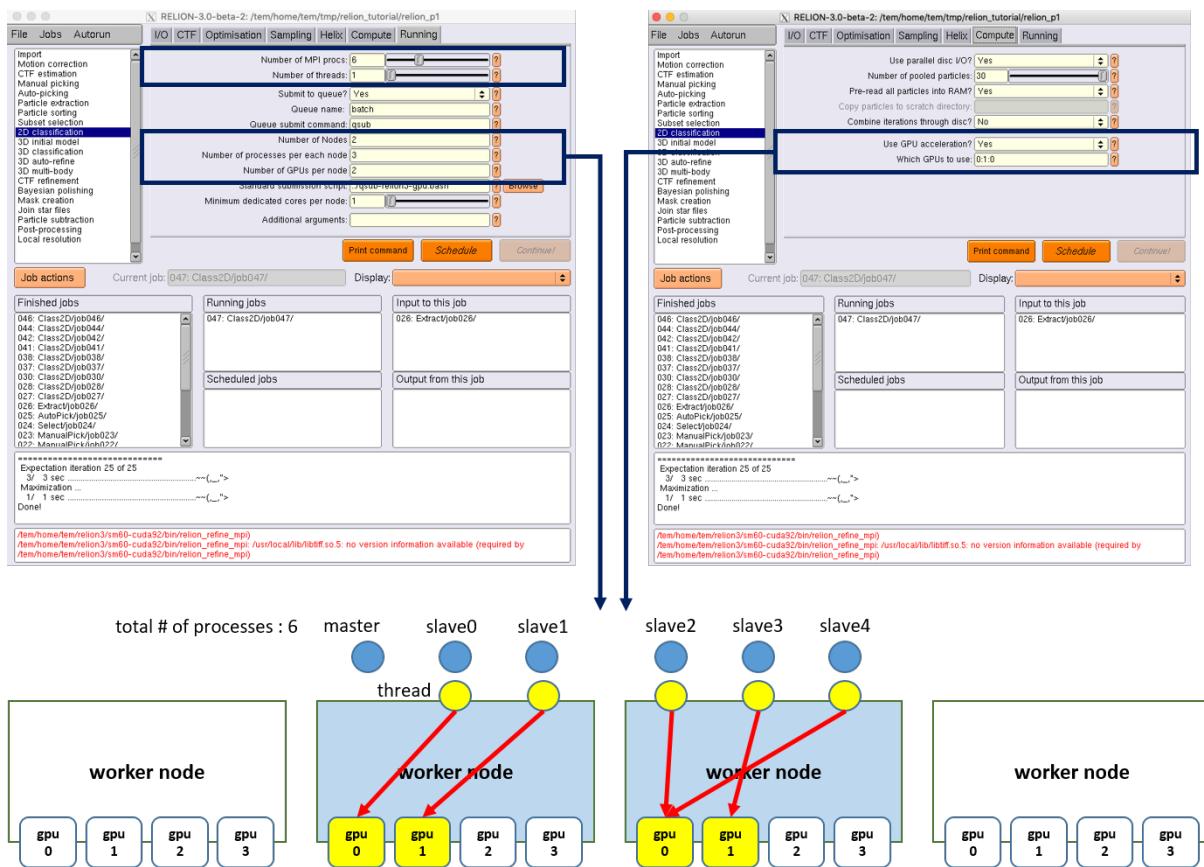


### Switch to the working directory;
cd ${PBS_O_WORKDIR}/${PBS_JOBNAME}
touch run.out
touch run.err
cd $PBS_O_WORKDIR
#####

### Run:
module load apps/relion/gpu/4.0.0
mpirun --mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --prefix /tem/el7/openmpi-4.0.3 -machinefile
$PBS_NODEFILE XXXcommandXXX

echo "Done!"
```

5.4.3 Specifying which GPGPUs to use



Here, we describe more advanced syntax for restricting RELION processes to certain GPUs on multi-GPU setups. You can use an argument to the `--gpu` option to provide a list of device-indices. The syntax is then to delimit ranks with colons [:], and threads by commas [,]. Any GPU indices provided is taken to be a list which is repeated if shorter than the total number of GPUs. By extension, the following rules applies

If a GPU id is specified more than once for a single mpi-rank, that GPU will be assigned proportionally more of the threads of that rank. If no colons are used (i.e. GPUs are only specified for a single rank), then the GPUs specified, apply to all ranks. If GPUs are specified for more than one rank but not for all ranks, the unrestricted ranks are assigned the same GPUs as the restricted ranks, by a modulo rule. For example, if you would only want to use two of the four GPUs for all mpi-ranks, because you want to leave another two free for a different user/job, then (by the above rule 2) you can specify

```
mpirun -n 3 'which relion_refine_mpi' --gpu 2:3
slave 1 is told to use GPU2. slave 2 is told to use GPU3.
```

If you want an even spread over ALL GPUs, then you should not specify selection indices, as RELION will handle this itself. On your hypothetical 4-GPU machine, you would simply say

```
mpirun -n 3 'which relion_refine_mpi' --gpu
## slave 1 will use GPU0 and GPU1 for its threads. slave 2 will use GPU2 and GPU3 for its threads
```

One can also schedule individual threads from MPI processes on the GPUs. This would be most useful when available RAM would be a limitation. Then one could for example run 3 MPI processes, each of which spawn a number of threads on two of the cards each, as follows:

