Running jobs on CBRG Servers

There are 3 basic ways of running jobs on a computer:

1. foreground
2. background
3. batch/in a queue

Foreground is the most common mode of operation, where you type in a command at the prompt for immediate execution, then wait for it to complete before executing another command. The printed output, if any, appears on the screen, followed by another prompt. This mode of operation is normally used for quick file manipulation operations, file editing and jobs like interactive graphics, where the input to the program depends on a previous output.

Background mode is the similar as foreground, with the exception that you don't wait before executing the next command. Under Unix you just add an "**&**" on the end of the command line. You must ensure that all output from the job has been redirected to a file, otherwise it will be lost. Foreground jobs can be 'backgrounded' by first stopping them with the *Ctrl-Z* character, and then using the **bg** command. Background jobs can be brought back into foreground with the **fg** command (unless you logged out since starting the job.
The use of background mode for long computational runs is thought to be *anti-social* because it competes with other users' foreground and batch jobs for memory and CPU. If you have long jobs to run you should be using *batch mode* aka *run in the queue*.

Batch mode ('in the queue') is recommended for long computational jobs where all the input can be provided in files. Jobs are submitted to queues that are common to all users, and so may or may not be run immediately, depending on the number of jobs already in a queue. The CBRG use 'SUN Grid Engine' as the batch-queuing system.

N.B. If you intend to run any programs for manipulating NGS date (*Bowtie*, *Tophat*, *Cufflinks*, etc) you should run these in batch mode (*in the queue*)

How to submit your job to the queue

To run a job in the queue you need to use a batch script file that contains all the commands and environment requests that you want for this job.

There is a file available on deva for you to copy to the directory from where you want to run the job (your 'working directory')

Copy this file into your working directory:

On the command line type:

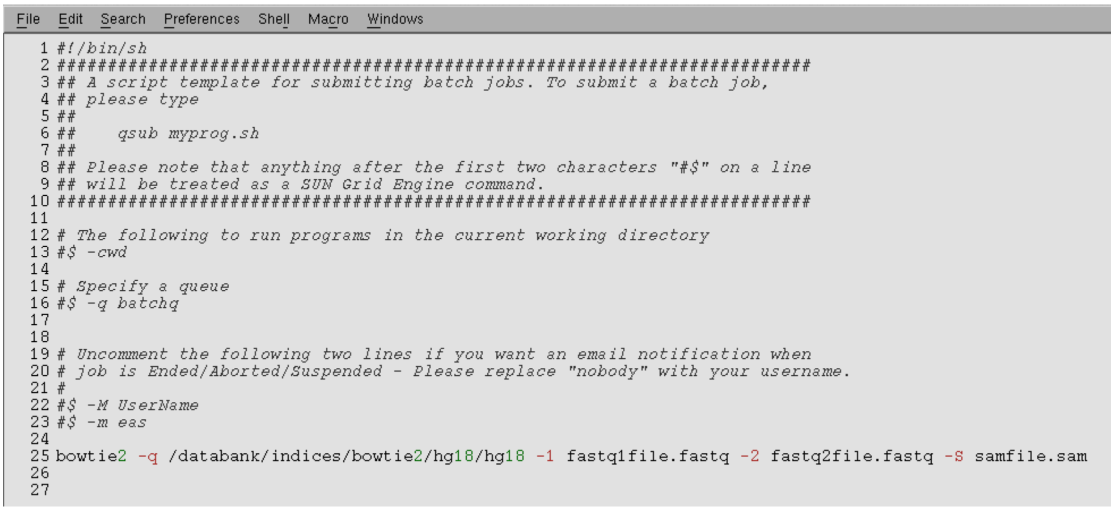
```
cp /package/cbrg/templates/qsub.sh .
```

Note the space followed by full stop (.) at the end of the command

You will need to edit this file so open it with a text editor like `nedit` or *Komodo*

Eg, type: **nedit qsub.sh**

This will open a new window:



```
#!/bin/sh
##########################################################################
## A script template for submitting batch jobs. To submit a batch job,
## please type
##
##     qsub myprog.sh
##
## Please note that anything after the first two characters "#$" on a line
## will be treated as a SUN Grid Engine command.
##########################################################################

# The following to run programs in the current working directory
#$ -cwd

# Specify a queue
#$ -q batchq


# Uncomment the following two lines if you want an email notification when
# job is Ended/Aborted/Suspended - Please replace "nobody" with your username.
#
#$ -M UserName
#$ -m eas

bowtie2 -q /databank/indices/bowtie2/hg18/hg18 -1 fastq1file.fastq -2 fastq2file.fastq -S samfile.sam


```

Lets look at what makes up this script.

