RNAseq workshop Day1 (organized)
1. How to log onto server
2. download the data
3. FastQC the data

Part_1. log onto server and check the tools
login address: ssh -l ngscls33 aristotle.med.cornell.edu
password: hs!acgt@2001
type of aristotle: ssh scu-node02

find the tools stored in the 2017_rnaseq folder
[ngscls21@scu-node02 ~]$ ls /zenodotus/abc/store/courses/2017_rnaseq/

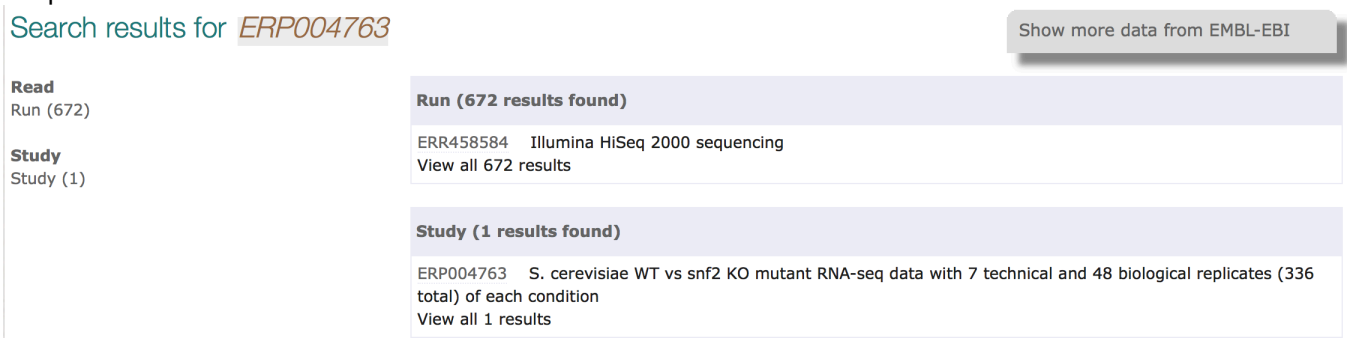
```
[ngscls21@scu-node02 ~]$ ls /zenodotus/abc/store/courses/2017_rnaseq/
aln_tmp  rawReads_yeast_Gierlinski  referenceGenomes  software
```

create a symbolic link for the folder /zenodotus/abc/store/courses/2017_rnaseq and name the directory as 'mat'
[ngscls21@scu-node02 ~]$ ln -s /zenodotus/abc/store/courses/2017_rnaseq/ mat
ln means generates a link; -s means symbolic, not hard link
[ngscls21@scu-node02 ~]$ ls mat/

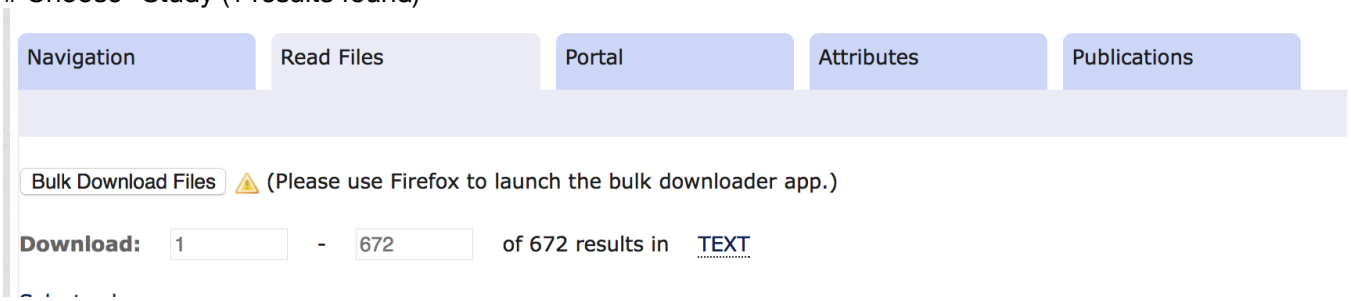
```
[ngscls21@scu-node02 ~]$ ls mat/
aln_tmp  rawReads_yeast_Gierlinski  referenceGenomes  software
```

to log off the server
[ngscls21@scu-node02 ~]$ exit

Part_2. Download the data
2.1
go to ena https://www.ebi.ac.uk/ena
input the reference number for the data to be downloaded “ERP004763”



Choose “Study (1 results found)



2.2
catch the TEXT link of the 672 ftp files, and name it as "samples_at_ENA.txt"

wget -O samples_at_ENA.txt "https://www.ebi.ac.uk/ena/data/warehouse/filereport?accession=PRJEB5348&result=read_run&fields=study_accession,sample_accession,secondary_sample_accession,experiment_accession,run_accession,tax_id,scientific_name,instrument_model,library_layout,fastq_ftp,fastq_galaxy,submitted_ftp,submitted_galaxy,sra_ftp,sra_galaxy,cram_index_ftp,cram_index_galaxy&download=txt"
Don't forget the quotation marks
Use wget to download a webpage; -O writes the downloaded documents to FILE

show the 11th column of the txt file, and only shows the first 10 items. Here it stores the filenames we need to download
[ngscls21@scu-node02 ~]$ cut -f11 samples_at_ENA.txt | head

```
[ngscls33@scu-node02 ~]$ cut -f11 samples_at_ENA.txt | head
fastq_galaxy
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458493/ERR458493.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458494/ERR458494.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458495/ERR458495.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458496/ERR458496.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458497/ERR458497.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458498/ERR458498.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458499/ERR458499.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458500/ERR458500.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458501/ERR458501.fastq.gz
```