```
mpirun -n 3 'which relion_refine_mpi' --j 4 --gpu 0,1,1,2,3
## slave 1 is told to put thread 1 on GPU0, threads 2 and 3 on GPU1, and thread 4 on GPU2. slave 2 is told
→ to put all 4 threads on GPU3.
```

Finally, for completeness, the following is a more complex example to illustrate the full functionality of the GPU-device specification options.

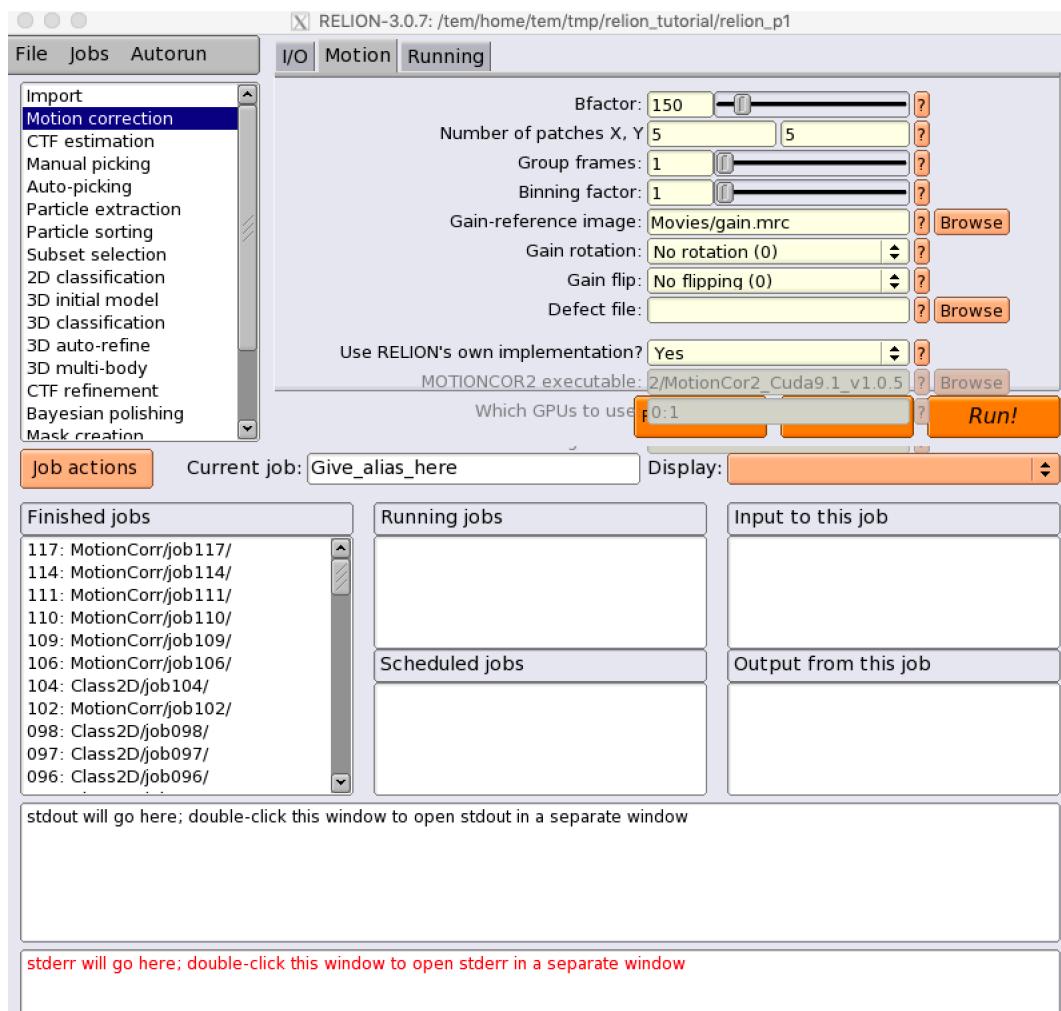
```
mpirun -n 4 ... -j 3 --gpu 2:2:1,3
## slave 1 w/ 3 threads on GPU2, slave 2 w/ 3 threads on GPU2, slave 3 distributes 3 threads as evenly as
→ possible across GPU1 and GPU3.
```

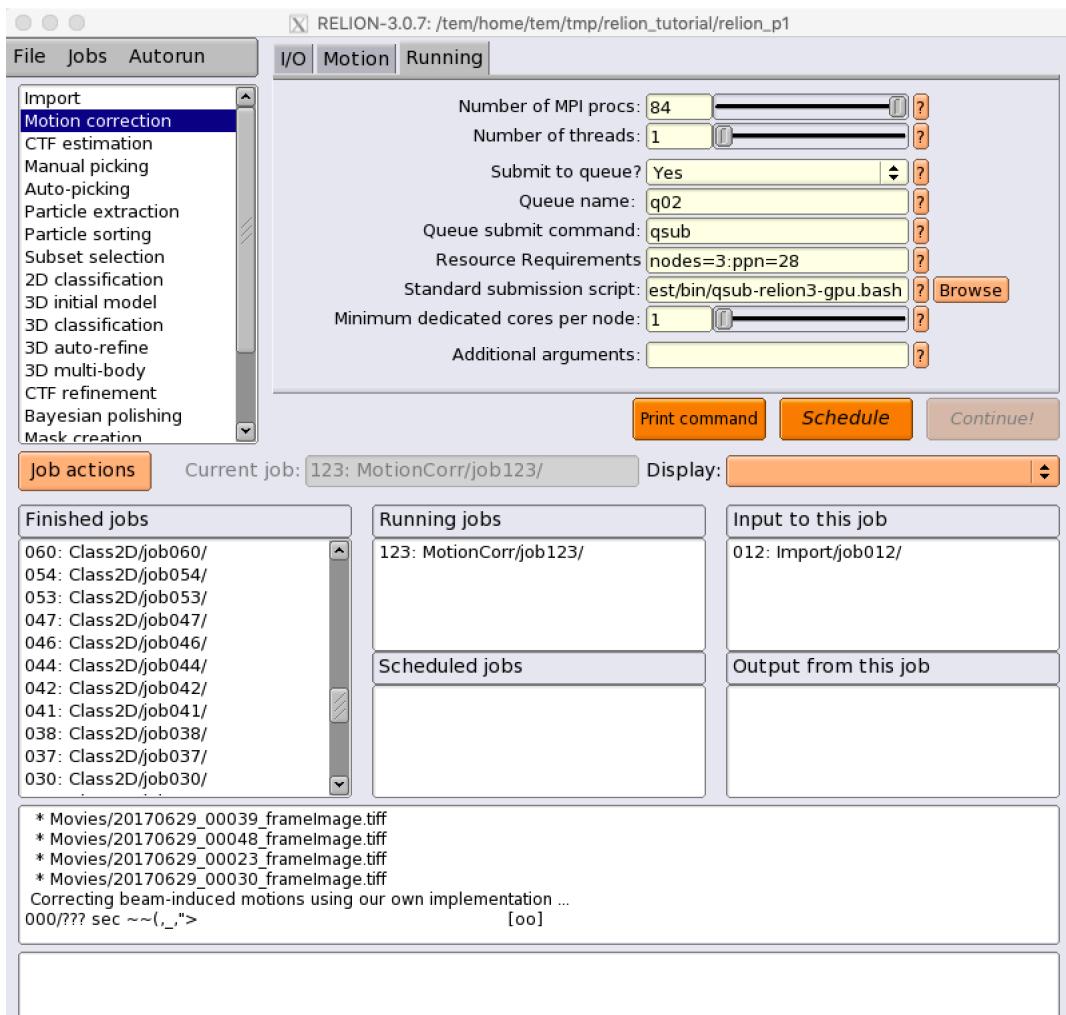
For more information, please refer to Relion Benchmarks and computer hardware (https://www3.mrc-lmb.cam.ac.uk/relion/index.php/Benchmarks_%26_computer_hardware)

5.5 Examples

5.5.1 Motion Correction

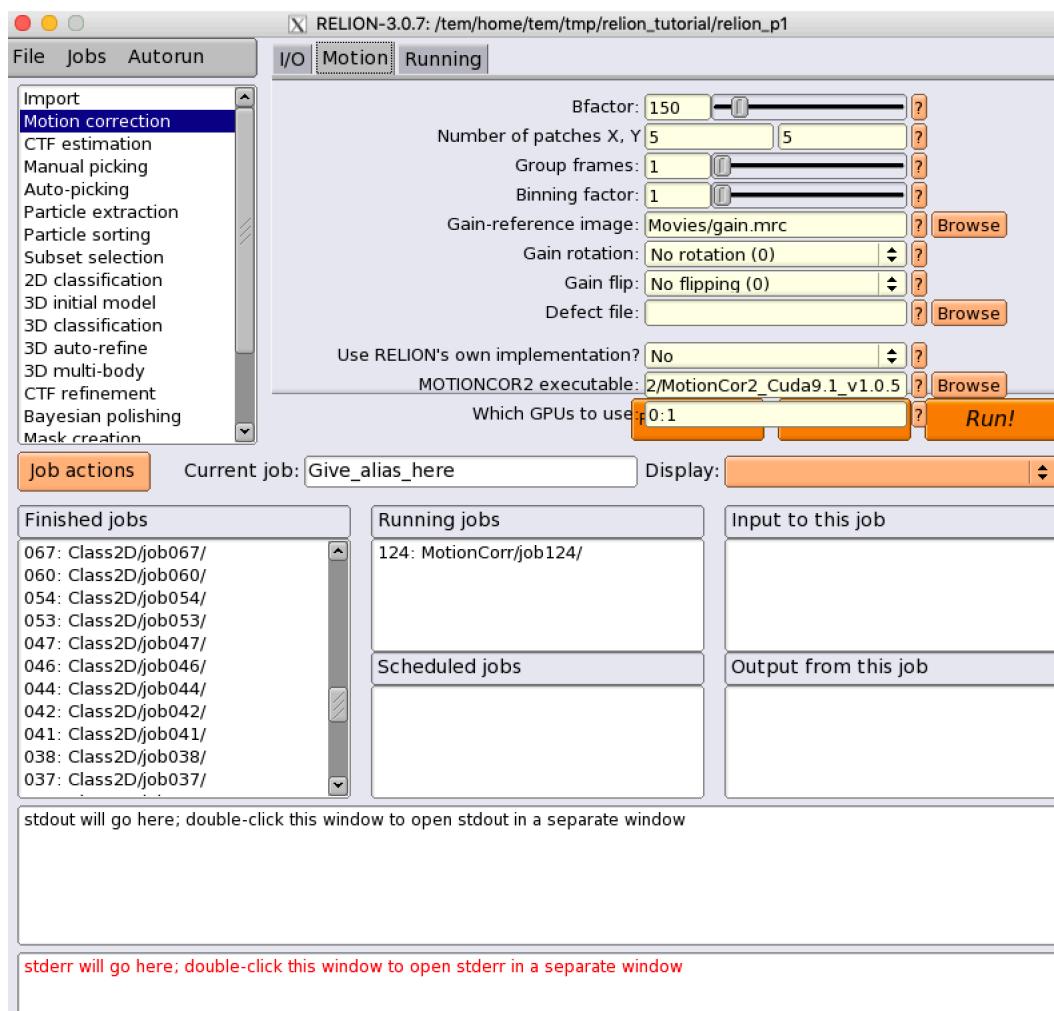
- **MotionCor2-like alignment algorithm** (CPU-only job, relion-own implementation)
 - (Motion) Use RELION's own implementation? : Yes
 - (Running) Number of MPI Procs : 84
 - (Running) Number of threads : 1
 - (Running) Queue name : **cpuQ**
 - (Running) Resource Requirements : nodes=3:ppn=28 (e.g., we assume that the job is allocated to the 3 nodes which have all 28 cores available for each node)
 - (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-cpu.bash ## Relion 3.0.7 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-cpu.bash ## Relion 3.1.0 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-4.0.0-cpu.bash ## Relion 4.0.0 CPU MPI 작업 템플릿

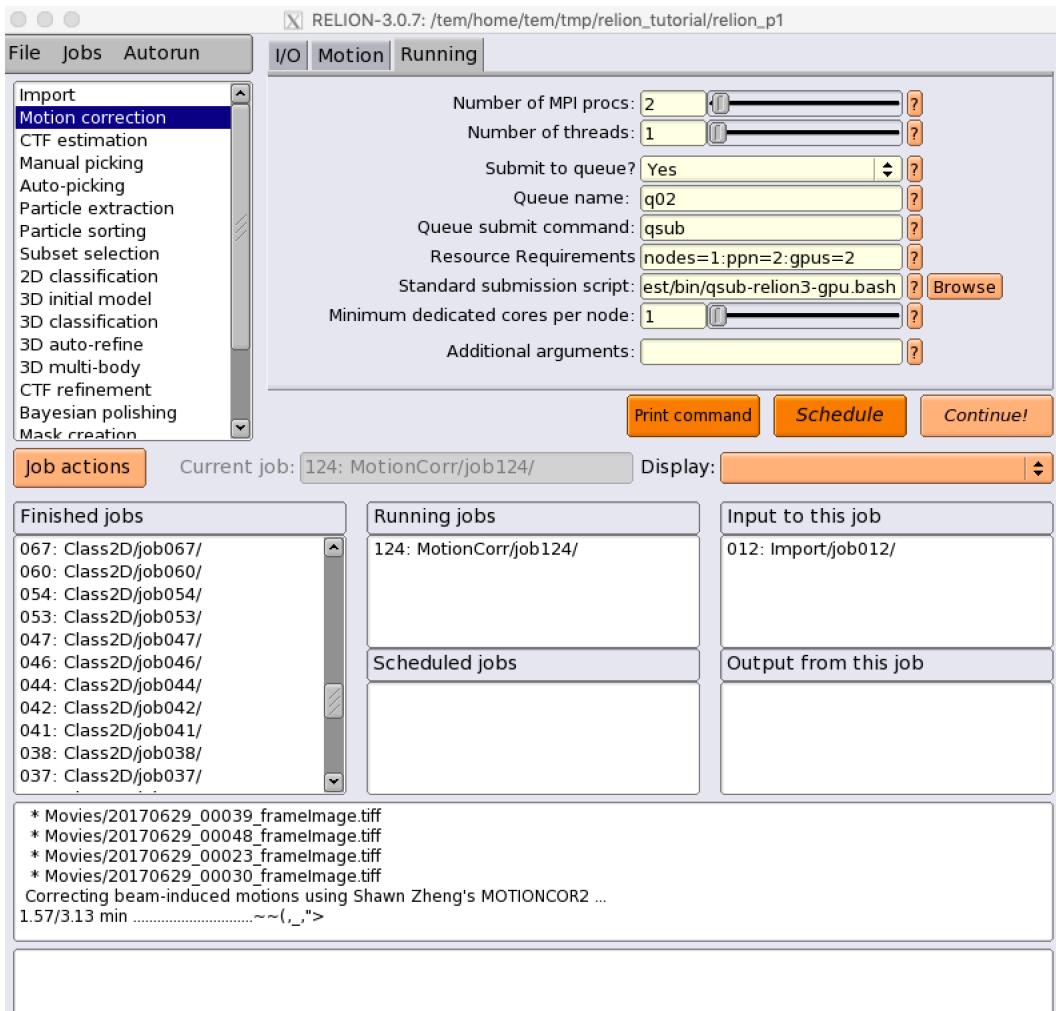




- **MotionCor2** (GPU-accelerated job)

- (Motion) Use RELION's own implementation? : No
- (Motion) MOTIONCOR2 executable : /tem/home/tem/_Applications/MotionCor2/MotionCor2_Cuda9.1_v1.0.5
- (Running) Number of MPI Procs : 2
- (Running) Number of threads : 1
- (Running) Queue name : **gpuQ**
- (Running) Resource Requirements : nodes=1:ppn=3:gpus=2 (e.g., we assume that the job is allocated to 1 node which has 3 cpu cores and 2 GPU devices available)
- (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-gpu.bash ## Relion 3.0.7 GPU 가속 활용하는 MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-gpu.bash ## Relion 3.1.0 GPU 가속 활용하는 MPI 작업 템플릿
 - * /tem/el7/qsub-relion-4.0.0-gpu.bash ## Relion 4.0.0 GPU 가속 활용하는 MPI 작업 템플릿

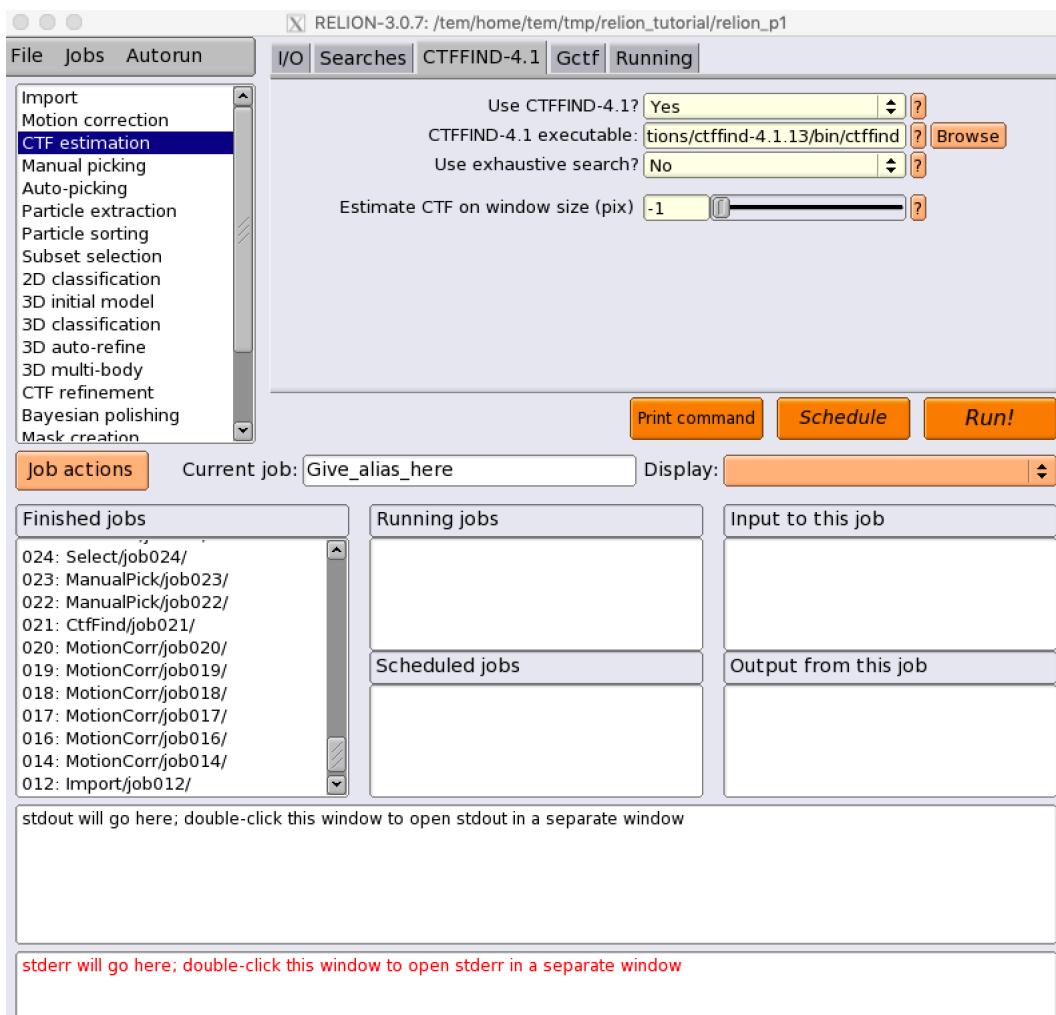


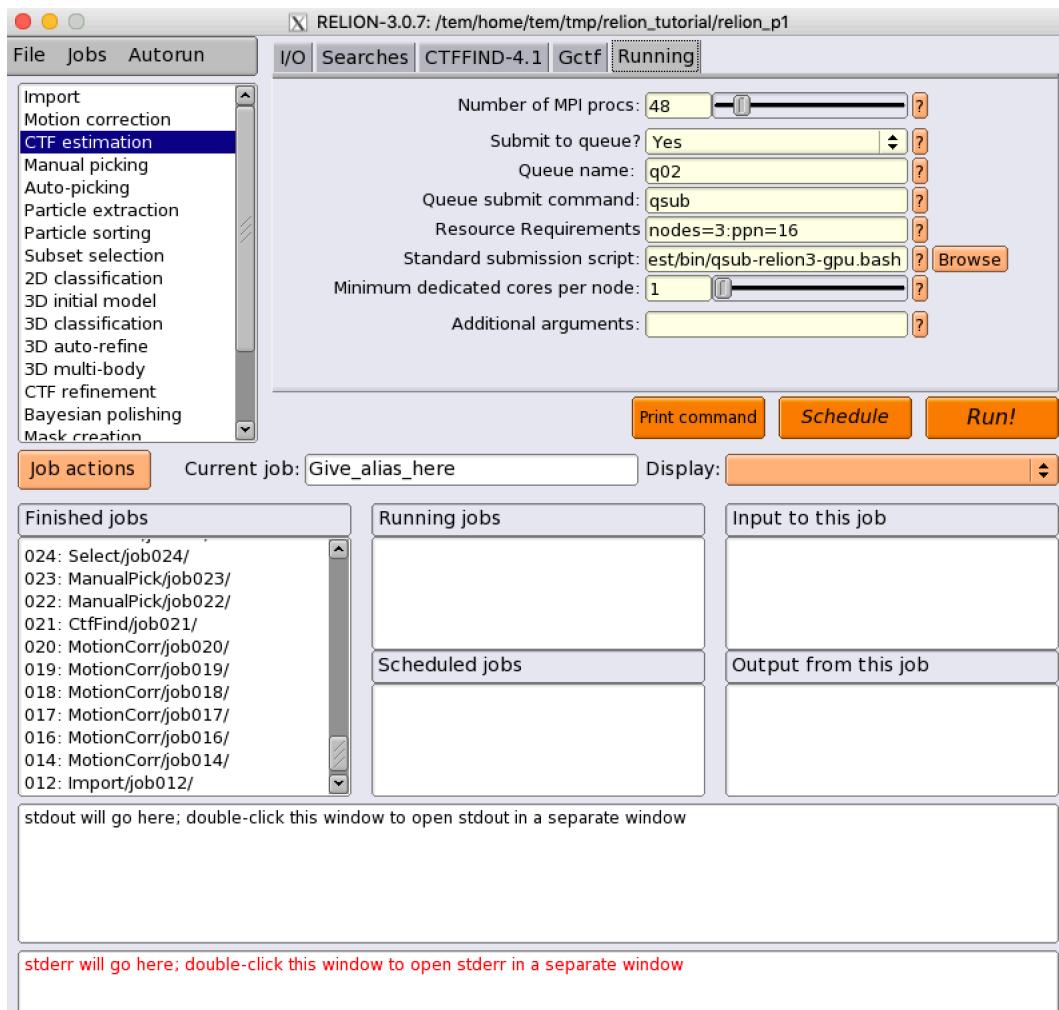


5.5.2 CTF Estimation

- **CTFFIND-4.1** (CPU-only job)
 - (CTFFIND-4.1) Use CTFFIND-4.1? : Yes
 - (CTFFIND-4.1) CTFFIND-4.1 executable? : /tem/el7/ctffind-4.1.14/bin/ctffind
 - (Gctf) Use Gctf instead? : No
 - (Running) Number of MPI procs: 48
 - (Running) Submit to queue? : Yes
 - (Running) Queue name : **cputQ**
 - (Running) Resource Requirements : nodes=3:ppn=16 (e.g., we assume the use of 3 nodes, 16 cpu cores per each node)
 - (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-cpu.bash ## Relion 3.0.7 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-cpu.bash ## Relion 3.1.0 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-4.0.0-cpu.bash ## Relion 4.0.0 CPU MPI 작업 템플릿

* /tem/el7/qsub-relion-4.0.0-cpu.bash ## Relion 4.0.0 CPU MPI 작업 템플릿

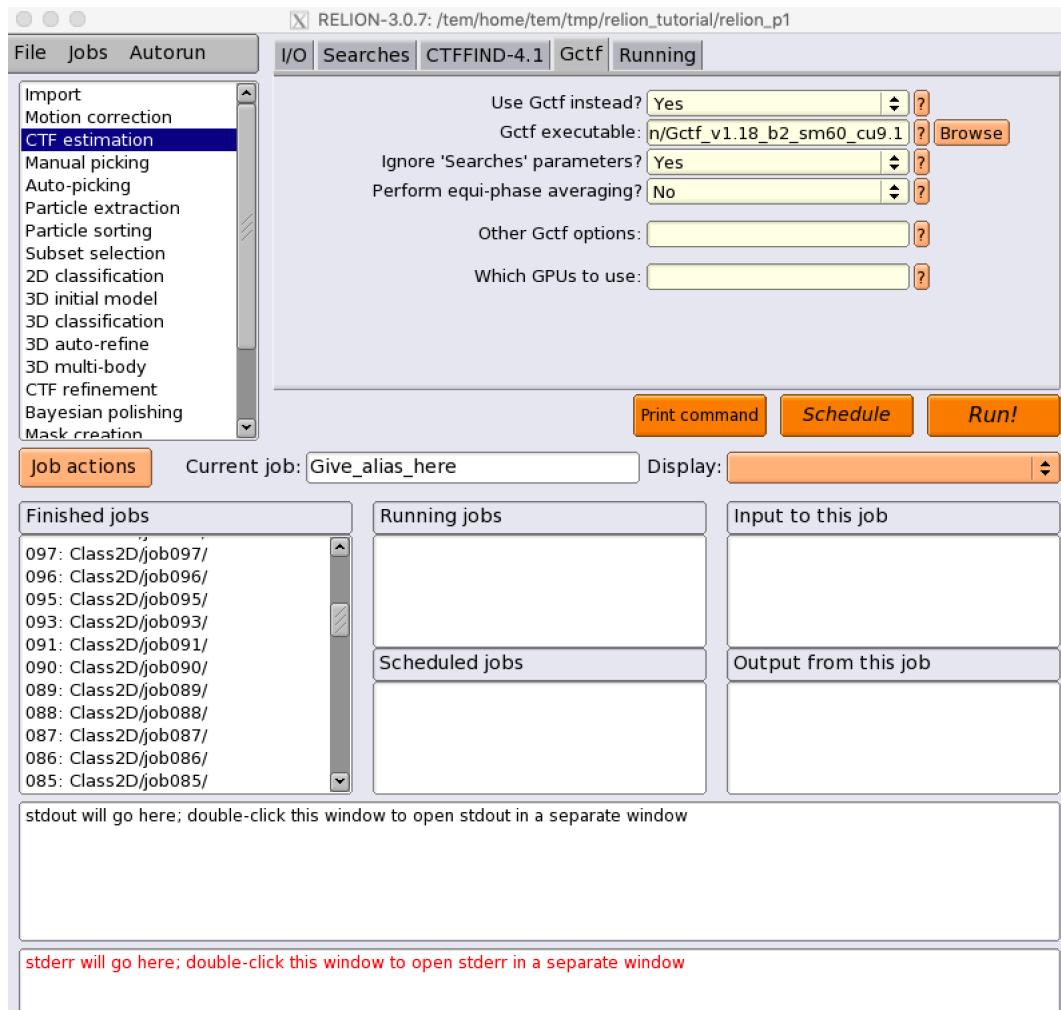


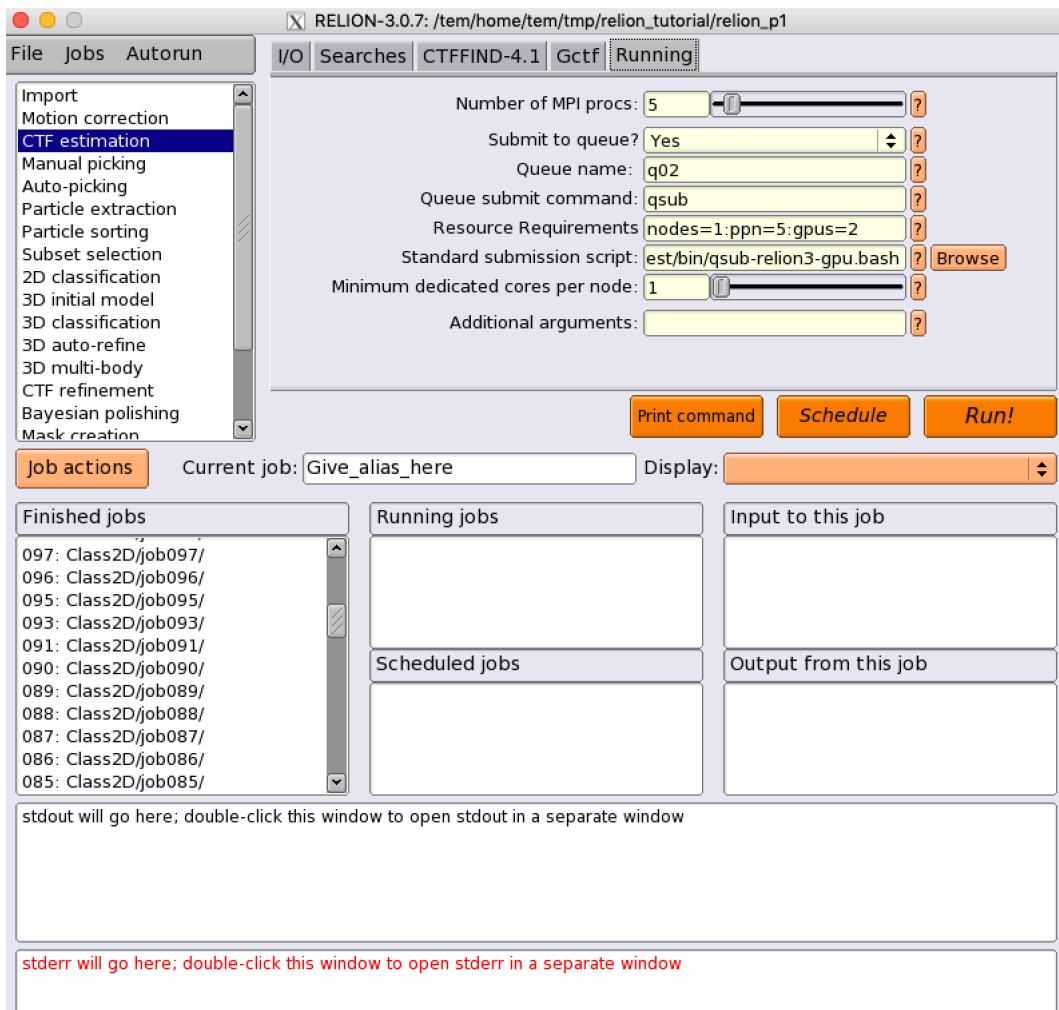


- **Gctf** (GPU-accelerated job)

- (CTFFIND-4.1) Use CTFFIND-4.1? : No
- (Gctf) Use Gctf instead? : Yes
- (Gctf) Gctf executable: /tem/el7/Gctf_v1.18_b2/bin/Gctf_v1.18_b2_sm60_cu9.2
- (Gctf) Which GPUs to use: <empty> (i.e., relion automatically assigned available GPU devices to the MPI processes)
- (Running) Number of MPI procs: 5 (1 master and 4 slave processes)
- (Running) Submit to queue? : Yes
- (Running) Queue name : **gpuQ**
- (Running) Resource Requirements : nodes=1:ppn=5:gpus=2
- (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-gpu.bash ## Relion 3.0.7 GPU 가속 활용하는 MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-gpu.bash ## Relion 3.1.0 GPU 가속 활용하는 MPI 작업 템플릿

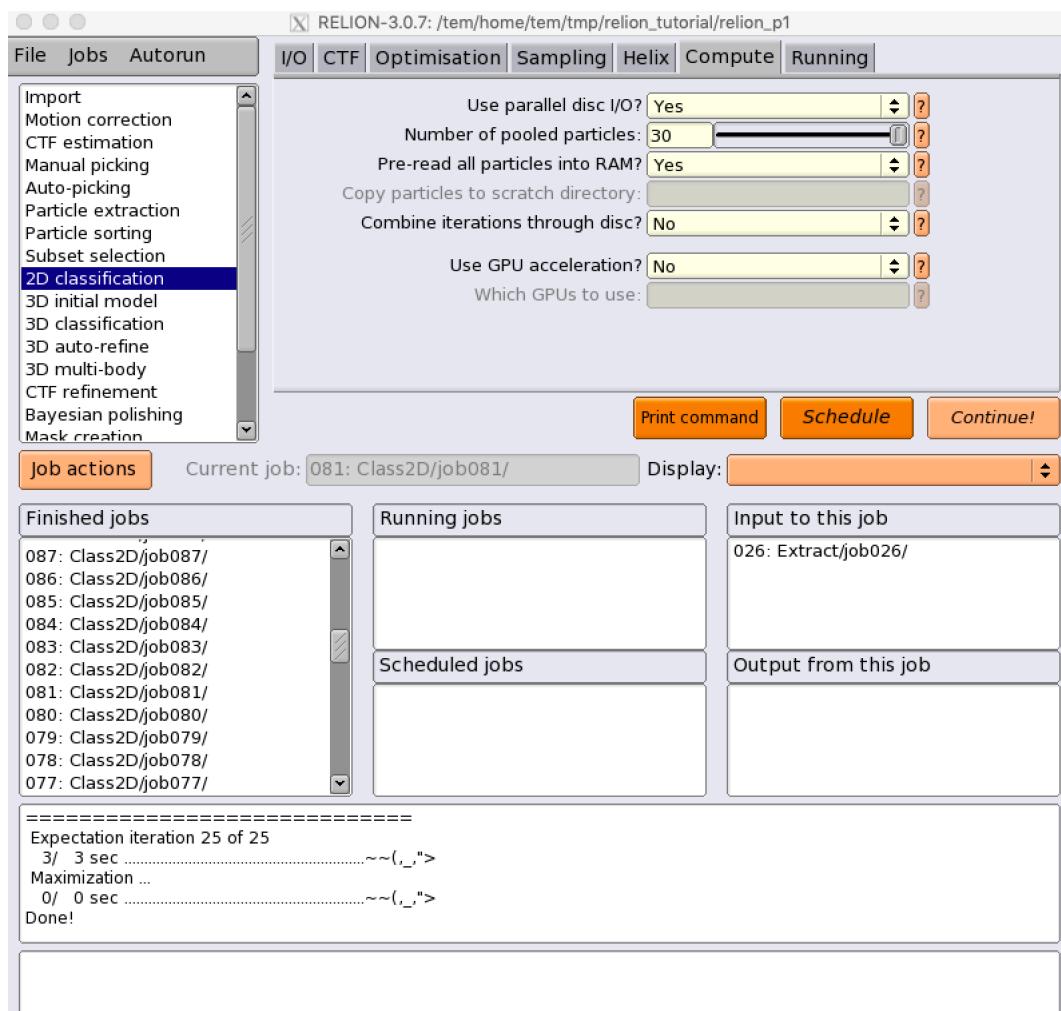
* /tem/el7/qsub-relion-4.0.0-gpu.bash ## Relion 4.0.0 GPU 가속 활용하는 MPI 작업 템플릿

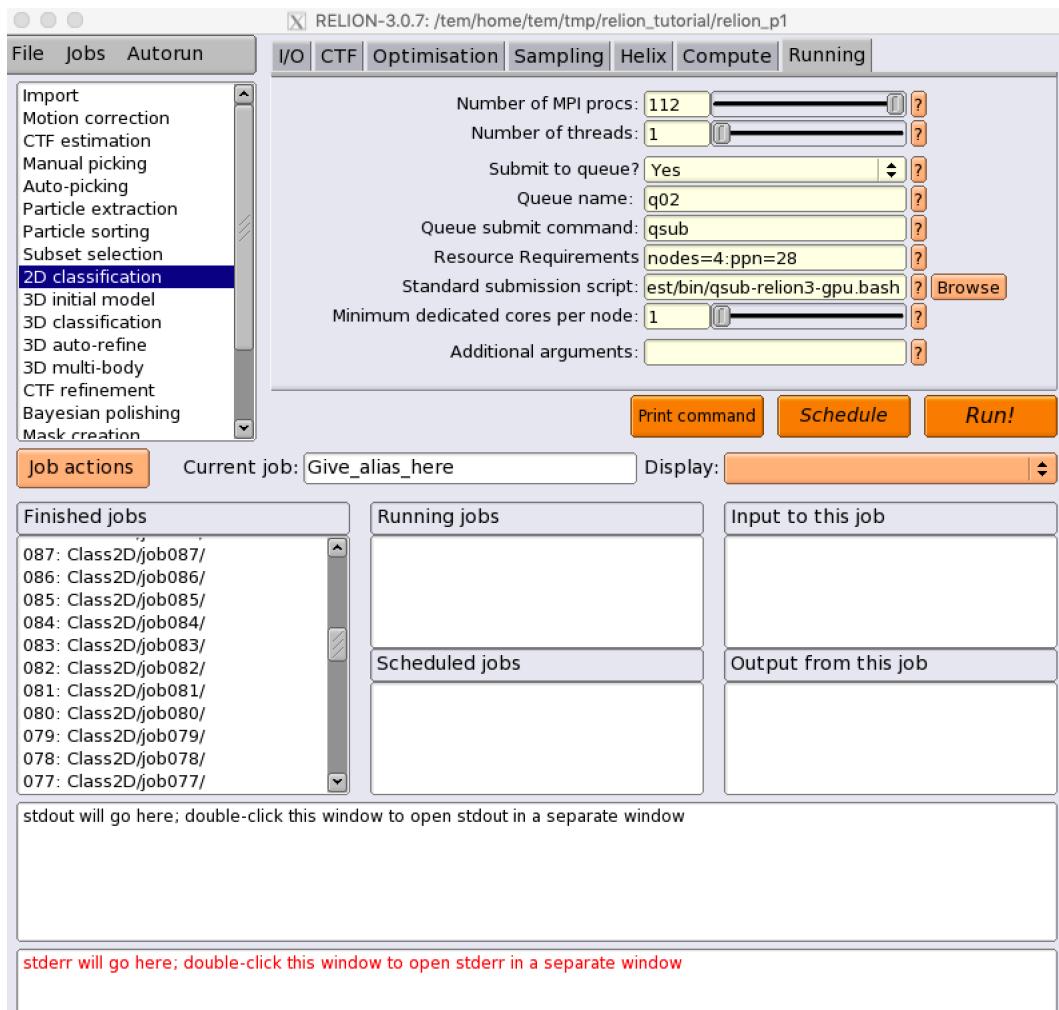




5.5.3 2D Classification

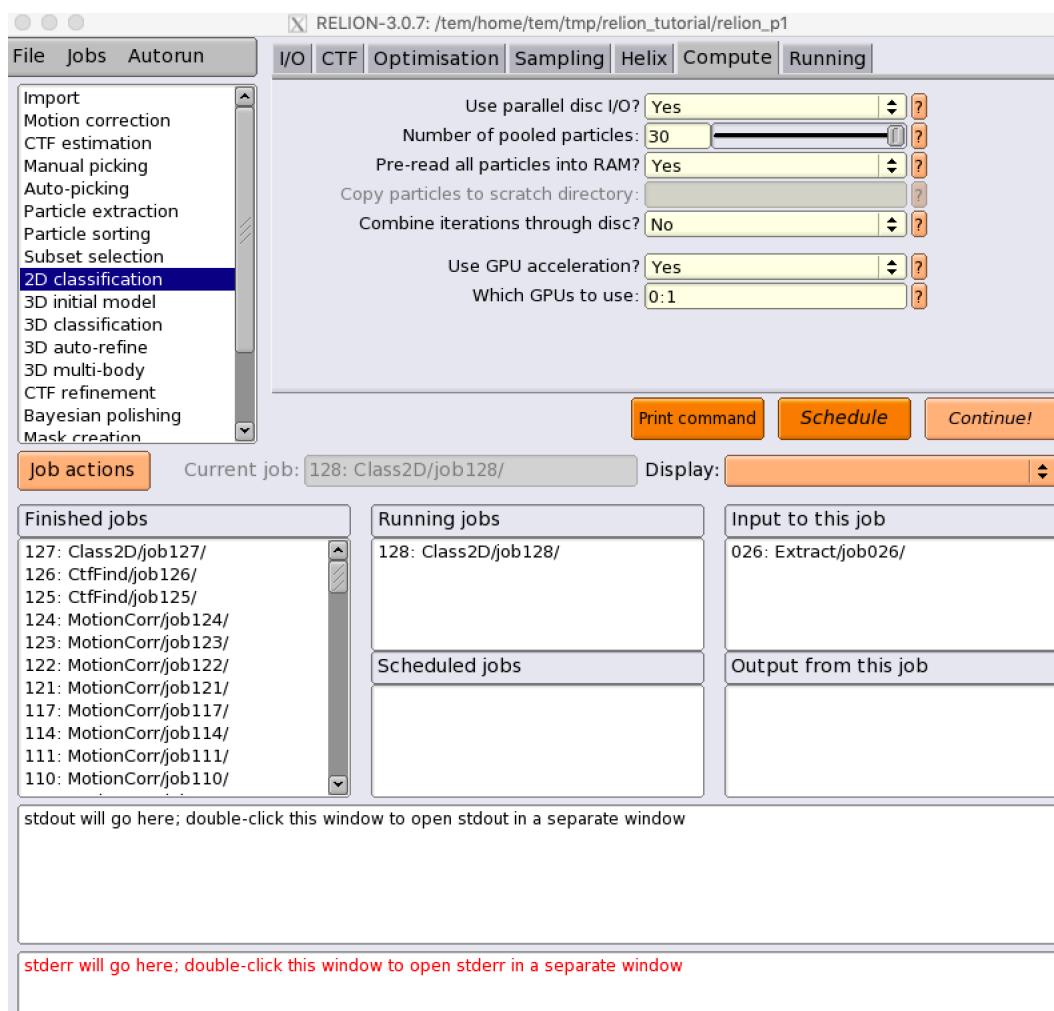
- **relion_refine_mpi** (CPU-only job)
 - (Compute) Use GPU acceleration? : No
 - (Running) Number of MPI procs: 112
 - (Running) Number of threads: 1
 - (Running) Submit to queue? : Yes
 - (Running) Queue name : **cpuQ**
 - (Running) Resource Requirements : nodes=4:ppn=28 (e.g., we assume the use of 4 nodes, 28 cpu cores per each node)
 - (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-cpu.bash ## Relion 3.0.7 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-cpu.bash ## Relion 3.1.0 CPU MPI 작업 템플릿
 - * /tem/el7/qsub-relion-4.0.0-cpu.bash ## Relion 4.0.0 CPU MPI 작업 템플릿

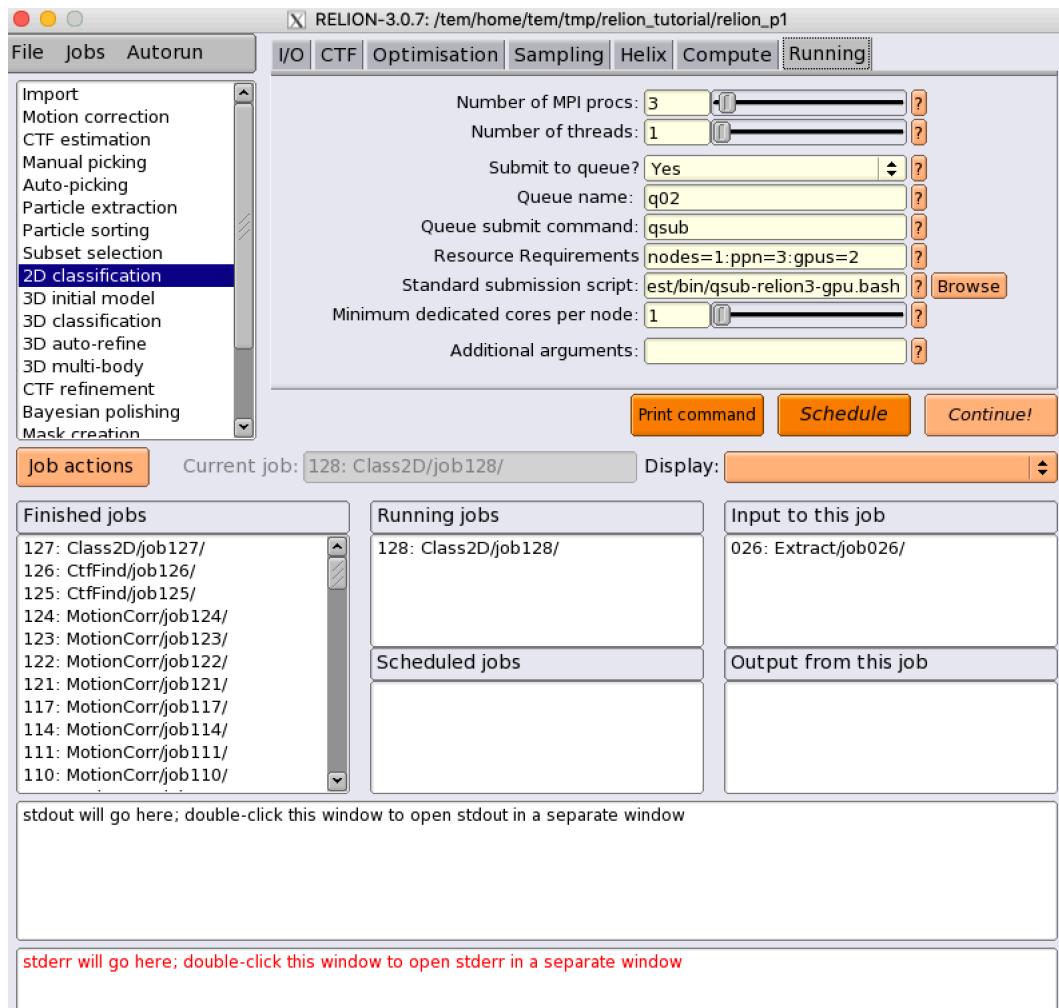




- **relion_refine_mpi** (GPU-accelerated job)

- (Compute) Use GPU acceleration? : Yes
- (Compute) Which GPUs to use? : 0:1 (i.e., we will assign each slave process to GPU device index 0 and 1, respectively)
- (Running) Number of MPI procs: 3 (1 master and 2 slave processes)
- (Running) Number of threads: 1
- (Running) Submit to queue? : Yes
- (Running) Queue name : **gpuQ**
- (Running) Resource Requirements : nodes=1:ppn=3:gpus=2
- (Running) Standard submission script :
 - * /tem/el7/qsub-relion-3.0.7-gpu.bash ## Relion 3.0.7 GPU 가속 활용하는 MPI 작업 템플릿
 - * /tem/el7/qsub-relion-3.1.0-gpu.bash ## Relion 3.1.0 GPU 가속 활용하는 MPI 작업 템플릿
 - * /tem/el7/qsub-relion-4.0.0-gpu.bash ## Relion 4.0.0 GPU 가속 활용하는 MPI 작업 템플릿





**CHAPTER
SIX**

CISTEM

cisTEM is user-friendly software to process cryo-EM images of macromolecular complexes and obtain high-resolution 3D reconstructions from them. It comprises a number of tools to process image data including movies, micrographs and stacks of single-particle images, implementing a complete “pipeline” of processing steps to obtain high-resolution single-particle reconstructions. (from cisTEM official site <https://cistem.org>)

6.1 Executing cisTEM

6.1.1 How to start cisTEM data analysis tool

1. You can find out cisTEM applications’ environment module path by listing all the module available on TEM service farm.

