

Simple guide to slurm job arrays on Zeus

For use when you need to run the same program over a number of files. In non-supercomputing environments, you might use a loop or gnu parallel. However, we can make slurm perform parallelisation for us with minimal effort. This has been tested on Zeus. The maximum number of jobs that can be created with a single array is 1000. However, the workq partition on Zeus is limited to a max of 512 jobs in the queue, and 16 jobs running concurrently. Therefore, do not create an array that will spawn more than 512 jobs. With the limit on number of concurrent jobs, a max of 16 will run at a time, with a new one spawning as an older one completes.

Guided example

We will make use of two scripts- a job script that performs the task we need, and a 'helper' script that sets up the array and launches the job script. The example program we will use is ExpansionHunter DeNovo, a bioinformatic tool for detecting repeat expansions from sequencing data.

Part 1- Making your job script

A template is provided below. This will print out helpful information into your log files to help you keep track of your jobs.

array_job.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --cpus-per-task=1
#SBATCH --time=1:00:00
#SBATCH --job-name=MYJOB
#SBATCH --partition=workq
#SBATCH --account=MYACCOUNT
#SBATCH --output=MYJOB-%j.log
#SBATCH --error=MYJOB-%J.log

#Do not edit the echo sections
echo "All jobs in this array have:"
echo "- SLURM_ARRAY_JOB_ID=${SLURM_ARRAY_JOB_ID}"
echo "- SLURM_ARRAY_TASK_COUNT=${SLURM_ARRAY_TASK_COUNT}"
echo "- SLURM_ARRAY_TASK_MIN=${SLURM_ARRAY_TASK_MIN}"
echo "- SLURM_ARRAY_TASK_MAX=${SLURM_ARRAY_TASK_MAX}"
echo "This job in the array has:"
echo "- SLURM_JOB_ID=${SLURM_JOB_ID}"
echo "- SLURM_ARRAY_TASK_ID=${SLURM_ARRAY_TASK_ID}"

# alter the following line to suit your files. It will grab all files matching whatever regular expression you
# provide.
FILES=$(ls -l *.bam)

# grabs our filename from a directory listing
FILENAME=${FILES[${SLURM_ARRAY_TASK_ID}]}
echo "My input file is ${FILENAME}"

#load modules
module load singularity

#set variables
basedir=$(pwd)
container=/group/$MYGROUP/$USER/expansion-hunter-denovo_v0.8.7.sif
ref=Homo_sapiens_assembly38.fasta

#job script
singularity exec ${container} /ExpansionHunterDenovo profile \
  --reads ${basedir}/${FILENAME} \
  --reference ${basedir}/${ref} \
  --output-prefix ${basedir}/str-profiles/${FILENAME} \
  --min-anchor-mapq 50 \
  --max-irr-mapq 40
```

Part 2- Making your helper script.

SBATCH_HELPER.SH

```
#!/bin/bash

# get count of files in this directory that match provided regular expression. Edit to suit your files.
NUMFILES=$(ls -l *.bam | wc -l)
# subtract 1 as we have to use zero-based indexing (first element is 0). Do not edit.
ZBNUMFILES=$((NUMFILES - 1))
# submit array of jobs to SLURM. Change to match your job script name.
if [ $ZBNUMFILES -ge 0 ]; then
  sbatch --array=0-$ZBNUMFILES array_job.sh
else
  echo "No jobs to submit, since no input files in this directory."
fi
```

When you are satisfied with the scripts, launch them with "sbatch sbatch_helper.sh". If you watch your jobqueue with "watch squeue -u USERNAME" you should see your helper script running, and then see it spawn a number of jobs.



What if I have more than 1000 jobs

If you are running something with a lot of jobs, another option is job packing, described here: [job packing with mpibash and libcircle](#)

Remember we are available to assist, so if you have a very complex job, please contact the helpdesk for advice.

Further info on job arrays can be found here: [Example Workflows](#)