Notes on SBATCH and software specifics

SBATCH

```
#!/bin/bash
#SBATCH --job-name=<JOB-NAME>     # Job name
#SBATCH --mail-type=ALL                  # Mail (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<EMAIL>       # Where to send mail
#SBATCH --nodes=<n>              # Number of nodes requested
#SBATCH --ntasks=<n>             # Number of CPUs
#SBATCH --mem=<n>gb               # Memory limit
#SBATCH --time=<00:00:00>         # Time limit hrs:min:sec
#SBATCH --partition=compute       # Partition/queue requested
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.out       # Standard
output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/<JOB-NAME>.%j.err        # Standard
error log
```

#SBATCH --job-name=<JOB-NAME>

Enter your job name below. On Fiji, %x will be replaced by the JOB-NAME in the standard output and error.

#SBATCH --mail-type=ALL

#SBATCH --mail-user=<EMAIL>

!!IMPORTANT!! Enter the email where job updates will go and select which updates you'd like to see. If you borrowed a script, it is especially important to change this

#SBATCH --nodes=<n>

Select the number of nodes you'd like to request for your job to run on. The default is 1

#SBATCH --ntasks=<n>

Number of CPUs (processer cores i.e. tasks) If none of your commands are parallelized, then only one CPU needed.

#SBATCH --time=<00:00:00>

Time Limit hrs:min:sec -- short < 24:00:00, long >/= 24:00:00, high mem > 50gb

#SBATCH --partition=compute

Designate partition/queue request on the server. This is useful for job assignment in the queue and will make your fellow server users happier. For Fiji -p short or -p long is typically used.

#SBATCH --partition=compute

Assign the memory limit -- this is something you have to play with in order to gauge how much you need. 200gb is relatively high but necessary for mapping. 1gb is good limit for FastQC jobs. Assigning lower memory limits to simpler jobs will again make the queue more efficient, so take the time to lower this if you anticipate needing less. You will get an error message if you exceed your memory limit and the job will stop.

#SBATCH --output=</path/to/your/directory/><job_name>_%j.out

#SBATCH --error=</path/to/your/directory/><job_name>_%j.err

Assign error and output files. There are a number of ways of doing this, but typically I will create an error and output folder and include %x (will be replaced with the job name, on amazon instance %x does not work so we will replace it with jobname) and %j (will be replaced with the job number e.g. the number assigned in the queue when you sbatch this script) as the output file name. You can look online for a number of different file designation options.

SOFTWARE SPECIFIC

rsync (command, not software)

- rsync -r user@remote.host:/path/to/fastqc/outdir/ /path/to/local/storage/

wc (command, not software)

- wc [options] filenames
- wc returns number of lines, number of words, number of bytes

module spider <program>

- For information, including how to load the program.

FastQC

Quality control of FASTQ files

- fastqc --format <format> --threads <n> <input_file> -o <output_file>
- module load fastqc/0.11.5
- FastQC can multi-thread. 1 node, 1 task or processor, 8gb for memory and 1 hour for wall time should be enough.

Trimmomatic

- Single end (SE)
  java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar SE [-threads <n>] [-phred33 | -phred64] [-trimlog <output_trimlog>] <input_file> <output_file> ILLUMINACLIP:<path_adapters_fasta>:<seed_mismatches>:<palindrome_clip_threshold>:<simple_clip_threshold> LEADING:<quality> TRAILING:<quality> SLIDINGWINDOW:<window_size>:<required_quality> MINLEN:<length>
- Pair end (PE)
  java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar PE [-threads <n>] [-phred33 | -phred64] [-trimlog <output_trimlog>] <input_file1> <input_file2> <output_fileP1> <output_fileU1> <output_fileP2> <output_fileU2> ILLUMINACLIP:<path_adapters_fasta>:<seed_mismatches>:<palindrome_clip_threshold>:<simple_clip_threshold> LEADING:<quality> TRAILING:<quality> SLIDINGWINDOW:<window_size>:<required_quality> MINLEN:<length>
- module load trimmomatic/0.36
- Trimmomatic can multi-thread, or use multiple processors per input file. 1 node, 8 tasks or processors, 8gb for memory and 1 hour for walltime should be enough.

Hisat2

- hisat2 [options] -x <genome_index> -U <input_fastq> > <output_sam>
- module load hisat2/2.1.0
- Hisat2 can use multiple processors per input file. 1 node, 4 tasks/processors/CPUs, 5 Gb for memory and 5 minutes for wall-time should be enough.

Samtools

- module load samtools/1.8
- samtools **view** to convert sam → bam:
  samtools view [options] <output_bam> > <input.sam>
- samtools **sort** sorts bam reads
  samtools sort [options] <input_bam> > <output_sorted.bam>
- samtools **index** index the bam file
  samtools index <input_sorted.bam> > <output_sorted.bam.bai>

Resource links:

1. FastQC guide: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

2. QC fail. Useful for looking through different FastQC plots of sequencing libraries that failed: https://sequencing.qcfail.com/software/fastqc/
3. Trimmomatic guide: http://www.usadellab.org/cms/?page=trimmomatic
4. BBDuk guide: https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbduk-guide/
5. Mapping via HISAT2 guide: https://ccb.jhu.edu/software/hisat2/manual.shtml
6. Illumina igenome. Provides packages for genome for model organism: https://support.illumina.com/sequencing/sequencing_software/igenome.html
7. Samtools guide: https://www.htslib.org/doc/samtools.html