The lines that begin **#$** will be read by the batch queueing system and the lines that start **#** are comments that will be ignored.

`#$ -cwd` This means run the program specified in the current working directory

`#$ -q batchq` This specifies which queue you wish to use – leave this as `batchq` unless advised by CBRG

The next 2 lines will give you email notification when your job has finished or if there is a problem:

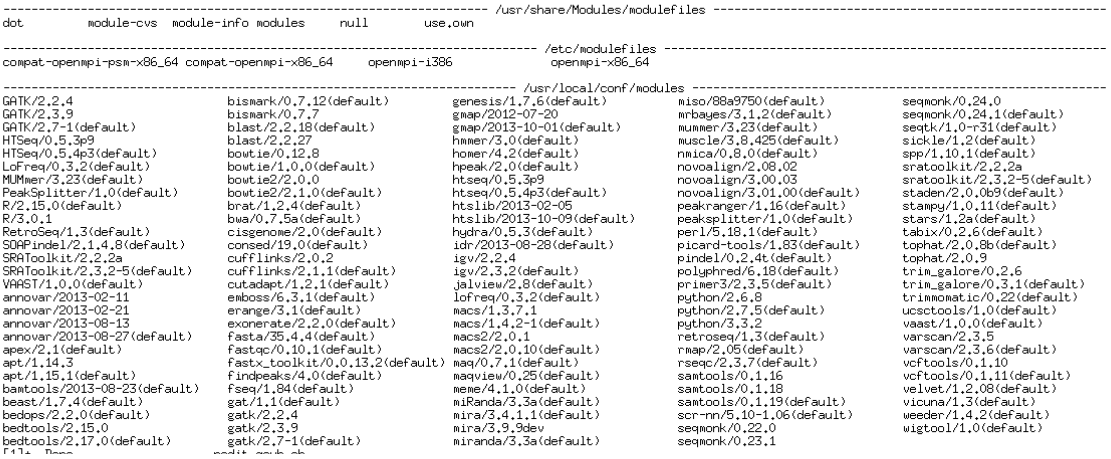
`#$ -M UserName` Change this to your CBRG username
`#$ -m eas` Leave this as it is

Under these 2 commands you need to add the *module* and the *command* you wish to run (explained below)

Modules on deva

Modules are useful in managing different versions of applications and for setting up variables required by a program. On deva usually the most recent version of a program is the default. It is useful to note which version of a program you are using.

