Aligning your reads to the genome: from FastQ to raw counts

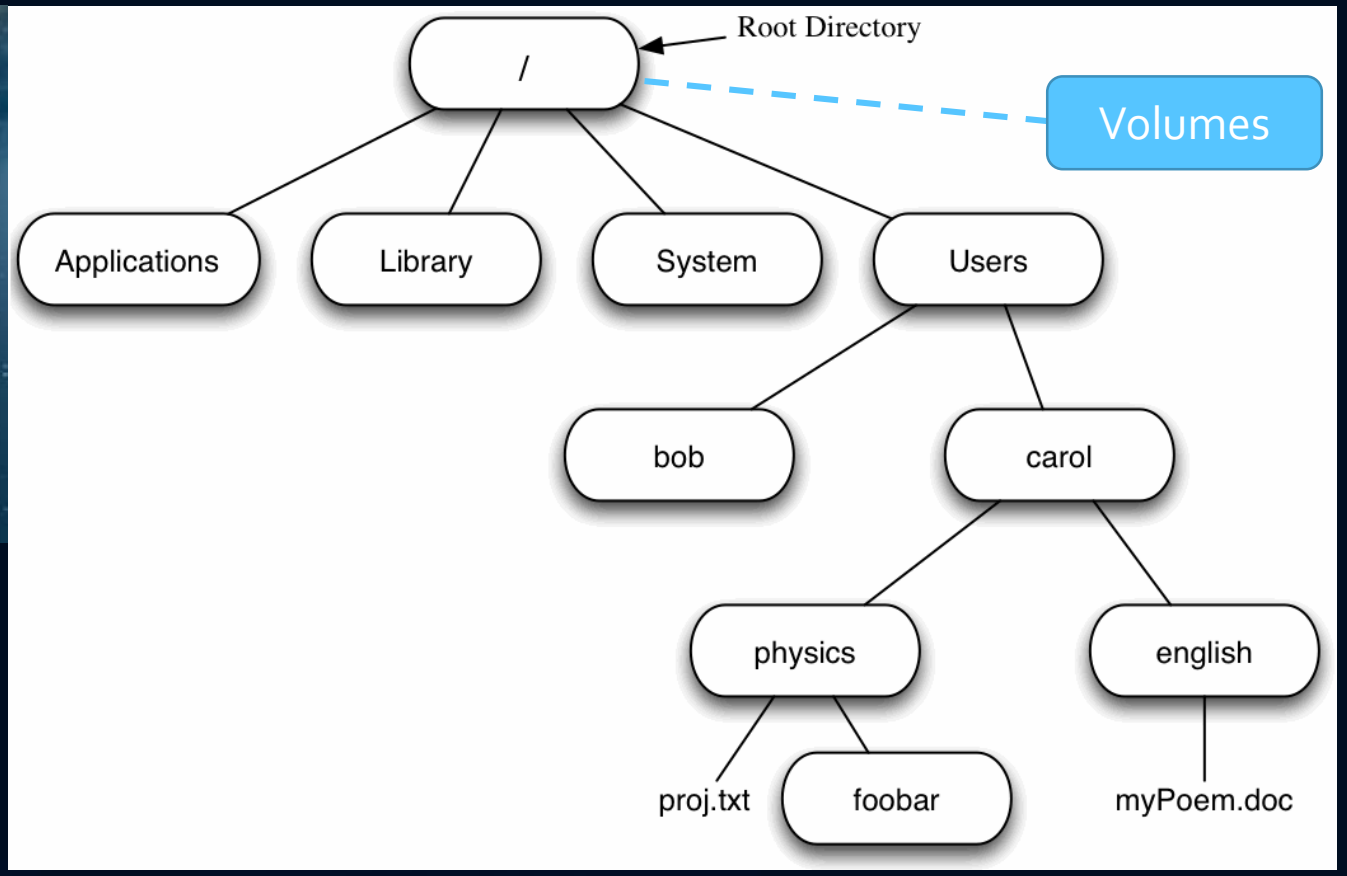
AN INTRODUCTION TO UNIX COMMAND LINE & SHELL SCRIPTING

RADIA JOHNSON, PH.D.