```
$> module avail

----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0
apps/relion/cpu/3.0.7
apps/relion/cpu/3.1.0
apps/relion/cpu/4.0.0
apps/relion/gpu/3.0.7
apps/relion/gpu/3.1.0

---- /tem/el7/Modules/acceleration ---
cuda/9.2 cuda/11.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/4.8.5/openmpi/4.0.3
mpi/gcc/8.3.1/mpich/3.4.3
mpi/gcc/8.3.1/openmpi/4.0.3
mpi/gcc/openmpi/4.0.3

----- /tem/el7/Modules/virtualenv -----
conda/2020.11 topaz/cuda-9.2/0.2.4
pyem/0.5    topaz/cuda-11.0/0.2.4

----- /tem/el7/Modules/tools -----
tools/aspera-cli/3.9.6
tools/ctffind/4.1.14
tools/gctf/1.18_b2
```

(continues on next page)

(continued from previous page)

```
tools/motioncor2/1.3.1
tools/resmap/1.1.4
tools/summovie/1.0.2
tools/unblur/1.0.2

----- /tem/el7/Modules/experiment -----
PyRosetta/4
python/3.7
rosetta/mpich-3.4.3/3.13
rosetta/openmpi-4.0.3/3.13
```

2. Check the module details for cisTEM application

```
$> module show apps/cistem/1.0.0

-----
/tem/el7/Modules/apps/apps/cistem/1.0.0:

module-whatis {Setups cistem 1.0.0 environment variables}
module      load mpi/gcc/openmpi/4.0.3
prepend-path PATH /tem/el7/cistem-1.0.0-beta
conflict    apps/cistem
-----
```

3. Load the environment module for cisTEM application which you want to execute. As the module specified is loaded, all the modules with dependency are also loaded (you can check these modules with “module list” command)

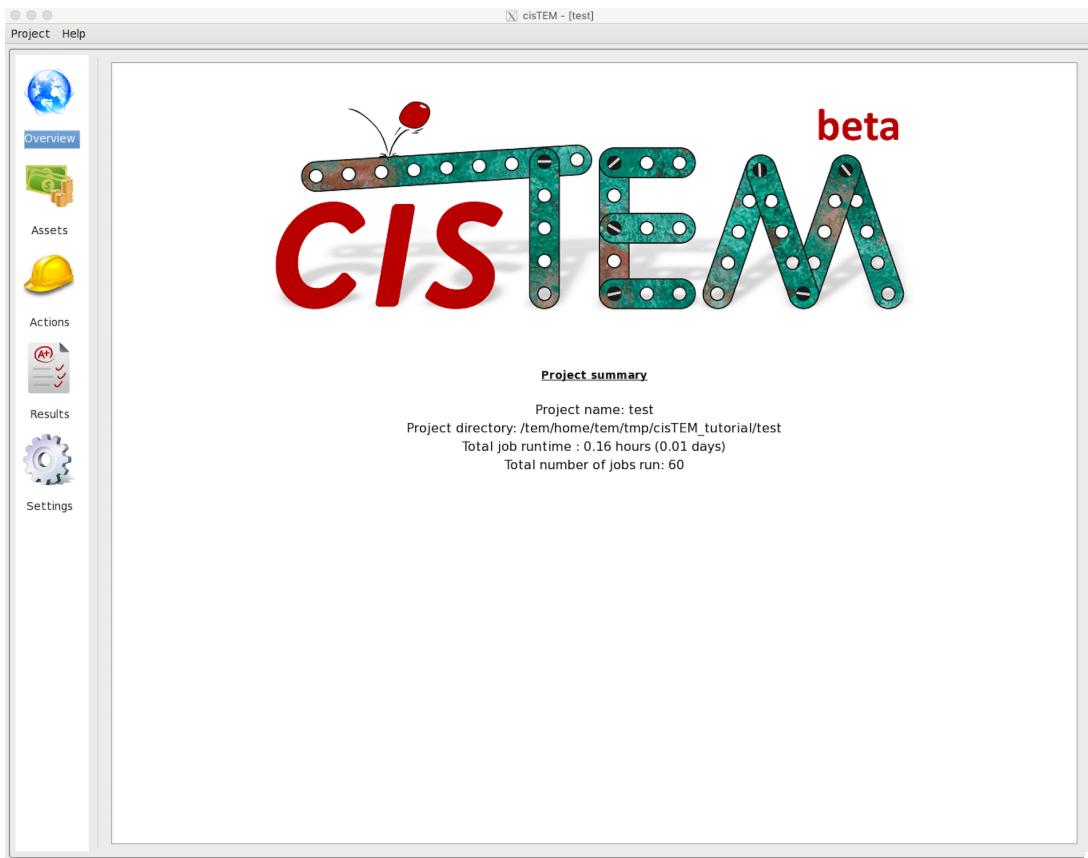
```
$> module load apps/cistem/1.0.0
$> module list
Currently Loaded Modulefiles:
 1) mpi/gcc/openmpi/4.0.3  2) apps/cistem/1.0.0
```

4. Check the cisTEM application binary path

