Running HPC array jobs

Slurm

To run array jobs the header or sbatch call should contain information relating to the size of the array.

The flag to use is \-a, \-\-array with the argument being the steps as a comma separated list of individual indexes and index ranges e.g. --array=1-6,10 to run for array values of 1, 2, 3, 4, 5, 6, 10.

Finally a step size can be given following a colon e.g. --array=1-15:4, which would run for 1, 5, 9, 13.

The maximum number of steps to run at once can be given following a % symbol, e.g. --array=1-15%4 would allow a maximum of 4 tasks to run at a time on the indexes of 1 through to 15.

The array index of the current job step can be accessed using $SLURM_ARRAY_TASK_ID

Other available array related variables are available below:

SLURM_ARRAY_TASK_COUNT: Total number of tasks in a job array.

SLURM_ARRAY_TASK_MAX: Job array's maximum ID (index) number.

SLURM_ARRAY_TASK_MIN: Job array's minimum ID (index) number.

SLURM_ARRAY_TASK_STEP: Job array's index step size.

SLURM_ARRAY_JOB_ID: Job array's master job ID number.

The given slurm script will be run once for each index provided, with the only changes being the value of the $SLURM_ARRAY_TASK_ID.

This can be useful in situations like performing an operation on each line in a file in parallel instead of in series.

The following script can be used for performing an action on every line of a file:

```
#!/usr/bin/env bash

#SBATCH --time=00:30:00
#SBATCH --array=1-10
#SBATCH --ntasks=1

suffix="q;d"
line_to_check=$SLURM_ARRAY_TASK_ID$suffix
line_data=`sed "$line_to_check" file_to_process.txt`
echo "Got line data $line_data"
python script_to_run.py line_data
```

In the above script 10 step will be created, each with one task/core for a wall clock time of 30mins.

Each line of the file file_to_process.txt will be passed as an argument to the script script_to_run.py