Outline

- Basics of UNIX command line
- Basics of SHELL scripting
- Launching **qsub** jobs on Calcul Quebec clusters
- Run STAR on sample RNA seq data
- Run HTSeq on SAM files

Navigating a file system (UNIX)



In the terminal

```
Last login: Thu Oct 20 12:31:39 on ttys001
Welcome radiamariejohnson, the current time is 13:14:21 10/24/16
web028074:ODC_data radiamariejohnson$ ls /
User Information            System/                     cores/                      home/
sw/                         etc                         Users/                      Library/
opt/                        tmp                         Network/                    Applications/
Dropbox/                    var                         usr/                        Volumes/
installer.failurerequests   sbin/                       dev/
bin/                        private/                    net/
web028074:ODC_data radiamariejohnson$
```

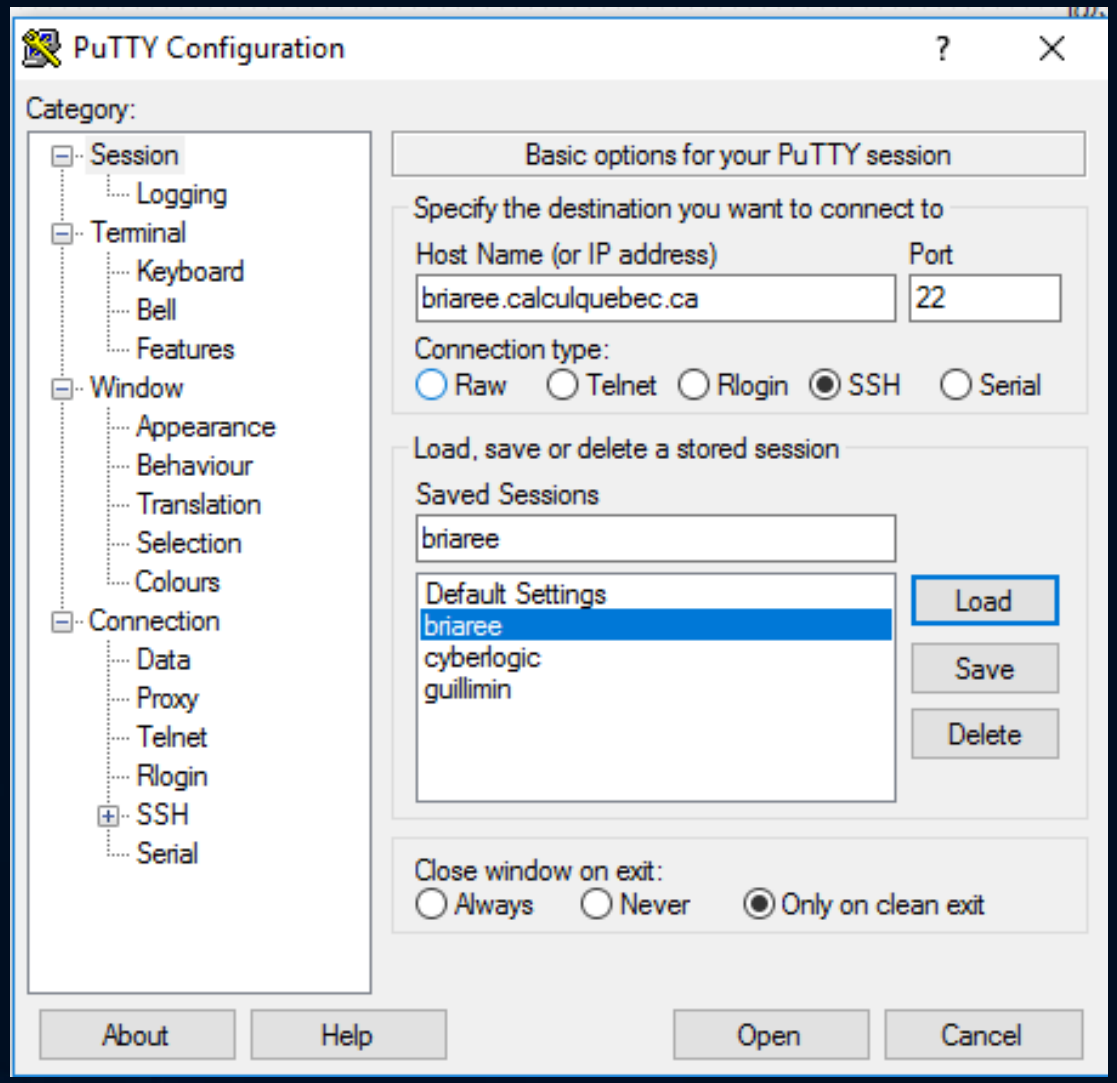
Login to cluster

Putty

- Host Name: briaree.calculquebec.ca
- Port: 22

Unix

- ssh username@briaree.calculquebec.ca



Basic UNIX/LINUX command lines

- pwd => *print working directory*
- cd => *change directory*
- ls => *list*
- less => *view files (doesn't load the full file – best for large files)*
- more = > *view files (prints to terminal)*
- man = > *manual*

In the terminal

```
web028074:~ radiamariejohnson$ pwd
/Users/radiamariejohnson
web028074:~ radiamariejohnson$ cd GCRCsession2/
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt
web028074:GCRCsession2 radiamariejohnson$ less test.txt
web028074:GCRCsession2 radiamariejohnson$ more test.txt
This is a test text file.
Thanks!
web028074:GCRCsession2 radiamariejohnson$ man more
web028074:GCRCsession2 radiamariejohnson$
```

Renaming, creating and deleting files & directories

- mv => rename or move file to a new location
- cp => copying files
- mkdir => create a new directory
- rm => remove files & directories (delete forever – no trash bin)
- rmdir => removes empty directories only

In the terminal

```
web028074:GCRCsession2 radiamariejohnson$ mkdir copy_of_files
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ cp test.txt test_dup.txt
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/  test_dup.txt
web028074:GCRCsession2 radiamariejohnson$ mv test_dup.txt copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ ls copy_of_files/
test_dup.txt
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ pwd