```
$> which cisTEM
/tem/el7/cistem-1.0.0-beta/cisTEM
```

5. Execute the cisTEM application (we assume that X11 forwarding is enabled)

```
$> cisTEM
```



On startup, the GUI presents a list of previously opened projects, as well as options to create a new project or open an existing project. To continue a previous project, click on the provided link.

6.2 Run profiles for job submission

6.2.1 Profile templates

If you need cisTEM to work on multiple computing servers in a cluster which is managed with Torque, you should check out (or create) a “Run Profile” in cisTEM’s settings tab. You can find a shell script available in following file paths.

<pre>/tem/el7/qsub-cisTEM-cpu-noout.sh</pre>	<code>## output, error</code> 로그 파일을 생성하지 않는 cisTEM 작업 템플릿 <pre>/tem/el7/qsub-cisTEM-cpu.sh</pre> <code>## output, error</code> 로그 파일을 생성하는 cisTEM 작업 템플릿
--	--

For qsub-cisTEM-cpu.sh,

```
#!/bin/bash
queue=
while getopts ":q:" OPTION
do
  case "${OPTION}" in
    q) queue="${OPTARG}";;
    esac
done
shift $((OPTIND-1))
```

(continues on next page)

(continued from previous page)

```
cat - <<EOF | qsub
#!/bin/bash
#PBS -N cisTEM.${1}
${queue:+#PBS -l nodes=1:ppn=1:${queue}}
${queue:+#PBS -q ${queue} }

module load apps/cistem/1.0.0
${{@}}
EOF
```

For qsub-cisTEM-cpu-noout.sh,

```
#!/bin/bash
queue=
while getopts ":q:" OPTION
do
  case "${OPTION}" in
    q) queue="${OPTARG}";;
  esac
done
shift $((OPTIND-1))

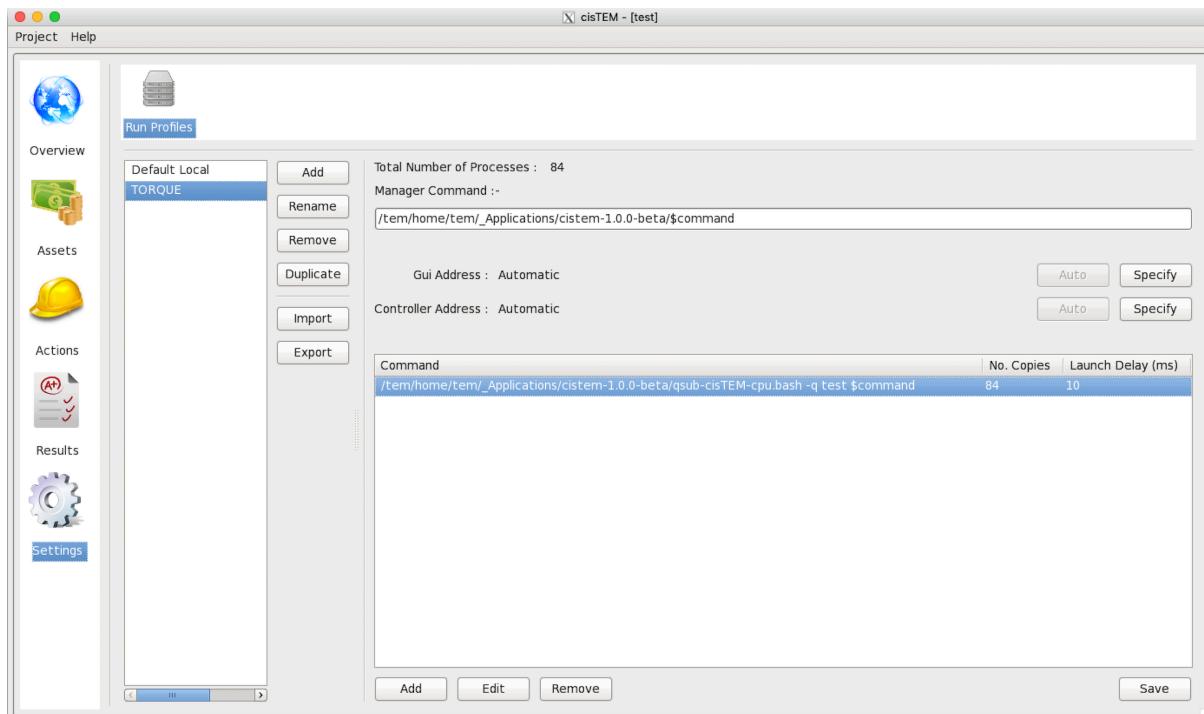
cat - <<EOF | qsub
#!/bin/bash
#PBS -N cisTEM.${1}
#PBS -e /dev/null
#PBS -o /dev/null
${queue:+#PBS -l nodes=1:ppn=1:${queue}}
${queue:+#PBS -q ${queue} }

module load apps/cistem/1.0.0
${{@}}
EOF
```

6.2.2 Adding a new Run Profile

In cisTEM settings, add a new “Run Profile” (called TORQUE here) with the following parameters :

- Manager Command: /tem/el7/cistem-1.0.0-beta/\$command
- Gui Address: Automatic
- Controller Address: Automatic
- Command -> Edit:
 - Command: /tem/el7/qsub-cisTEM-cpu.sh -q **cputQ** \$command
 - No. Copies: 84
 - Delay (ms): 10

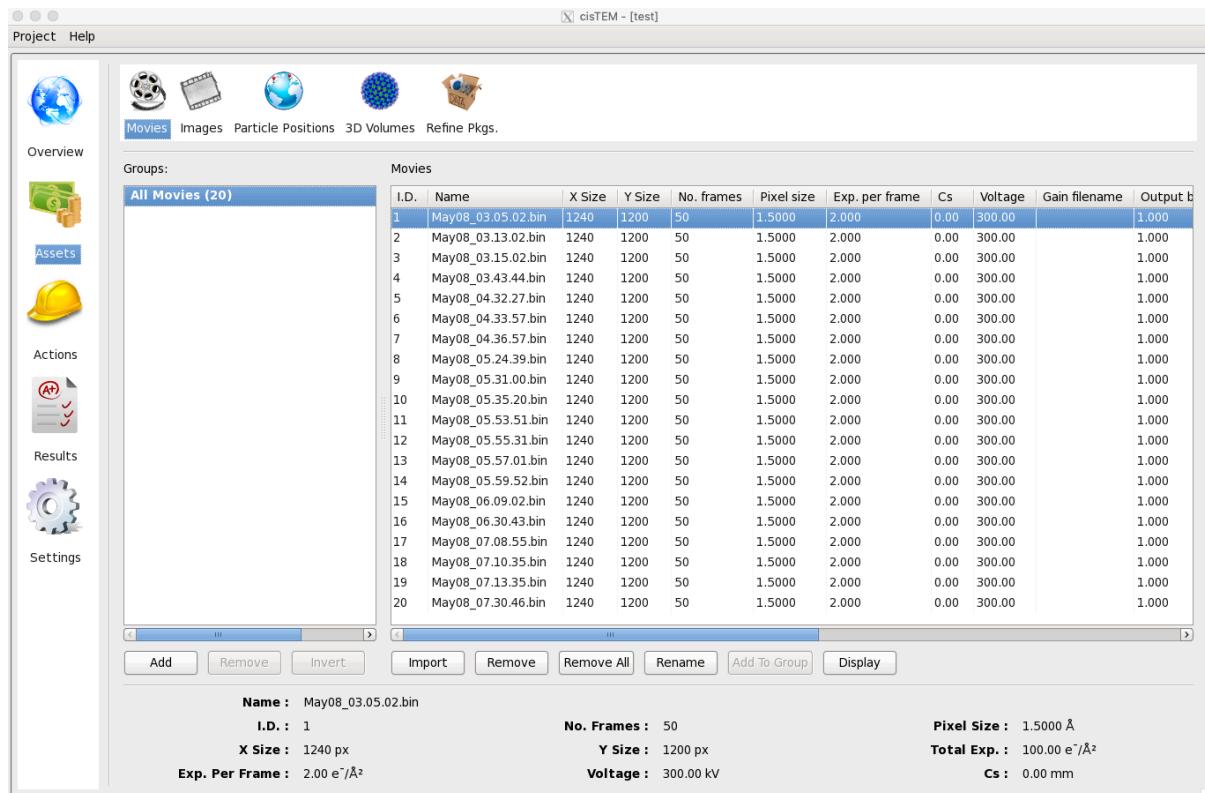


6.3 Examples of running cisTEM jobs

With the above cisTEM setting, here, we provide some examples of running cisTEM jobs with cisTEM GUI tools.

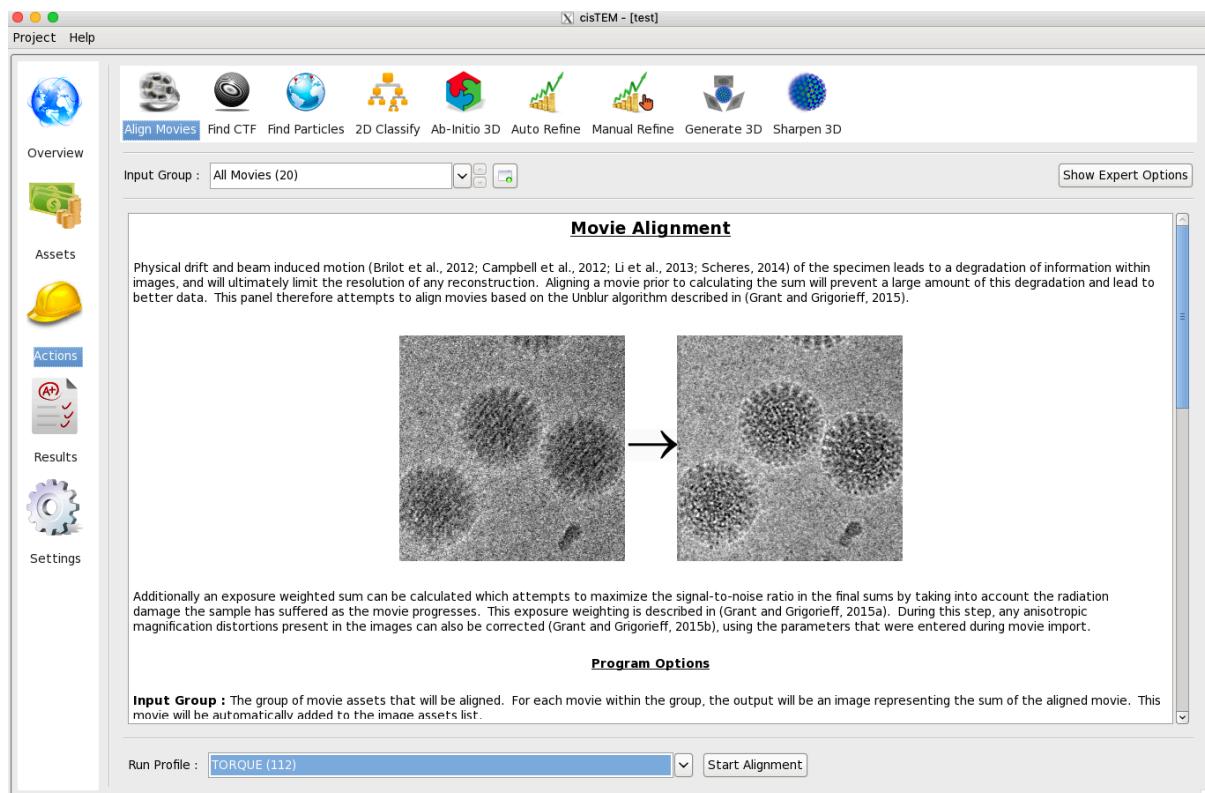
6.3.1 Importing Movies and images

Once a project is open or has been newly created, Assets can be imported. These will usually be Movies or Images but can also be Particle Positions, 3D Volumes and Refinement Packages. Click on Assets, then Movies and Import. In the dialog, select “Add Directory” and navigate to the directory containing your own movies. The movies are all part of a group called “All Movies”. Additional groups can be created using “Add” to select subsets of a dataset for further processing. You should continue with all the data for now. If images are available instead of movies, these can be imported as Image Assets in the same way as Movies, by clicking “Images” .



6.3.2 Movie Alignment

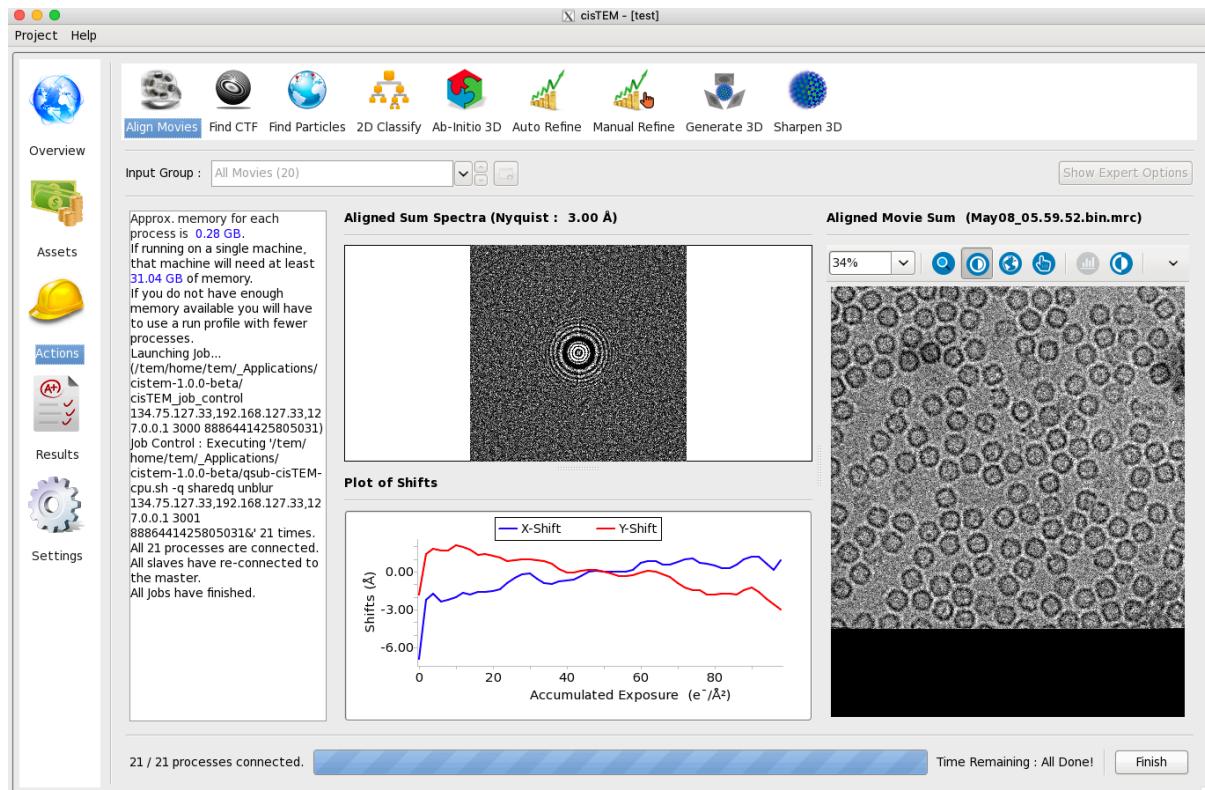
Movie data collection and frame alignment have been part of the single-particle image processing pipeline since it was first introduced by Brilot et al. in 2012. The original software **Unblur** was developed further by Grant & Grigorieff (2015) when exposure weighting was added to take into account the radiation-dependent signal loss when adding movie frames, yielding signal-optimized frame sums. cisTEM implements the Unblur algorithm in the Align Movies panel, which also provides some background to the method. Click “Actions” and select “Align Movies” to call up the panel.



Actions panels display parameters that you can change. Some of these are shown on the main panel while others are only accessible when “Show Expert Options” is selected. Movie alignment usually works with the default parameters and you should simply click “Start Alignment” near the bottom of the panel. You will notice that next to the start button a menu is shown that allows you to select different run profiles. The Local profile should **NOT** be selected because it will launch alignment jobs onto the login node but you should change to other profiles (for example, TORQUE profile) if these were previously set up under Settings.

The alignment of all the movies takes less than a minute. While the job is running, X,Y traces are displayed for some of the movies and a progress bar indicates the time left until completion of the job. After termination (you must click on “Finish” at the end of all jobs), you can inspect the results by clicking “Results”

초극저온전자현미경(Cryo-EM) 데이터 분석을 위한 분석 클러스터 및 소프트웨어 도구 활용 보고서, Release 2.0



CHAPTER
SEVEN

CRYOSPARC

CryoSPARC is the state-of-the-art platform used globally for obtaining 3D structural information from single particle cryo-EM data. The cryoSPARC platform enables automated, high quality and high-throughput structure discovery of proteins, viruses and molecular complexes for research and drug discovery.

Note: cryoSPARC official site : <https://cryosparc.com>

Note: At the time of writing this document (Jan. 2021), unfortunately, cryoSPARC v2.x, v3.x or v4.x does not provide the method of installing **a single cryoSPARC instance** (consisting of web application, command core, and database) **for use by a number of users with the complete isolation and security of their project data**. This problem might be resolved with later versions of cryoSPARC after CryoSPARC re-designs the product with the concept of “Hub” (as mentioned in cryoSPARC forum <https://discuss.cryosparc.com/t/use-linux-user-accounts/3480>). In the meanwhile, thus, we had to decide that each group must setup **a completely isolated cryoSPARC instance independently within thier own home directories** (/tem/scratch/<GroupDir>). This method relies on the UNIX system for security and is more tedious to manage but provides stronger access restrictions for users own dataset. For users convenience, we are ready to install and setup a cryoSPARC instance with **administrative automation codes on behalf of users**.

7.1 Prerequisites

Now, cryoSPARC is available free of charge for academic use. For a completely isolated cryoSPARC instance, user must have their own non-commercial license key for cryoSPARC v4. **Please visit the CryoSPARC official site, request a license key and inform the valid key to GSDC TEM service administrator by e-mail.**

7.2 Getting a cryoSPARC instance

CryoSPARC is a backend and frontend software system that provides data processing and image analysis capabilities for single particle cryo-EM, along with a browser based user interface and command line tools. CryoSPARC is composed of three major components : cryosparc_master, cryospace_database and cryosparc_worker.

- **cryosparc_master** : Master processes (webapps, command_core, databases, etc.) run together on one node (for our case, tem-cs-el7.sdfarm.kr login node). These processes host HTML5 based web applications, spawn or submit jobs to a cluster scheduler (for example, to PBS-based batch system)
- **cryosparc_worker** : Worker process can be spawned on any available worker nodes, and do data processing and image analysis tasks which are pre-defined within cryoSPARC software packages.
- **cryosparc_database** : CryoSPARC database is built on top of mongoDB, managing the metadata of users workflows, projects, jobs, backend clusters or workers as well as users.

7.2.1 1. (Admin) Install and setup a cryoSPARC instance

On behalf of users, administrator can execute ansible configuration automation code-snippets to install and setup a cryoSPARC instance, using a given valid license key. Master, worker and database sub-packages will be installed during configuration automation, which are located in **/tem/scratch/<GroupDir>/cryosparc** after finishing setup. A setup procedure includes registering both cluster(lane or worker nodes) instance and webapp's admin/normal users account. The whole setup will take about 10 minutes.

After finishing installation, **/tem/scratch/<GroupDir>/cryosparc** has following directories/files structure:

```
tem-cs-el7.sdfarm.kr $> cd ~/.cryosparc
tem-cs-el7.sdfarm.kr $> tree -L 1 .
.
├── cluster_info.json      ## cluster(lane) information to register
├── cluster_script.sh     ## PBS script template to submit jobs to worker cluster(lane)
├── cryosparc2_worker     ## dummy directory
├── cryosparc_master       ## cryosparc_master package install path
├── cryosparc_master.tar.gz
├── cryosparc_worker       ## cryosparc_worker package install path
├── cryosparc_worker.tar.gz
└── cryosparc_database    ## cryosparc_database package install path
```

Warning: !! CAUTION !! DO NOT delete or modify cryoSPARC instance base directory, **/tem/scratch/<GroupDir>/cryosparc**. The cryoSPARC base directory contains database. If this directory is deleted, all the project, job and workflow information will be corrupted and lost.

Also, the configuration code-snippets implicitly add cryoSPARC instance's binary path to PATH environment variable.