On the command line type: **module avail**



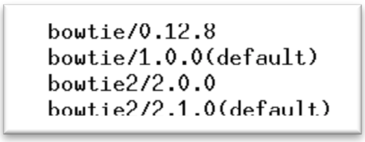
```
---------------------------------------------------------------------- /usr/share/Modules/modulefiles ----------------------------------------------------------------------
dot         module-cvs  module-info modules     null        use.own

----------------------------------------------------------------------------- /etc/modulefiles -----------------------------------------------------------------------------
compat-openmpi-psm-x86_64 compat-openmpi-x86_64     openmpi-i386              openmpi-x86_64

-------------------------------------------------------------------------- /usr/local/conf/modules --------------------------------------------------------------------------
GATK/2.2.4                      bismark/0.7.12(default)         genesis/1.7.6(default)          miso/88a9750(default)           seqmonk/0.24.0
GATK/2.3.9                      bismark/0.7.7                   gmap/2012-07-20                 mrbayes/3.1.2(default)          seqmonk/0.24.1(default)
GATK/2.7-1(default)             blast/2.2.18(default)           gmap/2013-10-01(default)        mummer/3.23(default)            seqtk/1.0-r31(default)
HTSeq/0.5.3p9                   blast/2.2.27                    hmmer/3.0(default)              muscle/3.8.425(default)         sickle/1.2(default)
HTSeq/0.5.4p3(default)          bowtie/0.12.8                   homer/4.2(default)              nmica/0.8.0(default)            spp/1.10.1(default)
LoFreq/0.3.2(default)           bowtie/1.0.0(default)           hpeak/2.0(default)              novoalign/2.08.02               sratoolkit/2.2.2a
MUMmer/3.23(default)            bowtie2/2.0.0                   htseq/0.5.3p9                   novoalign/3.00.03               sratoolkit/2.3.2-5(default)
PeakSplitter/1.0(default)       bowtie2/2.1.0(default)          htseq/0.5.4p3(default)          novoalign/3.01.00(default)      staden/2.0.0b9(default)
R/2.15.0(default)               brat/1.2.4(default)             htslib/2013-02-05               peakranger/1.16(default)        stampy/1.0.11(default)
R/3.0.1                         bwa/0.7.5a(default)             htslib/2013-10-09(default)      peaksplitter/1.0(default)       stars/1.2a(default)
RetroSeq/1.3(default)           cisgenome/2.0(default)          hydra/0.5.3(default)            perl/5.18.1(default)            tabix/0.2.6(default)
SOAPindel/2.1.4.8(default)      consed/19.0(default)            idr/2013-08-28(default)         picard-tools/1.83(default)      tophat/2.0.8b(default)
SRAToolkit/2.2.2a               cufflinks/2.0.2                 igv/2.2.4                       pindel/0.2.4t(default)          tophat/2.0.9
SRAToolkit/2.3.2-5(default)     cufflinks/2.1.1(default)        igv/2.3.2(default)              polyphred/6.18(default)         trim_galore/0.2.6
VAAST/1.0.0(default)            cutadapt/1.2.1(default)         jalview/2.8(default)            primer3/2.3.5(default)          trim_galore/0.3.1(default)
annovar/2013-02-11              emboss/6.3.1(default)           lofreq/0.3.2(default)           python/2.6.8                    trimmomatic/0.22(default)
annovar/2013-02-21              erange/3.1(default)             macs/1.3.7.1                    python/2.7.5(default)           ucsctools/1.0(default)
annovar/2013-08-13              exonerate/2.2.0(default)        macs/1.4.2-1(default)           python/3.3.2                    vaast/1.0.0(default)
annovar/2013-08-27(default)     fasta/35.4.4(default)           macs2/2.0.1                     retroseq/1.3(default)           varscan/2.3.5
apex/2.1(default)               fastqc/0.10.1(default)          macs2/2.0.10(default)           rmap/2.05(default)              varscan/2.3.6(default)
apt/1.14.3                      fastx_toolkit/0.0.13.2(default) maq/0.7.1(default)              rseqc/2.3.7(default)            vcftools/0.1.10
apt/1.15.1(default)             findpeaks/4.0(default)          maqview/0.25(default)           samtools/0.1.16                 vcftools/0.1.11(default)
bamtools/2013-08-23(default)    fseq/1.84(default)              meme/4.1.0(default)             samtools/0.1.18                 velvet/1.2.08(default)
beast/1.7.4(default)            gat/1.1(default)                miRanda/3.3a(default)           samtools/0.1.19(default)        vicuna/1.3(default)
bedops/2.2.0(default)           gatk/2.2.4                      mira/3.4.1.1(default)           scr-nn/5.10-1.06(default)       weeder/1.4.2(default)
bedtools/2.15.0                 gatk/2.3.9                      mira/3.9.9dev                   seqmonk/0.22.0                  wigtool/1.0(default)
bedtools/2.17.0(default)        gatk/2.7-1(default)             miranda/3.3a(default)           seqmonk/0.23.1
[1]+  Done                    gedit qsub.sh
```

You should see the list of modules available as above.

Lets look at *Bowtie* as an example. You can see in the list that there are 2 versions for each *Bowtie* and *Bowtie2*

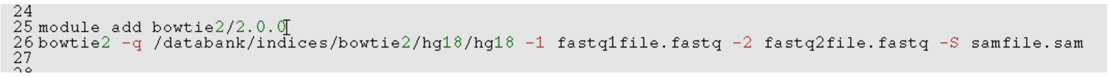


```
bowtie/0.12.8
bowtie/1.0.0(default)
bowtie2/2.0.0
bowtie2/2.1.0(default)
```

The lastest version of *Bowtie2*, 2.1.0, is set as the default. If you wanted to use version 2.0.0 you could do so by adding the following to the qsub script

Add to the qsub.sh file, eg **`module add module_name`**

This sets up all the variables required by *Bowtie2, 2.0.0.* Below this add the command to run the program:



```

module add bowtie2/2.0.0
bowtie2 -q /databank/indices/bowtie2/hg18/hg18 -1 fastq1file.fastq -2 fastq2file.fastq -S samfile.sam

```

Remember the files you specify must be in the same directory from which you have the qsub script or you must specify the full path.

Now save this qsub file and give it a new name so you can identify it, eg **`bowtie_Oct13.sh`**

To submit your job to the queue simply type the following onto the command line:

`qsub bowtie_Oct13.sh`

You should see something like:

```
Your job 5389 ("qsub.sh") has been submitted
```

To see jobs in the queue type, **`qstat`**

You should receive an email when your job has finished. Two new files will have been produced:

`qsub.sh.o5389` contains output
`qsub.sh.e5389` contains any error messages