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ mkdir empty_folder
web028074:GCRCsession2 radiamariejohnson$ rmdir copy_of_files/
rmdir: copy_of_files/: Directory not empty
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/  empty_folder/
```

In the terminal

```
web028074:GCRCsession2 radiamariejohnson$ rmdir empty_folder/
web028074:GCRCsession2 radiamariejohnson$ ls
test.txt        copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ mv test.txt example1.txt
web028074:GCRCsession2 radiamariejohnson$ ls
example1.txt    copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ cp example1.txt copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ ls copy_of_files/
test_dup.txt    example1.txt
web028074:GCRCsession2 radiamariejohnson$ ls
example1.txt    copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ rm example1.txt
remove example1.txt? y
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/
web028074:GCRCsession2 radiamariejohnson$ cp copy_of_files/example1.txt .
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt
web028074:GCRCsession2 radiamariejohnson$
```

Command options

- Short form (UNIX/LINUX)
  - ls -a
  - ls -l
  - rm -r
- Long form (LINUX only)
  - ls --all
  - ls --format=long
  - rm --recursive

In the terminal

```
web028074:GCRCsession2 radiamariejohnson$ ls -a
../             copy_of_files/  example1.txt    ./
web028074:GCRCsession2 radiamariejohnson$ ls -l
total 8
drwxr-xr-x  4 radiamariejohnson  staff  136 24 Oct 13:47 copy_of_files/
-rw-r--r--  1 radiamariejohnson  staff   34 24 Oct 13:48 example1.txt
web028074:GCRCsession2 radiamariejohnson$ ls -la
total 8
drwxr-xr-x+ 118 radiamariejohnson  staff  4012 24 Oct 13:35 ../
drwxr-xr-x    4 radiamariejohnson  staff   136 24 Oct 13:47 copy_of_files/
-rw-r--r--    1 radiamariejohnson  staff    34 24 Oct 13:48 example1.txt
drwxr-xr-x    4 radiamariejohnson  staff   136 24 Oct 13:48 ./
web028074:GCRCsession2 radiamariejohnson$ cp copy_of_files/ copy_of_files2/
cp: directory copy_of_files2 does not exist
web028074:GCRCsession2 radiamariejohnson$ cp copy_of_files/ copy_of_files2
cp: copy_of_files/ is a directory (not copied).
web028074:GCRCsession2 radiamariejohnson$ cp -r copy_of_files/ copy_of_files2
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt    copy_of_files2/
web028074:GCRCsession2 radiamariejohnson$ rm copy_of_files2/
rm: copy_of_files2/: is a directory
web028074:GCRCsession2 radiamariejohnson$ rm -rf copy_of_files2/
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt
```