```
tem-cs-el7.sdfarm.kr $> cat /tem/home/<UserID>/.bashrc
...
```

(continues on next page)

(continued from previous page)

```
# User specific aliases and functions
export PATH='/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/bin':$PATH
```

7.2.2 2. (User) Verifying installation

By default, master processes (webapp, command_core, database, etc.) are automatically started during configuration automation. Users should check and verify whether the master processes are working correctly on tem-cs-el7.sdfarm.kr login node or not.

- Checking environment variables for cryoSPARC instance

```
tem-cs-el7.sdfarm.kr $> cryosparcm env

export "CRYOSPARD_HTTP_PORT=39xxx"
export "CRYOSPARD_MASTER_HOSTNAME=tem-cs-el7.sdfarm.kr"
export "CRYOSPARD_CLICK_WRAP=true"
export "CRYOSPARD_COMMAND_VIS_PORT=39xxx"
export "CRYOSPARD_CONDA_ENV=cryosparc_master_env"
export "CRYOSPARD_FORCE_USER=false"
export "CRYOSPARD_INSECURE=true"
export "CRYOSPARD_DEVELOP=false"
export "CRYOSPARD_DB_PATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_database"
export "CRYOSPARD_HTTP RTP_PORT=39xxx"
export "CRYOSPARD_LICENSE_ID=<license_key>"
export "CRYOSPARD_HOSTNAME_CHECK=tem-cs-el7.sdfarm.kr"
export "CRYOSPARD_MONGO_PORT=39xxx"
export "CRYOSPARD_MONGO_CACHE_GB=4"
export "CRYOSPARD_HEARTBEAT_SECONDS=60"
export "CRYOSPARD_ROOT_DIR=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master"
export "CRYOSPARD_HTTP RTP_LEGACY_PORT=39xxx"
export "CRYOSPARD_COMMAND_CORE_PORT=39xxx"
export "CRYOSPARD_BASE_PORT=39000"
export "CRYOSPARD_PATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/external/mongodb/
↪bin:/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/bin"
export "CRYOSPARD_LIVE_ENABLED=true"
export "CRYOSPARD_COMMAND RTP_PORT=39xxx"
export "CRYOSPARD_SUPERVISOR_SOCK_FILE=/tmp/cryosparc-supervisor-
↪627a9991e2f2f069094732dfd78d1696.sock"
export "CRYOSPARD_LD_LIBRARY_PATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/
↪cryosparc_compute/blobio"
export "CRYOSPARD_FORCE_HOSTNAME=false"
export "PATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/external/mongodb/bin:/tem/scratch/
↪<GroupDir>/cryosparc/cryosparc_master/bin:/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/
↪anaconda/envs/cryosparc_master_env/bin:/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/
↪anaconda/condabin:/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/bin:/usr/local/torque/bin:/usr/
↪local/torque/sbin:/usr/local/torque/bin:/usr/local/torque/sbin:/tem/el7/Modules/bin:/usr/local/bin:/bin:/usr/
↪bin:/usr/local/sbin:/usr/sbin:/tem/home/<userid>/bin"
export "LD_LIBRARY_PATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/cryosparc_compute/
↪blobio:"
export "LD_PRELOAD="
export "PYTHONPATH=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master"
export "PYTHONNOUSERSITE=true"
export "CONDA_SHLVL=1"
export "CONDA_PROMPT_MODIFIER=(cryosparc_master_env)"
```

(continues on next page)

(continued from previous page)

```
export "CONDA_EXE=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/anaconda/bin/conda"
export "CONDA_PREFIX=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/anaconda/envs/
↪cryosparc_master_env"
export "CONDA_PYTHON_EXE=/tem/scratch/<GroupDir>/cryosparc/cryosparc_master/deps/anaconda/bin/
↪python"
export "CONDA_DEFAULT_ENV=cryosparc_master_env"
```

You can find what kinds of environment variables have been set for the cryoSPARC instance.

Note: Especially, user should check **CRYOSPARD_BASE_PORT** (above, for example, 39000), which is **the listening port of cryoSPARC web application**. Later, this port number is used to make SSH tunneling between client and tem-cs-el7.sdfarm.kr login node. **Via the tunneled connection over SSH, users can access the web UI of cryoSPARC instance.**

- Checking the status of cryoSPARC instance

```
tem-cs-el7.sdfarm.kr $> cryosparcm status
```

```
-----  
CryoSPARC System master node installed at  
/tem/scratch/<GroupID>/cryosparc/cryosparc_master  
Current cryoSPARC version: v4.0.0
```

CryoSPARC process status:

```
app                  RUNNING pid 14307, uptime 0:00:09
app_api              RUNNING pid 14317, uptime 0:00:08
app_api_dev          STOPPED Not started
app_legacy           STOPPED Not started
app_legacy_dev       STOPPED Not started
command_core         RUNNING pid 14153, uptime 0:00:40
command_rtp          RUNNING pid 14247, uptime 0:00:26
command_vis          RUNNING pid 14240, uptime 0:00:27
database             RUNNING pid 14035, uptime 0:00:44
```

```
-----  
License is valid
```

global config variables:

```
export CRYOSPARD_LICENSE_ID=<license_key>
export CRYOSPARD_MASTER_HOSTNAME="tem-cs-el7.sdfarm.kr"
export CRYOSPARD_DB_PATH="/tem/scratch/<GroupDir>/cryosparc/cryosparc_database"
export CRYOSPARD_BASE_PORT=39xxx
export CRYOSPARD_DB_CONNECTION_TIMEOUT_MS=20000
export CRYOSPARD_INSECURE=true
export CRYOSPARD_DB_ENABLE_AUTH=true
export CRYOSPARD_CLUSTER_JOB_MONITOR_INTERVAL=10
export CRYOSPARD_CLUSTER_JOB_MONITOR_MAX_RETRIES=1000000
export CRYOSPARD_PROJECT_DIR_PREFIX='CS-'
export CRYOSPARD_DEVELOP=false
export CRYOSPARD_CLICK_WRAP=true
```

7.3 Launching CryoSPARC instance

We assume that user's network setup looks like (most commonly used scenario):

internet

[localhost] ======[firewall | tem-cs-el7.sdfarm.kr]

7.3.1 For Linux/Mac users

With the following command, you can start an SSH tunnel to export **CRYOSPARC_BASE_PORT** from tem-cs-el7.sdfarm.kr to your local client machine.

```
localhost $> ssh -N -f -L localhost:39500:tem-cs-el7.sdfarm.kr:<CRYOSPARC_BASE_PORT> -o Port=<ssh_
→port> <userid>@tem-cs-el7.sdfarm.kr

## 39500 port on localhost : assume that the port number 39500 is available on your localhost. Otherwise, □
→you can use another port available.
## -N : Do not execute a remote command. This is useful option for just forwarding ports.
## -f : Requests ssh to go to background just before command execution.
## -L [bind_address:]port:host:hostport
```

Note: You should execute this ‘ssh’ command on **YOUR LOCAL PC/WORKSTATION** to make a tunnel between your local machine and tem-cs-el7.sdfarm.kr (localhost:39500 \leftrightarrow tem-cs-el7.sdfarm.kr:<CRYOSPARC_BASE_PORT>) over secure channel.

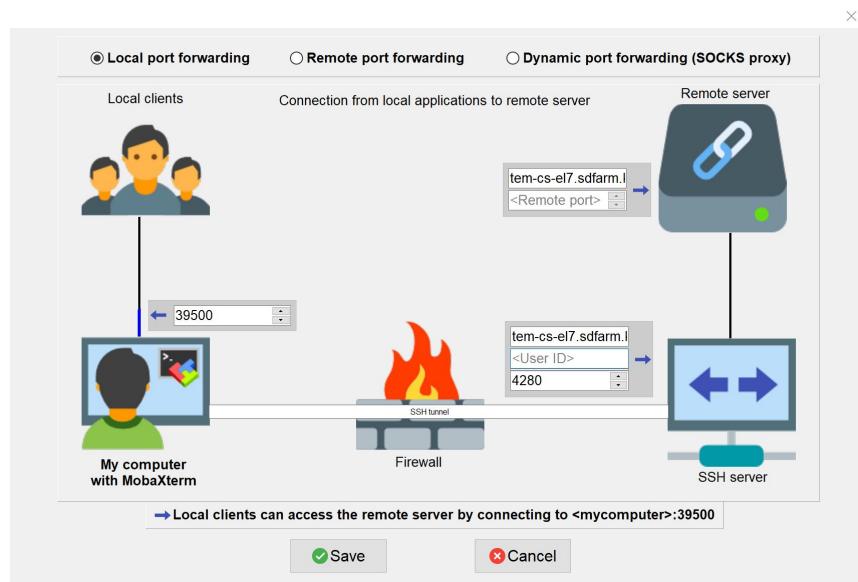
Note: You can close the terminal window (because ‘ssh’ will be run in the background) after running the above command. The tunnel will stay open.

Now, open your browser (Chrome/Firefox/Safari recommended) and navigate to <http://localhost:39500>. You should be presented with the cryoSPARC login page.

7.3.2 For Windows users

- Using MobaXterm
 - Open ‘MobaXterm’ application.
 - ‘MobaXterm’ → ‘Tools’ → ‘MobaSSHTunnel (port forwarding)’ : Open MobaSSHTunnel dialog box.
 - ‘New SSH tunnel’ : Set a forwarded port binding option and save the setting.
 - Give the name to the saved port forwarding settings, and start the tunnel connection.

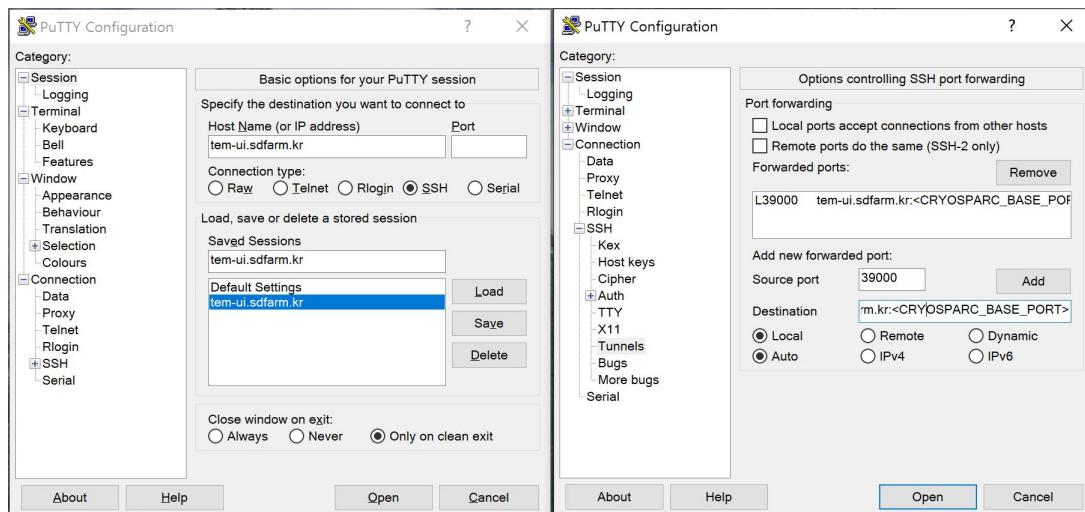
Note: You must use **CRYOSPARC_BASE_PORT** for the ‘Remote server’ port section.



Now, open your browser (Chrome/Firefox/Safari recommended) and navigate to <http://localhost:39500>. You should be presented with the cryoSPARC login page.

- Using Putty
 - Open ‘PuTTy Configuration’ dialog box.
 - ‘PuTTy Configuration’ -> ‘Session’ : Load a SSH session to connect tem-cs-el7.sdfarm.kr login node with the known <ssh_port>.
 - ‘PuTTy Configuration’ -> ‘Connection’ -> ‘SSH’ -> ‘Tunnels’ : Set a forwarded port binding option and add the entry.

Note: You must use **tem-cs-el7.sdfarm.kr:CRYOSPARC_BASE_PORT** for the ‘Destination’ field.

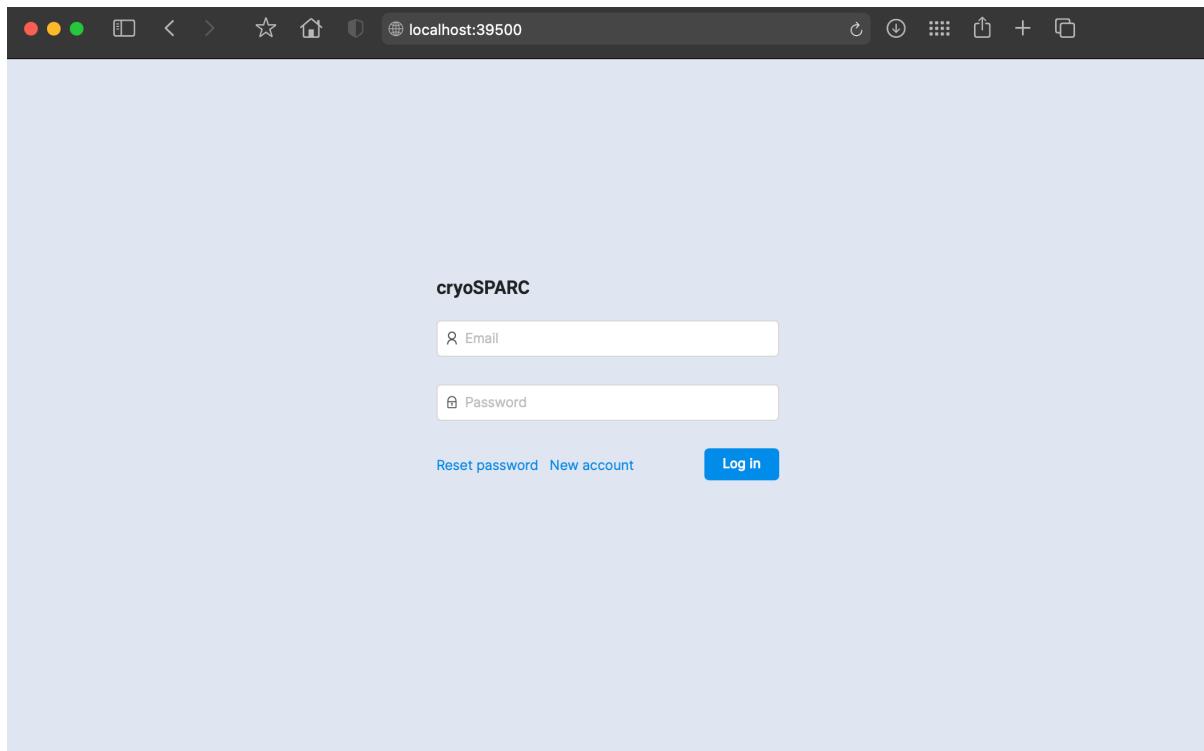


Now, open your browser (Chrome/Firefox/Safari recommended) and navigate to <http://localhost:39500>. You should be presented with the cryoSPARC login page.

7.4 Exploring CryoSPARC web apps

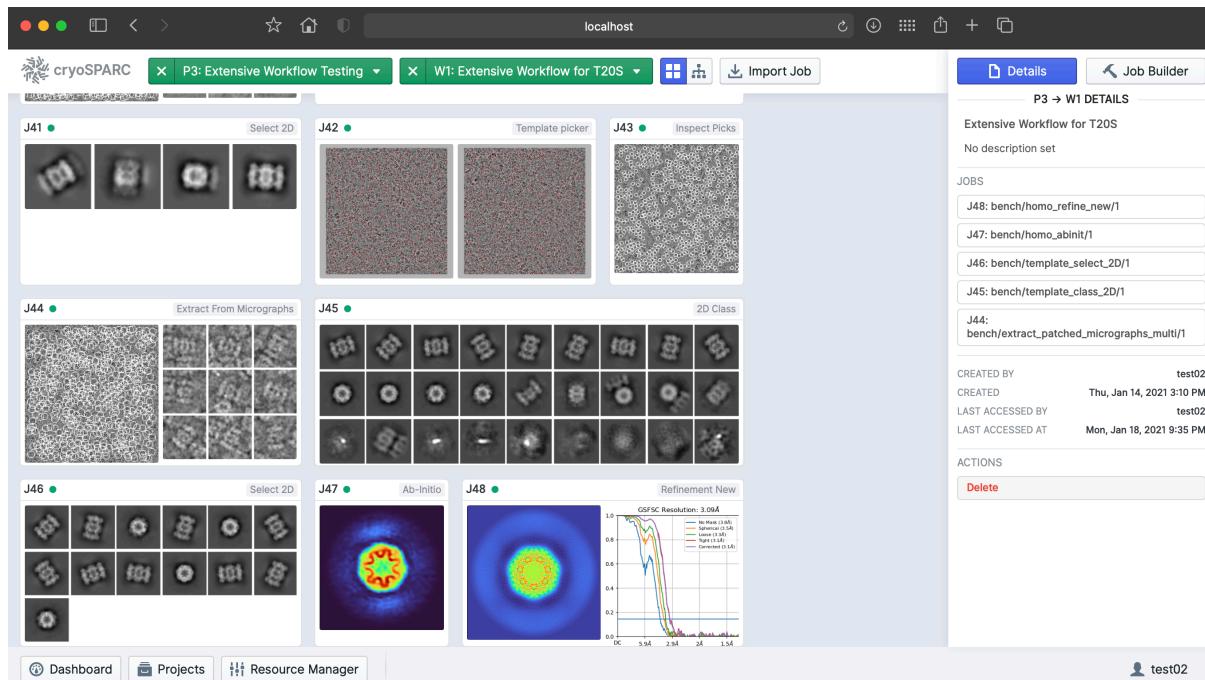
7.4.1 CryoSPARC login

E-mail and password information will be notified to users as the installation and setup is finished. Given e-mail and password, users can login to cryoSPARC web interfaces.

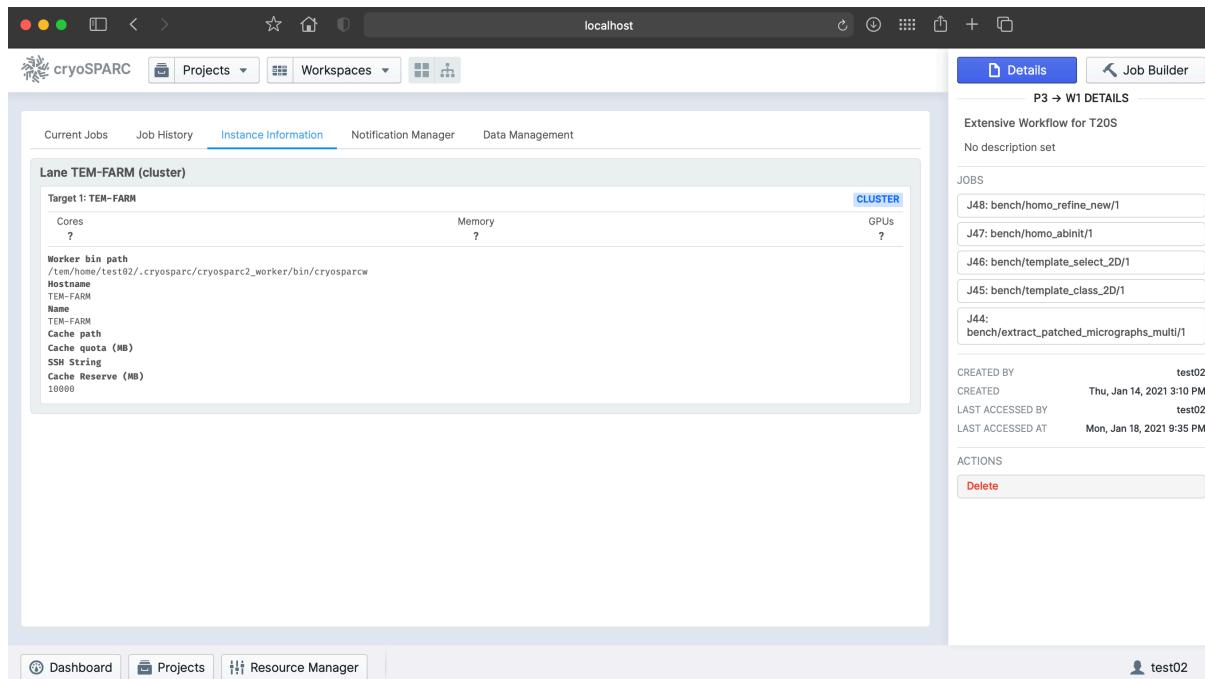


7.4.2 CryoSPARC dashboard

7.4.3 CryoSPARC project



7.4.4 CryoSPARC cluster(lane)



Note: For details on user interface and usages of cryoSPARC, refer to cryoSPARC's official document.
<https://cryosparc.com/docs/reference/general>

7.5 Tutorial on processing T20S

- Please refer to CryoSPARC' s webpage for the tutorial on processing T20S : <https://cryosparc.com/docs/tutorials/t20s>

7.6 Trouble shooting

7.6.1 1. How to stop or start the cryoSPARC instance?

- Stop the running cryoSPARC instance