Environment variables

- You can create variables -> VARIABLE=value

  For example, let's create a variable to store our GCRCsession2/ (only in current session):

  - FAVDIR="/Users/radiamariejohnson/GCRCsession2"
  - echo $FAVDIR
  - echo "${FAVDIR}"

  Other common environment variables

  - $HOME or "${HOME}"
  - $PATH or "${PATH}"
  - $PWD or "${PWD}"
  - $SCRATCH or "${SCRATCH}"

In the terminal

```
web028074:GCRCsession2 radiamariejohnson$ ls -a
../             copy_of_files/  example1.txt    ./
web028074:GCRCsession2 radiamariejohnson$ ls -l
total 8
drwxr-xr-x  4 radiamariejohnson  staff  136 24 Oct 13:47 copy_of_files/
-rw-r--r--  1 radiamariejohnson  staff   34 24 Oct 13:48 example1.txt
web028074:GCRCsession2 radiamariejohnson$ ls -la
total 8
drwxr-xr-x+ 118 radiamariejohnson  staff  4012 24 Oct 13:35 ../
drwxr-xr-x    4 radiamariejohnson  staff   136 24 Oct 13:47 copy_of_files/
-rw-r--r--    1 radiamariejohnson  staff    34 24 Oct 13:48 example1.txt
drwxr-xr-x    4 radiamariejohnson  staff   136 24 Oct 13:48 ./
web028074:GCRCsession2 radiamariejohnson$ cp copy_of_files/ copy_of_files2/
cp: directory copy_of_files2 does not exist
web028074:GCRCsession2 radiamariejohnson$ cp copy_of_files/ copy_of_files2
cp: copy_of_files/ is a directory (not copied).
web028074:GCRCsession2 radiamariejohnson$ cp -r copy_of_files/ copy_of_files2
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt    copy_of_files2/
web028074:GCRCsession2 radiamariejohnson$ rm copy_of_files2/
rm: copy_of_files2/: is a directory
web028074:GCRCsession2 radiamariejohnson$ rm -rf copy_of_files2/
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt
```

Creating/running a simple BASH script

- nano helloworld.sh
- chmod a+x helloworld.sh
- ./helloworld.sh
- ~/GCRCsession2/helloworld.sh
- $FAVDIR/helloworld.sh

In the terminal

```
web028074:GCRCsession2 radiamariejohnson$ pwd
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ FAVDIR="/Users/radiamariejohnson/GCRCsession2"
web028074:GCRCsession2 radiamariejohnson$ echo $FAVDIR
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ echo "${FAVDIR}"
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ echo $HOME
/Users/radiamariejohnson
web028074:GCRCsession2 radiamariejohnson$ cd $HOME
web028074:~ radiamariejohnson$ pwd
/Users/radiamariejohnson
web028074:~ radiamariejohnson$ cd $FAVDIR
web028074:GCRCsession2 radiamariejohnson$ pwd
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ $PWD
-bash: /Users/radiamariejohnson/GCRCsession2: is a directory
web028074:GCRCsession2 radiamariejohnson$ FAVDIR=$PWD
web028074:GCRCsession2 radiamariejohnson$ echo $FAVDIR
/Users/radiamariejohnson/GCRCsession2
web028074:GCRCsession2 radiamariejohnson$ FAVDIR=pwd
web028074:GCRCsession2 radiamariejohnson$ echo $FAVDIR
pwd
web028074:GCRCsession2 radiamariejohnson$ FAVDIR=$PWD
web028074:GCRCsession2 radiamariejohnson$ cd $FAVDIR
web028074:GCRCsession2 radiamariejohnson$ ls
copy_of_files/  example1.txt
```

Writing a BASH script to run STAR

- https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

3 Running mapping jobs.

3.1 Basic options.

The basic options to run a mapping job are as follows:

```
--runThreadN NumberOfThreads
--genomeDir /path/to/genomeDir
--readFilesIn /path/to/read1 [/path/to/read2]
```


`--genomeDir` specifies path to the genome directory where genome indices where generated (see Section 2. Generating genome indexes).