```
tem-cs-el7.sdfarm.kr $> cryosparcm stop
```

```
CryoSPARC is running.  
Stopping cryoSPARC  
app: stopped  
command_core: stopped  
command_rtp: stopped  
command_vis: stopped  
liveapp: stopped  
webapp: stopped  
database: stopped  
Shut down
```

Stop the cryosparc instance if running. This will gracefully kill all the master processes, and will cause any running jobs (potentially on other nodes) to fail.

- Start the cryoSPARC instance

```
tem-cs-el7.sdfarm.kr $> cryosparcm start
```

```
Starting cryoSPARC System master process..  
CryoSPARC is not already running.  
configuring database  
  configuration complete  
database: started  
  database configuration is OK.  
command_core: started  
  command_core connection succeeded  
  command_core startup successful  
command_vis: started  
command_rtp: started  
  command_rtp connection succeeded  
  command_rtp startup successful  
app: started  
app_api: started
```

```
-----  
CryoSPARC master started.  
From this machine, access cryoSPARC and cryoSPARC Live at  
  http://localhost:39xxx
```

```
From other machines on the network, access cryoSPARC and cryoSPARC Live at
```

(continues on next page)

(continued from previous page)

http://tem-cs-el7.sdfarm.kr:39xxx

Startup can take several minutes. Point your browser to the address
and refresh **until** you see the cryoSPARC web interface.

Start the cryosparc instance if stopped. This will cause the database, command, webapp etc processes to start up. Once these processes are started, they are run in the background, so the current shell can be closed and the web UI will continue to run, as will jobs that are spawned.

7.6.2 2. How to reset the password of non-admin user?

Users can reset the non-admin user' s password to a new password with the following command-line execution:

```
tem-cs-el7.sdfarm.kr $> cryosparcm resetpassword --email <email address> --password <newpassword>
```

7.6.3 3. Job (or Workflow) failed caused by SSD caching

Job failure log looks like:

```
[CPU: 166.4 MB] Traceback (most recent call last):
File "cryosparc2_worker/cryosparc2_compute/run.py", line 82, in cryosparc2_compute.run.main
File "cryosparc2_worker/cryosparc2_compute/jobs/class2D/run.py", line 64, in cryosparc2_compute.jobs.
  ↪class2D.run.run_class_2D
File "cryosparc2_compute/particles.py", line 61, in read_blobs
  ↪u_blob_paths = cache.download_and_return_cache_paths(u_rel_paths)
File "cryosparc2_compute/jobs/cache.py", line 129, in download_and_return_cache_paths
  ↪other_instance_ids = get_other_instance_ids(instance_id, ssd_cache_path)
File "cryosparc2_compute/jobs/cache.py", line 250, in get_other_instance_ids
  ↪all_instance_ids = [p for p in os.listdir(ssd_cache_path) if os.path.isdir(os.path.join(ssd_cache_path, p)) and p.
  ↪startswith('instance_')]
OSSError: [Errno 2] No such file or directory: "
```

During cryoSPARC configuration, we did not provide an option to support any **SSD caching** due to the lack of SSD (or NVMe SSD) drives on the worker nodes. However, by default, cryoSPARC seems to have 'SSD caching' enabled on its Web user interface. When you are running jobs that process particles (for example: Ab-Initio, Homogeneous Refinement, 2D Classification, 3D Variability), you will find a parameter at the bottom of the job builder under "Compute Settings" called **Cache particle images on SSD**. Turn this option off to load raw data from their original location instead.

Also, you can set a default parameter value of each project. By default, the Cache particle images on SSD parameter is always on for every job you build, but if you'd like to keep this option off across all jobs in a project, you can set a project-level default by running the following command in a shell on the UI node:

```
tem-cs-el7.sdfarm.kr $> cryosparcm cli "set_project_param_default('PX', 'compute_use_ssd', False)"
```

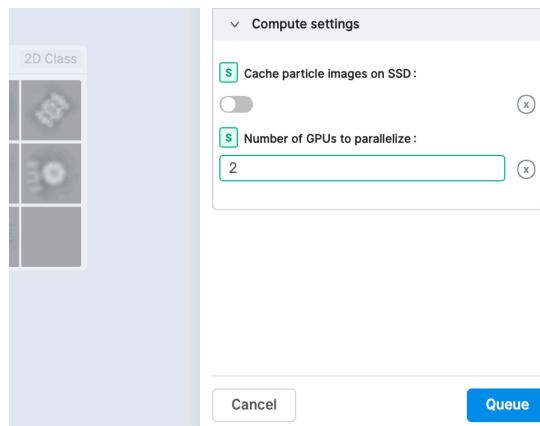
where 'PX' is the Project ID you want to set the default value for (e.g., 'P1' , 'P2' , etc.)

You can undo this setting by running:

```
tem-cs-el7.sdfarm.kr $> cryosparcm cli "unset_project_param_default('PX', 'compute_use_ssd')"
```

7.6.4 4. Failed to launch! 190

As you submit a cryoSPARC job to TEM-FARM cluster, you can encounter an error such as “Failed to launch! 190”. If you face with this error, you should check **Number of GPUs to parallelize** option under “Compute Settings” of job builder first. Unlike relion, it seems that all the cryoSPARC jobs can not distribute their worker processes onto multiple worker nodes (that is, all the cryoSPARC job is executed on a single CPU or GPU node). So, the maximum number of GPUs which can be used to parallelize within a job is 2 (because there are 2 P100 or P40 GPUs on each GPU node in GSDC TEM FARM).



7.6.5 5. Binary locations of Gctf, MotionCor2

The GPU environment of GSDC TEM farm is built on top of NVIDIA CUDA SDK (driver version 460.27.04 and CUDA library version 9.2). Some 3rd-party applications with GPU acceleration, for example, Gctf, MotionCor2, which can be utilized within various number of Cryo-EM toolkit are provided, and you can find those binaries in the following locations:

```
## Gctf

$> module avail
----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0 apps/relion/cpu/3.0.7 apps/relion/cpu/3.1.0 apps/relion/gpu/3.0.7 apps/relion/gpu/3.1.0

----- /tem/el7/Modules/acceleration -----
cuda/9.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/openmpi/4.0.3

----- /tem/el7/Modules/virtualenv -----
conda/2020.11

----- /tem/el7/Modules/tools -----
tools/ctffind/4.1.14 tools/motioncor2/1.3.1 tools/summovie/1.0.2
tools/gctf/1.18_b2 tools/resmap/1.1.4 tools/unblur/1.0.2
```

(continues on next page)

(continued from previous page)

```
$> module show tools/gctf/1.18_b2
```

```
-----  
/tem/el7/Modules/tools/tools/gctf/1.18_b2:
```

```
module-whatis {Setup gctf v1.18_b2}  
module load cuda/9.2  
prepend-path PATH /tem/el7/Gctf_v1.18_b2/bin  
conflict tools/gctf
```

```
$> ls -al /tem/el7/Gctf_v1.18_b2/bin
```

```
total 63122
```

```
drwxr-xr-x. 2 tem tem 462 Apr 9 2020 .  
drwxr-xr-x. 4 tem tem 42 Apr 9 2020 ..  
-rwxr-xr-x. 1 tem tem 3429036 Aug 22 2018 Gctf_v1.18_b2_sm60_cu8.0  
-rwxr-xr-x. 1 tem tem 3520460 Aug 22 2018 Gctf_v1.18_b2_sm60_cu9.0  
-rwxr-xr-x. 1 tem tem 3674669 Aug 22 2018 Gctf_v1.18_b2_sm60_cu9.2 # (compatible)  
-rwxr-xr-x. 1 tem tem 3429036 Aug 22 2018 Getf_v1.18_b2_sm61_cu8.0  
-rwxr-xr-x. 1 tem tem 3520460 Aug 22 2018 Getf_v1.18_b2_sm61_cu9.0  
-rwxr-xr-x. 1 tem tem 3674669 Aug 22 2018 Getf_v1.18_b2_sm61_cu9.2 # (compatible)  
-rwxr-xr-x. 1 tem tem 3429036 Aug 22 2018 Getf_v1.18_b2_sm62_cu8.0  
-rwxr-xr-x. 1 tem tem 6224329 Aug 22 2018 Getf_v1.18_b2_sm62_cu9.0  
-rwxr-xr-x. 1 tem tem 6373822 Aug 22 2018 Getf_v1.18_b2_sm62_cu9.2 # (compatible)  
-rwxr-xr-x. 1 tem tem 3959148 Aug 22 2018 Getf_v1.18_b2_sm70_cu9.0  
-rwxr-xr-x. 1 tem tem 4117037 Aug 22 2018 Getf_v1.18_b2_sm70_cu9.2
```

```
## MotionCor2
```

```
$> module show tools/motioncor2/1.3.1
```

```
-----  
/tem/el7/Modules/tools/tools/motioncor2/1.3.1:
```

```
module-whatis {Setup motioncor2 1.3.1}  
module load cuda/9.2  
prepend-path PATH /tem/el7/MotionCor2_v1.3.1  
conflict tools/motioncor2
```

```
$> ls -al /tem/el7/MotionCor2_v1.3.1
```

```
total 24840
```

```
drwxr-xr-x. 2 tem tem 182 Oct 27 00:34 .  
drwxr-xr-x. 15 tem tem 653 Jan 15 17:51 ..  
-rwxr-xr-x. 1 tem tem 10200208 Oct 23 2019 MotionCor2-UserManual-10-22-2019.pdf  
-rwxr-xr-x. 1 tem tem 2712344 Jan 24 2020 MotionCor2_v1.3.1-Cuda101  
-rwxr-xr-x. 1 tem tem 2696304 Jan 24 2020 MotionCor2_v1.3.1-Cuda102  
-rwxr-xr-x. 1 tem tem 2712312 Jan 24 2020 MotionCor2_v1.3.1-Cuda92 # (compatible)
```

**CHAPTER
EIGHT**

TOPAZ

A pipeline for particle detection in cryo-electron microscopy images using convolutional neural networks trained from positive and unlabeled examples. Topaz also includes methods for micrograph and tomogram denoising using deep denoising models.

- Official **Topaz** site : <http://cb.csail.mit.edu/cb/topaz/>

8.1 Topaz Executables

Basically, topaz uses NVIDIA GPUs with CUDA support for GPU acceleration in order to provide the features like automatic particle picking, denoising, etc. Here, we have installed Topaz into separate conda environments, where each is built upon CUDA 9.x and CUDA 11.x. You can find out each version of topaz executables with the following commands:

```
$> module avail

----- /tem/el7/Modules/apps -----
apps/cistem/1.0.0
apps/relion/cpu/3.0.7
apps/relion/cpu/3.1.0
apps/relion/cpu/4.0.0
apps/relion/gpu/3.0.7
apps/relion/gpu/3.1.0
apps/relion/gpu/4.0.0

---- /tem/el7/Modules/acceleration ---
cuda/9.2 cuda/11.2

----- /tem/el7/Modules/mpi -----
mpi/gcc/4.8.5/openmpi/4.0.3
mpi/gcc/8.3.1/mpich/3.4.3
mpi/gcc/8.3.1/openmpi/4.0.3
mpi/gcc/openmpi/4.0.3

---- /tem/el7/Modules/virtualenv ---
conda/2020.11 topaz/cuda-9.2/0.2.4
pyem/0.5    topaz/cuda-11.0/0.2.4

----- /tem/el7/Modules/tools -----
tools/aspera-cli/3.9.6
tools/ctffind/4.1.14
tools/gctf/1.18_b2
```

(continues on next page)

(continued from previous page)

```
tools/motioncor2/1.3.1
tools/resmap/1.1.4
tools/summovie/1.0.2
tools/unblur/1.0.2

----- /tem/el7/Modules/experiment -----
PyRosetta/4
python/3.7
rosetta/mpich-3.4.3/3.13
rosetta/openmpi-4.0.3/3.13
```

8.1.1 Topaz binary executable with CUDA 9.2

```
$> module load topaz/cuda-9.2/0.2.4
Loading topaz/cuda-9.2/0.2.4
Loading requirement: conda/2020.11

(topaz-v0.2.4-cuda9.2) $>
```

```
(topaz-v0.2.4-cuda9.2) $> which topaz
/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda9.2/bin/topaz
```

With above command, you can find that the path of topaz executable with CUDA 9.x support is **/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda9.2/bin/topaz**.

8.1.2 Topaz binary executable with CUDA 11.2

```
$> module load topaz/cuda-11.0/0.2.4
Loading topaz/cuda-11.0/0.2.4
Loading requirement: conda/2020.11

(topaz-v0.2.4-cuda11.0) $>
```

```
(topaz-v0.2.4-cuda11.0) $> which topaz
/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda11.0/bin/topaz
```

With above command, you can find that the path of topaz executable with CUDA 11.x support is **/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda11.0/bin/topaz**.

8.2 Using Topaz in CryoSPARC

If your CryoSPARC instance's version is v3.0.1, you should use the topaz executable **/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda9.2/bin/topaz** because the CryoSPARC is built upon CUDA 9.2 support. Otherwise if you use CryoSPARC v3.2.0+, the topaz executable is **/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda11.0/bin/topaz**.

Please refer to <https://guide.cryosparc.com/processing-data/all-job-types-in-cryosparc/deep-picking/topaz> for executing topaz jobs in CryoSPARC.

8.3 Using Topaz in Relion (v3.1.0 above)

To run topaz jobs within relion data analysis tool, you **SHOULD USE Relion v3.1 above and Topaz v0.2.4 above**. For your convenience, we have cloned all the `relion_run_topaz` packages to GSDC TEM farm.

Here is the location of `relion_run_topaz` packages:

- `/tem/el7/topaz/relion_run_topaz/run_topaz_denoise.py`
- `/tem/el7/topaz/relion_run_topaz/run_topaz_pick.py`
- `/tem/el7/topaz/relion_run_topaz/run_topaz_train.py`
- `/tem/el7/topaz/relion_run_topaz/run_topaz_train_denoise.py`

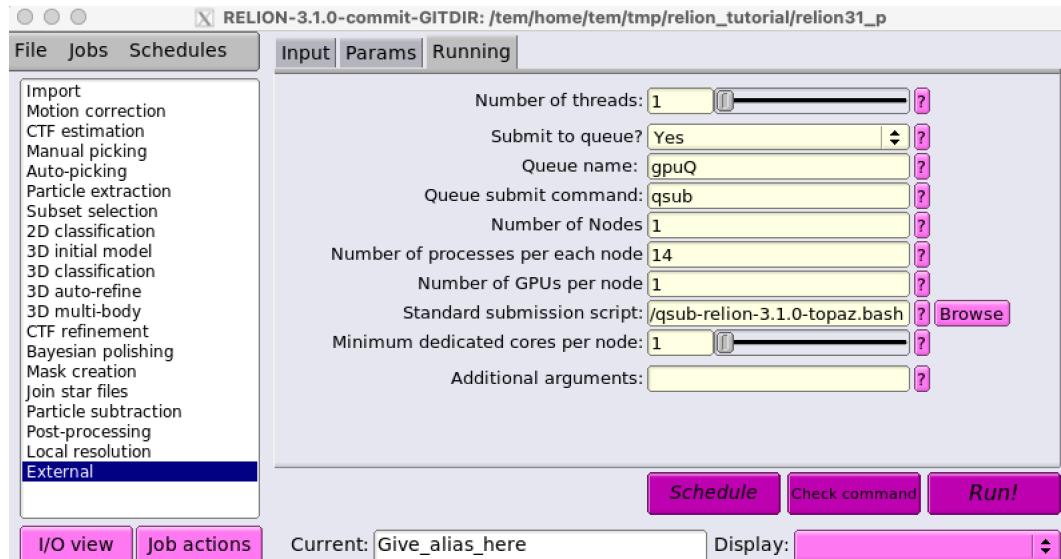
```
$> pwd
/tem/el7/topaz/relion_run_topaz
[tem@tem-cs-el7 relion_run_topaz]$ tree .
.
├── LICENSE
├── README.md
├── run_topaz_denoise.py
├── run_topaz_pick.py
├── run_topaz_train.py
└── run_topaz_train_denoise.py