`--readFilesIn` name(s) (with path) of the files containing the sequences to be mapped (e.g. RNA-seq FASTQ files). If using Illumina paired-end reads, the *read1* and read2 files have to be supplied. STAR can process both FASTA and FASTQ files. Multi-line (i.e. sequence split in multiple lines) FASTA file are supported. If the read files are compressed, use the `--readFilesCommand` *UncompressionCommand* option, where *UncompressionCommand* is the un-compression command that takes the file name as input parameter, and sends the uncompressed output to stdout. For example, for gzipped files (*.gz) use `--readFilesCommand zcat` OR `--readFilesCommand gunzip -c`. For bzip2-compressed files, use `--readFilesCommand bunzip2 -c`.

Using STAR Options

- --genomeDir
- --runThreadN
- --readFilesIn
- --readFilesCommand
- --outFileNamePrefix
- **--quantMode**

quantmode

7 Counting number of reads per gene.

With `--quantMode GeneCounts` option STAR will count number reads per gene while mapping. A read is counted if it overlaps (1nt or more) one and only one gene. Both ends of the paired-end read are checked for overlaps. The counts coincide with those produced by htseq-count with default parameters. This option requires annotations (GTF or GFF with –sjdbGTFfile option) used at the genome generation step, or at the mapping step. STAR outputs read counts per gene into ReadsPerGene.out.tab file with 4 columns which correspond to different strandedness options:

column 1: gene ID

column 2: counts for unstranded RNA-seq

column 3: counts for the 1st read strand aligned with RNA (htseq-count option -s yes)

column 4: counts for the 2nd read strand aligned with RNA (htseq-count option -s reverse)

Select the output according to the strandedness of your data. Note, that if you have stranded data and choose one of the columns 3 or 4, the other column (4 or 3) will give you the count of antisense reads. With `--quantMode TranscriptomeSAM GeneCounts`, and get both the `Aligned.toTranscriptome.out.bam` and `ReadsPerGene.out.tab` outputs.

C:\Users\Radia\Downloads\star.sh - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

microArrayEXMD.R | minimal_example.sh | star.sh

```bash
#!/bin/bash

# Add the modules needed for the analysis
module add STAR/2.4.2
module add SAMtools

# Set variables
FILENAME=$1
myDIR=$2
mySCRIPTDIR=$3
myGENOMEDIR="/RQexec/johnsonr/Homo_sapiens/UCSC/hg19/Sequence/Chromosomes"

# Start the count variable from 0
count=0

#-----------------------------------------------------------------------------
## Run once after you can comment the line or remove it from your script ##
#-----------------------------------------------------------------------------
# Go to the STAR directory
# cd "${myGENOMEDIR}"

# Build a STAR genome index into the Chromosomes folder
# STAR  --runMode genomeGenerate --genomeDir ./ --genomeFastaFiles hg19.fa --runThreadN 4 --sjdbGTFfile /RQexec/johnsonr/Homo_sapiens/UCSC/hg19/Annotation/Genes/genes.gtf  --sjdbOverhang 10

#-----------------------------------------------------------------------------
## Run from here for all samples
#-----------------------------------------------------------------------------
# Go to the directory with your scripts
cd "${mySCRIPTDIR}"

while read mySAMPLE
        do
                let count++
                echo "$count $mySAMPLE"

                # Go to the STAR directory where the results will be stored
                # mkdir $SCRATCH/STAR # command to create folder if not already done
                cd $SCRATCH/STAR

                # Create the folder to store the STAR results
                mkdir "${mySAMPLE}"_STAR
                cd "${mySAMPLE}"_STAR

                # Align the RNAseq reads to the genome with STAR
                STAR --genomeDir "${myGENOMEDIR}"  --quantMode GeneCounts --runThreadN 4 --readFilesIn "${myDIR}"/"${mySAMPLE}"*1.f*q.gz "${myDIR}"/"${mySAMPLE}"*2.f*q.gz --readFilesCommand zcat --outFileNamePrefix "${mySAMPLE}"

done < $FILENAME

echo -e "\nTotal $count lines read"
```