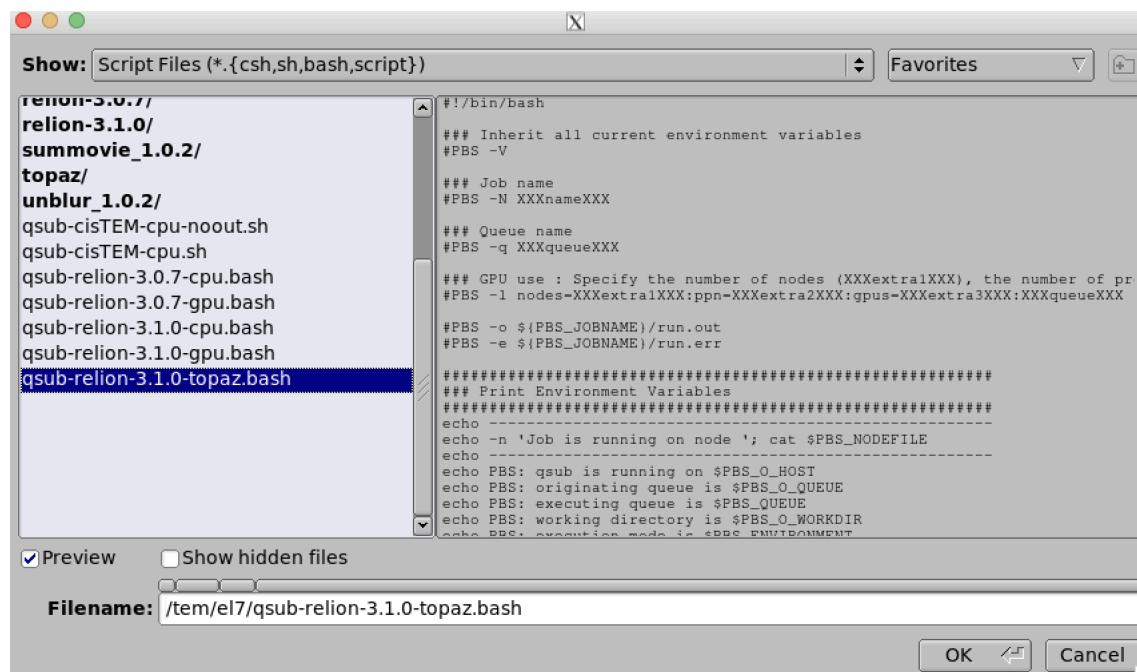
0 directories, 6 files
```

Since Relion v3.1+ has built using CUDA 9.2 support, we have modified `topaz_path` with this executable `/tem/el7/conda3-2020.11/envs/topaz-v0.2.4-cuda9.2/bin/topaz` in all the above python scripts.

For an external job type in relion GUI, after providing python executable path, inputs and optional parameters, you can use one of the following job submission templates.

- `/tem/el7/qsub-relion-3.1.0-topaz.bash`
- `/tem/el7/qsub-relion-4.0.0-topaz.bash`





Note that both number of nodes and number of GPUs are set to 1 due to the lack of parallelism across multiple servers and GPUs in Topaz application. For more details on running topaz jobs in relion (using external job type), please refer to https://github.com/tbepler/topaz/tree/master/relion_run_topaz.

CHAPTER
NINE

FAQS

- *how to resolve the problems on (re)starting your own cryosparc instance*
- *how to repair cryosparc database corruption caused by duplicated mongod executions*
- *how to update cryosparc softwares*

9.1 How to resolve the problems on (re)starting your own cryosparc instance?

You should check all the cryosparc related processes (i.e., supervisord, mongod, command_core, command_vis, command_rtp, webapp, app, liveapp) to be terminated successfully first on **tem-cs-el7.sdfarm.kr** server.

```
(log into tem-cs-el7.sdfarm.kr server first)
(example) userid@tem-cs-el7 $> cryosparcm stop
```

```
CryoSPARC is running.
Stopping cryoSPARC
app: stopped
command_core: stopped
command_rtp: stopped
command_vis: stopped
liveapp: stopped
webapp: stopped
database: stopped
Shut down
```

```
(example) userid@tem-cs-el7 $> ps aux | grep <userid> | grep cryosparc
userid 2449 0.0 0.0 152792 17480 ? Ss Jun24 0:18 python /tem/scratch/<GroupDir>/.cryosparc/
  ↵cryosparc_master/deps/anaconda/envs/cryosparc_master_env/bin/supervisord -c /tem/scratch/<GroupDir>/
  ↵cryosparc/cryosparc_master/supervisord.conf
userid 2472 1.2 0.0 1429268 57412 ? Sl Jun24 13:19 mongod --dbpath /tem/scratch/<GroupDir>/
  ↵cryosparc/cryosparc_database --port 39031 --oplogSize 64 --replSet meteor --nojournal --
  ↵wiredTigerCacheSizeGB 4
userid 2900 0.2 0.0 860572 83028 ? Sl Jun24 2:23 python -c import cryosparc_command.command_
  ↵core as serv; serv.start(port=39032)
userid 4332 0.1 0.1 854192 236396 ? Sl Jun24 1:53 python -c import cryosparc_command.command_
  ↵vis as serv; serv.start(port=39033)
userid 4378 0.4 0.0 1190536 196316 ? Sl Jun24 4:59 python -c import cryosparc_command.command_
  ↵rtp as serv; serv.start(port=39035)
userid 5586 0.1 0.0 1331136 116972 ? Sl Jun24 1:17 /tem/scratch/<GroupDir>/.cryosparc/cryosparc_
```

(continues on next page)

(continued from previous page)

```
↪master/cryosparc_webapp/nodejs/bin/node ./bundle/main.js
userid 5625 0.1 0.0 1049868 97640 ? S1 Jun24 1:22 /tem/scratch/<GroupDir>/cryosparc/cryosparc_
↪master/cryosparc_app/nodejs/bin/node ./bundle/main.js
userid 5690 0.2 0.0 1326136 81432 ? S1 Jun24 2:12 /tem/scratch/<GroupDir>/cryosparc/cryosparc_
↪master/cryosparc_liveapp/nodejs/bin/node ./bundle/main.js
```

```
(example) userid@tem-cs-el7 $> kill -9 2449 2472 2900 4332 4378 5586 5625 5690
```

Second, find your own cryosparc unix socket files on /tmp directory, and if exists, delete the files using rm command.

```
userid@tem-cs-el7 $> cd /tmp
(example) userid@tem-cs-el7 $> ls -al | grep <userid> | grep sock

srwx-----. 1 userid      userid      0 Jun 24 16:39 cryosparc-supervisor-627a9991e2f2f069094732dfd78d1696.
↪sock
srwx-----. 1 userid      userid      0 Jun 24 16:39 mongodb-39031.sock

(example) userid@tem-cs-el7 $> rm cryosparc-supervisor-627a9991e2f2f069094732dfd78d1696.sock
(example) userid@tem-cs-el7 $> rm mongodb-39031.sock
```

Then, start your cryosparc instance.

```
(example) userid@tem-cs-el7 $> cryosparcm start
```

Starting cryoSPARC System master process..

CryoSPARC is not already running.

database: started

command_core: started

 command_core connection succeeded

 command_core startup successful

command_vis: started

command_rtp: started

 command_rtp connection succeeded

 command_rtp startup successful

webapp: started

app: started

liveapp: started

CryoSPARC master started.

From this machine, access cryoSPARC at

 http://localhost:39030

and access cryoSPARC Live at

 http://localhost:39036

please note the legacy cryoSPARC Live application is running at

 http://localhost:39037

From other machines on the network, access cryoSPARC at

 http://tem-cs-el7.sdfarm.kr:39030

and access cryoSPARC Live at

 http://tem-cs-el7.sdfarm.kr:39036

Startup can take several minutes. Point your browser to the address
and refresh **until** you see the cryoSPARC web interface.

9.2 How to repair cryosparc database corruption?

With duplicated mongod executions, cryosparc database can be corrupted resulting in “**database: ER-ROR (spawn error)**” on (re)starting cryosparc instance. To address this abnormal case, you can try to repair the database with followings:

First, **stop all the cryosparc processes and delete the unix socket files**. See *how to resolve the problems on (re)starting your own cryosparc instance* for more details.

Second, try to repair the cryosparc database i.e., mongodb.

```
userid@tem-cs-el7 $> cryosparcm env
userid@tem-cs-el7 $> cd /tem/scratch/<GroupDir>/cryosparc
userid@tem-cs-el7 $> tar cvfz cryosparc_database.backup.tar.gz cryosparc_database
userid@tem-cs-el7 $> eval $(cryosparcm env)
userid@tem-cs-el7 $> cd cryosparc_database
userid@tem-cs-el7 $> mongod ./ --repair
```

9.3 How to upgrade (or downgrade) to the specific version of cryosparc softwares?

The following update guides summarize the procedure for cryosparc’s master and worker software updates. For more details, please refer to <https://guide.cryosparc.com/setup-configuration-and-management/software-updates>.

9.3.1 1. Checking for updates

Log into the tem-cs-el7.sdfarm.kr server where the cryosparc master is installed using ssh. Then, run this command if you want to check updates.

```
userid@tem-cs-el7 $> cryosparcm update --check
CryoSPARC current version v4.0.0
    update starting on Wed Mar 18 12:09:52 EDT 2021

current version v4.0.0
    new version v4.1.0

Update available!
```

Also, you can use this command **cryosparcm update -list** to get a full list of available versions.

```
userid@tem-cs-el7 $> cryosparcm update --check
CryoSPARC current version v4.0.0
    update starting on Wed Mar 18 12:09:52 EDT 2021

Available versions:

v2.0.18
v2.0.20
v2.0.23
v2.0.27
```

(continues on next page)

(continued from previous page)

```
v2.1.0
v2.2.0
v2.3.0
v2.3.2
v2.4.0
v2.4.2
v2.4.5
v2.4.6
v2.5.0
v2.5.2-locref-mask-patch
v2.8.0
v2.8.1
v2.8.2
v2.8.3
v2.9.0
v2.11.0
v2.12.0
v2.12.2
v2.12.4
v2.13.0
v2.13.2
v2.14.0
v2.14.2
v2.15.0
v3.0.0
v3.0.1
v3.1.0
v3.2.0
```

To install a specific version, use

```
$ cryosparcm update --version=vXX.YY.ZZ[-branchname]
```

9.3.2 2. Before you update: complete or kill running jobs

Before you update the cryosparc softwares, you must wait for all the running cryosparc jobs completed (or kill your jobs). You also must check all the cryosparc related processes (i.e., supervisord, mongod, command_core, command_vis, command_rtp, webapp, app, liveapp) to be terminated successfully.

```
(example) userid@tem-cs-el7 $> cryosparcm stop
```

```
CryoSPARC is running.
Stopping cryoSPARC
app: stopped
command_core: stopped
command_rtp: stopped
command_vis: stopped
liveapp: stopped
webapp: stopped
database: stopped
Shut down
```

```
(example) userid@tem-cs-el7 $> ps aux | grep <userid> | grep cryosparc
userid 2449 0.0 0.0 152792 17480 ? Ss Jun24 0:18 python /tem/home/userid/.cryosparc/cryosparc2_
→masterdeps/anaconda/envs/cryosparc_master_env/bin/supervisord -c /tem/home/userid/.cryosparc/
```

(continues on next page)

(continued from previous page)

```
↪cryosparc2_master/supervisord.conf
userid 2472 1.2 0.0 1429268 57412 ? S Jun24 13:19 mongod --dbpath /tem/home/userid/.cryosparc/
↪cryosparc_database --port 39031 --oplogSize 64 --replSet meteor --nojournal --wiredTigerCacheSizeGB 4
userid 2900 0.2 0.0 860572 83028 ? S Jun24 2:23 python -c import cryosparc_command.command_
↪core as serv; serv.start(port=39032)
userid 4332 0.1 0.1 854192 236396 ? S Jun24 1:53 python -c import cryosparc_command.command_
↪vis as serv; serv.start(port=39033)
userid 4378 0.4 0.0 1190536 196316 ? S Jun24 4:59 python -c import cryosparc_command.command_
↪rtp as serv; serv.start(port=39035)
userid 5586 0.1 0.0 1331136 116972 ? S Jun24 1:17 /tem/home/userid/.cryosparc/cryosparc2_master/
↪cryosparc_webapp/nodejs/bin/node ./bundle/main.js
userid 5625 0.1 0.0 1049868 97640 ? S Jun24 1:22 /tem/home/userid/.cryosparc/cryosparc2_master/
↪cryosparc_app/nodejs/bin/node ./bundle/main.js
userid 5690 0.2 0.0 1326136 81432 ? S Jun24 2:12 /tem/home/userid/.cryosparc/cryosparc2_master/
↪cryosparc_liveapp/nodejs/bin/node ./bundle/main.js
```

(example) userid@tem-cs-el7 \$> kill -9 2449 2472 2900 4332 4378 5586 5625 5690

Find your own cryosparc unix socket files on /tmp directory, and if exists, delete the files using rm command.

```
userid@tem-cs-el7 $> cd /tmp
(example) userid@tem-cs-el7 $> ls -al | grep <userid> | grep sock

srwx-----. 1 userid      userid      0 Jun 24 16:39 cryosparc-supervisor-627a9991e2f2f069094732dfd78d1696.
↪sock
srwx-----. 1 userid      userid      0 Jun 24 16:39 mongodb-39031.sock

(example) userid@tem-cs-el7 $> rm cryosparc-supervisor-627a9991e2f2f069094732dfd78d1696.sock
(example) userid@tem-cs-el7 $> rm mongodb-39031.sock
```

9.3.3 3. Back-up cryosparc databases

We also highly recommend making a backup of your database as described below.

```
userid@tem-cs-el7 $> cryosparcm backup

Backing up to /tem/scratch/<GroupDir>/cryosparc/cryosparc_database/backup/cryosparc_backup_2021_04_20_
↪15h00.archive

CryoSPARC is not already running.

Starting the database in case it's not already running.
database: started

Executing mongodump.

2021-04-20T15:00:42.606+0900  writing admin.system.version to archive '/tem/scratch/<GroupDir>/
↪cryosparc/cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'
2021-04-20T15:00:42.608+0900  done dumping admin.system.version (1 document)
2021-04-20T15:00:42.609+0900  writing meteor.events to archive '/tem/scratch/<GroupDir>/cryosparc/
↪cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'
2021-04-20T15:00:42.617+0900  writing meteor.fs.files to archive '/tem/scratch/<GroupDir>/cryosparc/
↪cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'
```

(continues on next page)

(continued from previous page)

```
2021-04-20T15:00:42.617+0900 writing meteor.notifications to archive '/tem/scratch/<GroupDir>/.cryosparc/  
˓→cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'  
2021-04-20T15:00:42.618+0900 writing meteor.fs.chunks to archive '/tem/scratch/<GroupDir>/.cryosparc/  
˓→cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'  
2021-04-20T15:00:42.661+0900 done dumping meteor.notifications (315 documents)  
2021-04-20T15:00:42.661+0900 writing meteor.jobs to archive '/tem/scratch/<GroupDir>/.cryosparc/  
˓→cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'  
2021-04-20T15:00:42.692+0900 done dumping meteor.fs.files (8386 documents)  
2021-04-20T15:00:42.692+0900 writing meteor.cache_files to archive '/tem/scratch/<GroupDir>/.cryosparc/  
˓→cryosparc_database/backup/cryosparc_backup_2021_04_20_15h00.archive'  
2021-04-20T15:00:42.693+0900 done dumping meteor.jobs (166 documents)  
  
... (omit)  
  
2021-04-20T15:00:42.751+0900 done dumping meteor.file_index (0 documents)  
2021-04-20T15:00:42.755+0900 done dumping meteor.exposures (0 documents)  
2021-04-20T15:00:42.770+0900 done dumping meteor.events (25676 documents)  
2021-04-20T15:00:45.182+0900 [#####.....] meteor.fs.chunks 1535/8386 (18.3%)  
2021-04-20T15:00:48.182+0900 [#######.....] meteor.fs.chunks 3048/8386 (36.3%)  
2021-04-20T15:00:51.182+0900 [#####///////////.....] meteor.fs.chunks 5475/8386 (65.3%)  
2021-04-20T15:00:53.903+0900 [#####///////////#####.....] meteor.fs.chunks 8386/8386 (100.0%)  
2021-04-20T15:00:53.905+0900 done dumping meteor.fs.chunks (8386 documents)
```

Complete!

After backing up your cryosparc database, you should check the status of cryosparc daemons and stop them again.

```
userid@tem-cs-el7 $> cryosparcm status  
userid@tem-cs-el7 $> cryosparcm stop
```

9.3.4 4. Cryosparc master updates

To begin automatic master updates with the newest available version of cryoSPARC, just run

```
userid@tem-cs-el7 $> cryosparcm update  
  
CryoSPARC current version v4.0.0  
    update starting on Tue Apr 20 15:36:12 KST 2021  
  
No version specified - updating to latest version.  
  