Writing a BASH script to run HTSeq

	union	intersection _strict	intersection _nonempty
read / gene_A	gene_A	gene_A	gene_A
read / gene_A	gene_A	no_feature	gene_A
read / gene_A / gene_A	gene_A	no_feature	gene_A
read / read / gene_A / gene_A	gene_A	gene_A	gene_A
read / gene_A / gene_B	gene_A	gene_A	gene_A
read / gene_A / gene_B	ambiguous	gene_A	gene_A
read / gene_A / gene_B	ambiguous	ambiguous	ambiguous

http://www-huber.embl.de/users/anders/HTSeq/doc/count.html

Options

-f <format>, **--format**=<format>
Format of the input data. Possible values are sam (for text SAM files) and bam (for

-r <order>, **--order**=<order>
For paired-end data, the alignment have to be sorted either by read name or by
indicate how the input data has been sorted. The default is name.

If name is indicated, htseq-count expects all the alignments for the reads of a given
been seen are kept in a buffer in memory until the mate is found. While, strictly s
the buffer is much less likely to overflow.

-s <yes/no/reverse>, **--stranded**=<yes/no/reverse>
whether the data is from a strand-specific assay (default: yes)

For stranded=no, a read is considered overlapping with a feature regardless of w
same strand as the feature. For paired-end reads, the first read has to be on the

-a <minaqual>, **--a**=<minaqual> ¶
skip all reads with alignment quality lower than the given minimum value (default:

-t <feature type>, **--type**=<feature type>
feature type (3rd column in GFF file) to be used, all features of other type are ign

-i <id attribute>, **--idattr**=<id attribute>
GFF attribute to be used as feature ID. Several GFF lines with the same feature
Seq analysis using an Ensembl GTF file, is gene_id.

-m <mode>, **--mode**=<mode>
Mode to handle reads overlapping more than one feature. Possible values for <m

-o <samout>, **--samout**=<samout>
write out all SAM alignment records into an output SAM file called <samout>, anr

-q, **--quiet**
suppress progress report and warnings

-h, **--help**
Show a usage summary and exit

C:\Users\Radia\Downloads\hstseqUnion.sh - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

microArrayEXMD.R | minimal_example.sh | star.sh | hstseqUnion.sh

```bash
#!/bin/bash

# Add the modules needed for the analysis
module add HTSeq/0.6.1p1
module add SAMtools

# Variables
FILENAME=$1
myGTF="/RQexec/johnsonr/Homo_sapiens/UCSC/hg19/Annotation/Genes/genes.gtf"
count=0

while read mySAMPLE
        do
                let count++
                echo "$count $mySAMPLE"

                # Go to the STAR directory in the PARK_LAB folder
                cd $SCRATCH/STAR/"${mySAMPLE}"_STAR/

                python -m HTSeq.scripts.count -m intersection-nonempty -f sam "${mySAMPLE}"Aligned.out.sam "${myGTF}" >  "${mySAMPLE}".cnts


done < $FILENAME

echo -e "\nTotal $count lines read"
```

Writing a PBS script

C:\Users\Radia\Downloads\minimal_example.sh - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

microArrayEXMD.R | minimal_example.sh

```
#!/bin/bash
#PBS -A abc-123-aa
#PBS -l walltime=30:00:00
#PBS -l nodes=2:ppn=8
#PBS -q queue
#PBS -r n

module load compilers/intel/12.0.4
module load mpi/openmpi/1.4.5_intel

cd $SCRATCH/workdir
mpiexec /path/to/my/mpi_program
```

https://wiki.calculquebec.ca/w/Ex%C3%A9cuter_une_t%C3%A2che/en

Launching jobs on the cluster

- qsub script.pbs
- qstat -u username
- qdel job_id

Other ressources

- Unix less command navigation: http://www.thegeekstuff.com/2010/02/unix-less-command-10-tips-for-effective-navigation/