=====  
Updating to version v4.1.0.  
=====  
  
CryoSPARC is not already running.  
If you would like to restart, use cryosparcm restart  
Removing previous downloads...  
Downloading master update...  
% Total % Received % Xferd Average Speed Time Time Current  
          Dload Upload Total Spent Left Speed  
0 0 0 0 0 0 0 0:00:02 --:--:-- 0  
100 785M 100 785M 0 0 2072k 0 0:06:27 0:06:27 --:--:-- 3897k  
Downloading worker update...
```

(continues on next page)

(continued from previous page)

```
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0:00:02 --:--:-- 0
100 1807M 100 1807M 0 0 2312k 0 0:13:20 0:13:20 --:--:-- 7988k
Done.
```

Update will now be applied to the master installation,
followed by worker installations on other nodes.

Deleting old files...

Extracting...

Done.

Installing latest master dependencies.

Checking dependencies...

Dependencies **for** python have changed - reinstalling...

Installing anaconda python...

PREFIX=/tem/scratch/<GroupDir>/.cryosparc/cryosparc_master/deps/anaconda
Unpacking payload ...

Solving environment: **done**

Package Plan

environment location: /tem/scratch/<GroupDir>/.cryosparc/cryosparc_master/deps/anaconda

added / updated specs:

- **_libgcc_mutex==0.1**=main
- **ca-certificates==2020.1.1=0**
- **certifi==2020.4.5.1=py37_0**
- **cffi==1.14.0=py37he30daa8_1**
- **chardet==3.0.4=py37_1003**
- **conda-package-handling==1.6.1=py37h7b6447c_0**
- **conda==4.8.3=py37_0**
- **cryptography==2.9.2=py37h1ba5d50_0**
- **idna==2.9=py_1**
- **ld_impl_linux-64==2.33.1=h53a641e_7**
- **libedit==3.1.20181209=hc058e9b_0**
- **libffi==3.3=he6710b0_1**
- **libgcc-ng==9.1.0=hdf63c60_0**
- **libstdcxx-ng==9.1.0=hdf63c60_0**
- **ncurses==6.2=he6710b0_1**
- **openssl==1.1.1g=h7b6447c_0**
- **pip==20.0.2=py37_3**
- **pycosat==0.6.3=py37h7b6447c_0**
- **pyparser==2.20=py_0**
- **pyopenssl==19.1.0=py37_0**
- **pysocks==1.7.1=py37_0**
- **python==3.7.7=hcff3b4d_5**
- **readline==8.0=h7b6447c_0**

(continues on next page)

(continued from previous page)

```
- requests==2.23.0=py37_0
- ruamel_yaml==0.15.87=py37h7b6447c_0
- setuptools==46.4.0=py37_0
- six==1.14.0=py37_0
- sqlite==3.31.1=h62c20be_1
- tk==8.6.8=hbc83047_0
- tqdm==4.46.0=py_0
- urllib3==1.25.8=py37_0
- wheel==0.34.2=py37_0
- xz==5.2.5=h7b6447c_0
- yaml==0.1.7=had09818_2
- zlib==1.2.11=h7b6447c_3
```

The following NEW packages will be INSTALLED:

```
_libgcc_mutex    pkgs/main/linux-64::_libgcc_mutex-0.1-main
ca-certificates  pkgs/main/linux-64::ca-certificates-2020.1.1-0
certifi          pkgs/main/linux-64::certifi-2020.4.5.1-py37_0
cffi              pkgs/main/linux-64::cffi-1.14.0-py37he30daa8_1
chardet          pkgs/main/linux-64::chardet-3.0.4-py37_1003
conda            pkgs/main/linux-64::conda-4.8.3-py37_0
conda-package-han~ pkgs/main/linux-64::conda-package-handling-1.6.1-py37h7b6447c_0
cryptography     pkgs/main/linux-64::cryptography-2.9.2-py37h1ba5d50_0
idna              pkgs/main/noarch::idna-2.9-py_1
ld_impl_linux-64 pkgs/main/linux-64::ld_impl_linux-64-2.33.1-h53a641e_7
libedit           pkgs/main/linux-64::libedit-3.1.20181209-hc058e9b_0
libffi             pkgs/main/linux-64::libffi-3.3-he6710b0_1
libgcc-ng         pkgs/main/linux-64::libgcc-ng-9.1.0-hdf63c60_0
libstdcxx-ng      pkgs/main/linux-64::libstdcxx-ng-9.1.0-hdf63c60_0
ncurses           pkgs/main/linux-64::ncurses-6.2-he6710b0_1
openssl           pkgs/main/linux-64::openssl-1.1.1g-h7b6447c_0
pip               pkgs/main/linux-64::pip-20.0.2-py37_3
pycosat          pkgs/main/linux-64::pycosat-0.6.3-py37h7b6447c_0
pycparser         pkgs/main/noarch::pycparser-2.20-py_0
pyopenssl        pkgs/main/linux-64::pyopenssl-19.1.0-py37_0
pysocks           pkgs/main/linux-64::pysocks-1.7.1-py37_0
python             pkgs/main/linux-64::python-3.7.7-hcff3b4d_5
readline          pkgs/main/linux-64::readline-8.0-h7b6447c_0
requests          pkgs/main/linux-64::requests-2.23.0-py37_0
ruamel_yaml       pkgs/main/linux-64::ruamel_yaml-0.15.87=py37h7b6447c_0
setuptools        pkgs/main/linux-64::setuptools-46.4.0=py37_0
six               pkgs/main/linux-64::six-1.14.0=py37_0
sqlite            pkgs/main/linux-64::sqlite-3.31.1-h62c20be_1
tk                pkgs/main/linux-64::tk-8.6.8-hbc83047_0
tqdm              pkgs/main/noarch::tqdm-4.46.0=py_0
urllib3           pkgs/main/linux-64::urllib3-1.25.8=py37_0
wheel             pkgs/main/linux-64::wheel-0.34.2=py37_0
xz                pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
yaml              pkgs/main/linux-64::yaml-0.1.7=had09818_2
zlib              pkgs/main/linux-64::zlib-1.2.11=h7b6447c_3
```

Preparing transaction: **done**

Executing transaction: **done**

installation finished.

(continues on next page)

(continued from previous page)

Done.
anaconda python installation successful.

Extracting all conda packages...

Done.
conda packages installation successful.

Main dependency installation completed. Continuing...

Completed.
Currently checking **hash for** mongodb
Dependencies **for** mongodb have not changed.
Completed dependency check.

Successfully updated master to version v4.1.0.

Starting cryoSPARC System master process..
CryoSPARC is not already running.
database: started
command_core: started
 command_core connection succeeded
 command_core startup successful
command_vis: started
command_rtp: started
 command_rtp connection succeeded
 command_rtp startup successful
webapp: started
app: started
liveapp: started

CryoSPARC master started.
From this machine, access cryoSPARC at
 <http://localhost:39030>
and access cryoSPARC Live at
 <http://localhost:39036>
please note the legacy cryoSPARC Live application is running at
 <http://localhost:39037>

From other machines on the network, access cryoSPARC at
 <http://tem-cs-e17.sdfarm.kr:39030>
and access cryoSPARC Live at
 <http://tem-cs-e17.sdfarm.kr:39036>

Startup can take several minutes. Point your browser to the address
and refresh **until** you see the cryoSPARC web interface.

Now updating worker nodes.

(continues on next page)

(continued from previous page)

All workers:

Done updating all worker nodes.
If any nodes failed to update, you can manually update them.
Cluster worker installations must be manually updated.

To update manually, copy the cryosparc_worker.tar.gz file into the cryosparc worker installation directory, and **then** run
 \$ bin/cryosparcw update
from inside the worker installation directory.

Or, you can update the master with a specific version.

```
userid@tem-cs-el7 $> cryosparcm update --version=vXX.YY.ZZ
```

After updating the master of your cryosparc instance, you should check the status of cryosparc daemons and stop them again in order to re-install the worker softwares.

```
userid@tem-cs-el7 $> cryosparcm status  
userid@tem-cs-el7 $> cryosparcm stop
```

9.3.5 5. Cryosparc worker updates

Since we adopt the clustered installation method for cryosparc instances, we shoud manually update the cryosparc worker. But simply with cryosparc worker updates (guided from cryosparc official site), you might face up with the version mismatch problem of CUDA SDK runtime libraries (the root cause is unkwnon now). So we decide to newly install all the cryosparc worker softwares to address this issues.

If you successully update the cryosparc master softwares above, you must find **cryosparc_worker.tar.gz** tar ball in **~/.cryosparc/cryosparc_master** directory.

```
userid@tem-cs-el7 $> cd ~/.cryosparc/cryosparc_master  
userid@tem-cs-el7 $> ls -al *.tar.gz  
-rw-r----- 1 userid userid 823226956 Apr 20 15:42 cryosparc_master.tar.gz  
-rw-r----- 1 userid userid 1895278500 Apr 20 15:56 cryosparc_worker.tar.gz
```

First, modity the name of the previous worker directory to that with .orig postfix and copy/uncompress the worker tar ball to .cryosparc directory.

If your master installation directory is “**cryosparc_master**”, use these commands.

```
userid@tem-cs-el7 $> cd /tem/scratch/<GroupDir>/cryosparc  
userid@tem-cs-el7 $> mv cryosparc_worker cryosparc_worker.orig  
userid@tem-cs-el7 $> cp /tem/scratch/<GroupDir>/cryosparc/cryosparc_master/cryosparc_worker.tar.gz /tem/  
    ↳scratch/<GroupDir>/cryosparc  
userid@tem-cs-el7 $> tar xvzf cryosparc_worker.tar.gz
```

Then, re-install all the cryosparc worker softwares with the followings (note that cryosparc version 3.2.0+ requires CUDA SDK 11.x+):

```
userid@tem-cs-el7 $> cd /tem/scratch/<GroupDir>/cryosparc/cryosparc_worker  
userid@tem-cs-el7 $> eval $(cryosparcm env)
```

(continues on next page)

(continued from previous page)

```
userid@tem-cs-el7 $> ./install.sh --license $CRYOSPARC_LICENSE_ID --cudapath /usr/local/cuda-11.2
***** CRYOSPARC SYSTEM: WORKER INSTALLER *****
```

Installation Settings:

```
License ID      : xxxxxxxxxxxxx
Root Directory   : /tem/scratch/<GroupDir>/cryosparc/cryosparc_worker
Standalone Installation : false
Version         : v4.1.0
```

```
*****
```

CUDA check..

```
Found nvidia-smi at /bin/nvidia-smi
```

```
CUDA Path was provided as /usr/local/cuda-11.2
```

```
Checking CUDA installation...
```

```
Found nvcc at /usr/local/cuda-11.2/bin/nvcc
```

```
The above cuda installation will be used but can be changed later.
```

```
*****
```

Setting up hard-coded config.sh environment variables

```
*****
```

Installing all dependencies.

```
Warning: conda environment not found; this indicates that a cryoSPARC installation is either incomplete or in □
→progress
```

```
Checking dependencies...
```

```
Dependencies for python have changed - reinstalling...
```

```
-----  
Installing anaconda python...
```

```
PREFIX=/tem/scratch/<GroupDir>/cryosparc/cryosparc_worker/deps/anaconda
Unpacking payload ...
```

```
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: /tem/scratch/<GroupDir>/cryosparc/cryosparc_worker/deps/anaconda
```

```
added / updated specs:
```

- `_libgcc_mutex==0.1=main`
- `ca-certificates==2020.1.1=0`
- `certifi==2020.4.5.1=py37_0`
- `cffi==1.14.0=py37he30daa8_1`
- `chardet==3.0.4=py37_1003`
- `conda-package-handling==1.6.1=py37h7b6447c_0`
- `conda==4.8.3=py37_0`
- `cryptography==2.9.2=py37h1ba5d50_0`
- `idna==2.9=py_1`
- `ld_impl_linux-64==2.33.1=h53a641e_7`
- `libedit==3.1.20181209=hc058e9b_0`
- `libffi==3.3=he6710b0_1`

(continues on next page)

(continued from previous page)

```
- libgcc-ng==9.1.0=hdf63c60_0
- libstdcxx-ng==9.1.0=hdf63c60_0
- ncurses==6.2=he6710b0_1
- openssl==1.1.1g=h7b6447c_0
- pip==20.0.2=py37_3
- pycosat==0.6.3=py37h7b6447c_0
- pyparser==2.20=py_0
- pyopenssl==19.1.0=py37_0
- pysocks==1.7.1=py37_0
- python==3.7.7=hcff3b4d_5
- readline==8.0=h7b6447c_0
- requests==2.23.0=py37_0
- ruamel_yaml==0.15.87=py37h7b6447c_0
- setuptools==46.4.0=py37_0
- six==1.14.0=py37_0
- sqlite==3.31.1=h62c20be_1
- tk==8.6.8=hbc83047_0
- tqdm==4.46.0=py_0
- urllib3==1.25.8=py37_0
- wheel==0.34.2=py37_0
- xz==5.2.5=h7b6447c_0
- yaml==0.1.7=had09818_2
- zlib==1.2.11=h7b6447c_3
```

The following NEW packages will be INSTALLED:

```
_libgcc_mutex    pkgs/main/linux-64::_libgcc_mutex-0.1-main
ca-certificates  pkgs/main/linux-64::ca-certificates-2020.1.1-0
certifi          pkgs/main/linux-64::certifi-2020.4.5.1-py37_0
cffi              pkgs/main/linux-64::cffi-1.14.0-py37he30daa8_1
chardet          pkgs/main/linux-64::chardet-3.0.4-py37_1003
conda            pkgs/main/linux-64::conda-4.8.3-py37_0
conda-package-han~ pkgs/main/linux-64::conda-package-handling-1.6.1-py37h7b6447c_0
cryptography     pkgs/main/linux-64::cryptography-2.9.2-py37h1ba5d50_0
idna              pkgs/main/noarch::idna-2.9-py_1
ld_impl_linux-64 pkgs/main/linux-64::ld_impl_linux-64-2.33.1-h53a641e_7
libedit           pkgs/main/linux-64::libedit-3.1.20181209-hc058e9b_0
libffi            pkgs/main/linux-64::libffi-3.3-he6710b0_1
libgcc-ng         pkgs/main/linux-64::libgcc-ng-9.1.0-hdf63c60_0
libstdcxx-ng      pkgs/main/linux-64::libstdcxx-ng-9.1.0-hdf63c60_0
ncurses           pkgs/main/linux-64::ncurses-6.2-he6710b0_1
openssl           pkgs/main/linux-64::openssl-1.1.1g-h7b6447c_0
pip               pkgs/main/linux-64::pip-20.0.2-py37_3
pycosat          pkgs/main/linux-64::pycosat-0.6.3=py37h7b6447c_0
pyparser          pkgs/main/noarch::pyparser-2.20=py_0
pyopenssl         pkgs/main/linux-64::pyopenssl-19.1.0=py37_0
pysocks           pkgs/main/linux-64::pysocks-1.7.1=py37_0
python             pkgs/main/linux-64::python-3.7.7-hcff3b4d_5
readline          pkgs/main/linux-64::readline-8.0=h7b6447c_0
requests          pkgs/main/linux-64::requests-2.23.0=py37_0
ruamel_yaml       pkgs/main/linux-64::ruamel_yaml-0.15.87=py37h7b6447c_0
setuptools        pkgs/main/linux-64::setuptools-46.4.0=py37_0
six               pkgs/main/linux-64::six-1.14.0=py37_0
sqlite            pkgs/main/linux-64::sqlite-3.31.1-h62c20be_1
tk                pkgs/main/linux-64::tk-8.6.8-hbc83047_0
```

(continues on next page)

(continued from previous page)

```
tqdm      pkgs/main/noarch::tqdm-4.46.0-py_0
urllib3    pkgs/main/linux-64::urllib3-1.25.8-py37_0
wheel      pkgs/main/linux-64::wheel-0.34.2-py37_0
xz         pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
yaml       pkgs/main/linux-64::yaml-0.1.7-had09818_2
zlib       pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3
```

Preparing transaction: **done**

Executing transaction: **done**

installation finished.

Done.

anaconda python installation successful.

Extracting all conda packages...

Done.

conda packages installation successful.

Preparing to install all pip packages...

Processing ./deps_bundle/python/python_packages/pip_packages/pycuda-2020.1.tar.gz

Skipping wheel build **for** pycuda, due to binaries being disabled **for** it.

Installing collected packages: pycuda

 Running setup.py install **for** pycuda ... **done**

Successfully installed pycuda-2020.1

Done.

pip packages installation successful.

Main dependency installation completed. Continuing...

Completed.

Currently checking **hash for** ctffind

Dependencies **for** ctffind have changed - reinstalling...

ctffind **4.1.10** installation successful.

Completed.

Currently checking **hash for** cudnn

Dependencies **for** cudnn have changed - reinstalling...

cudnn **8.1.0.77** **for** CUDA **11** installation successful.

Completed.

Currently checking **hash for** gctf

Dependencies **for** gctf have changed - reinstalling...

Gctf v1.06 installation successful.

Completed.

Completed dependency check.

(continues on next page)

(continued from previous page)

***** CRYOSPARC WORKER INSTALLATION COMPLETE *****

In order to run processing jobs, you will need to connect this worker to a cryoSPARC master.

9.3.6 6. Running the newer cryosparc instance

All the cryosparc master and worker updates has completed. So, you need to re-execute cryosparc instance daemons (assume userid's CRYOSPARC_BASE_PORT is 39030).

```
userid@tem-cs-el7 $> cd ~/  
userid@tem-cs-el7 $> cryosparcm env  
userid@tem-cs-el7 $> cryosparcm status  
userid@tem-cs-el7 $> cryosparcm start
```

```
Starting cryoSPARC System master process..  
CryoSPARC is not already running.  
database: started  
command_core: started  
    command_core connection succeeded  
    command_core startup successful  
command_vis: started  
command_rtp: started  
    command_rtp connection succeeded  
    command_rtp startup successful  
webapp: started  
app: started  
liveapp: started
```

```
CryoSPARC master started.  
From this machine, access cryoSPARC at  
    http://localhost:39030  
and access cryoSPARC Live at  
    http://localhost:39036  
please note the legacy cryoSPARC Live application is running at  
    http://localhost:39037
```

```
From other machines on the network, access cryoSPARC at  
    http://tem-cs-el7.sdfarm.kr:39030  
and access cryoSPARC Live at  
    http://tem-cs-el7.sdfarm.kr:39036
```

Startup can take several minutes. Point your browser to the address and refresh **until** you see the cryoSPARC web interface.

CHAPTER
TEN

APPENDIX : APPLICATION FORM & MATERIALS 

10.1 GSDC TEM Application Form

If you want to get user account to access TEM farm, please download this Application Form, fill-out the blanks, sign and send it to GSDC service managers.

- [GSDC TEM Application Form](#)

10.2 Tutorial

- [GSDC TEM 소개 자료 \(korean\)](#)
- [GSDC TEM 분석 팜 사용자 매뉴얼 \(korean, ppt, 2020 SNU CryoEM Hands-On Workshop\)](#)

**CHAPTER
ELEVEN**

CONTACTS

If you have any questions on using TEM service farm, please do not hesitate to contact us.

- Il-Yeon Yeo (ilyeon9 at kisti.re.kr) Tel: +82-42-869-0658
- Jung-Lok Yu (junglok.yu at kisti.re.kr) Tel: +82-42-869-0622

Since Mar. 2021, we use GSDC's ticketing system to support all the technical problems for users (including TEM application form), so feel free to create tickets.

- E-mail address : gsdc-support at kisti.re.kr (use [TEM] prefix in e-mail title)