2.3
catch the LINK that shows the correlation between ID name and sample name/info, and name it as "ERP004763_sample_mapping.tsv".
wget -O ERP004763_sample_mapping.tsv "http://dx.doi.org/10.6084/m9.figshare.1416210"

show the first 10 items of the code
head ERP004763_sample_mapping.tsv

```
[ngscls33@scu-node02 ~]$ head ERP004763_sample_mapping.tsv
RunAccession    Lane    Sample  BiolRep
ERR458493       1       WT      1
ERR458494       2       WT      1
ERR458495       3       WT      1
ERR458496       4       WT      1
ERR458497       5       WT      1
ERR458498       6       WT      1
ERR458499       7       WT      1
ERR458500       1       SNF2    1
ERR458501       2       SNF2    1
```

2.4 download the actual data for ERR458493 (one of the files)
check the files that need to be downloaded by finding replicate 1 samples:

```
[ngscls33@scu-node02 ~]$ awk '$4 == 1' ERP004763_sample_mapping.tsv | cut -f1
```

ERR458493
ERR458494
ERR458495
ERR458496
ERR458497
ERR458498
ERR458499
ERR458500
ERR458501
ERR458502
ERR458503
ERR458504
ERR458505
ERR458506

method1. Download one by one with the ftp address
```
[ngscls33@scu-node02 ~]$ wget ftp.sra.ebi.ac.uk/vol1/fastq/ERR458/ERR458500/ERR458500.fastq.gz
```

#Awk breaks each line of input passed to it into fields. By default, a field is a string of consecutive characters delimited by whitespace, though there are options for changing this. Awk parses and operates on each separate field. This makes it ideal for handling structured text files -- especially tables -- data organized into consistent chunks, such as rows and columns.
Arbitrarily long lists of parameters cannot be passed to a command in some situations, so `xargs` breaks the list of arguments into sublists small enough to be acceptable.

method2. Download one by one with the sample name
[ngscls33@scu-node02 WT_rep1]$ awk '$5 == "ERR458493"' samples_at_ENA.txt | cut -f11 | xargs wget

method3. Generate a for loop to download selected files
```
[ngscls33@scu-node02 ~]$ for i in `seq 4 6`; do SAMPLE=ERR45850${i}; egrep ${SAMPLE}
samples_at_ENA.txt | cut -f11 | xargs wget; done

# the above code download the files with ERR458504 ~ ERR458506
# there is not gap between SAMPLE, = and ERR45850${i}

#method4. Generate a txt file to download all the files
??????????????????????????????
??????????wget vs asw???????????????????

# How FastQ file look like
[ngscls33@scu-node02 WT_rep1]$ zcat ERR458493.fastq.gz | head
```

```
@ERR458493.1 DHKW5DQ1:219:D0PT7ACXX:1:1101:1724:2080/1
CGCAAGACAAGGCCCAAACGAGAGATTGAGCCCAATCGGCAGTGTAGTGAA
+
B@@FFFFFHHHGHJJJJJJIJJGIGIIIGI9DGGIIIEIGIIFHHGGHJIB
@ERR458493.2 DHKW5DQ1:219:D0PT7ACXX:1:1101:2179:2231/1
ACTAATCATCAACAAAACAATGCAATTCAAGACCATCGTCGCTGCCTTCGC
+
B@=DDFFFHHHHHJJJJIJJJJJJIJJJJJJJJJJJJJJJJJJJJJIJJJJI
@ERR458493.3 DHKW5DQ1:219:D0PT7ACXX:1:1101:2428:2116/1
CTCAAAACGCCTACTTGAAGGCTTCTGGTGCTTTCACCGGTGAAAACTCCG
```

Part_3. Check QC of the data with FastQC

```
# 3.1 Check one FastQC of one sample
# Now the fast.qz files are stored in ~/WT_rep1
# We are going to do FastQC on one file and extract the information into
~/fastqc_results/WT_rep1

# We can use the following code to know how to use fastqc
[ngscls33@scu-node02 WT_rep1]$ ~/mat/software/FastQC/fastqc --help

3.1.1 Method 1
# move to the ~/WT_rep1 directory where the downloaded fastq.gz files are stored before
applying fastqc. Put the corresponding fastQC files into ~/fastqc_results/WT_rep

[ngscls33@scu-node02 WT_rep1]$ ~/mat/software/FastQC/fastqc ERR458493.fastq.gz --extract -o
../fastqc_results/WT_rep1
[ngscls33@scu-node02 ~]$ cd fastqc_results/WT_rep1
[ngscls33@scu-node02 WT_rep1]$ ls
ERR458493_fastqc       ERR458494_fastqc       ERR458495_fastqc
ERR458493_fastqc.html  ERR458494_fastqc.html  ERR458495_fastqc.html
ERR458493_fastqc.zip   ERR458494_fastqc.zip   ERR458495_fastqc.zip
```

The results shows the FastQC results will have 1) fastqc folder; 2) html; 3) zip

```
3.1.2 Method 2
# just extract the information directly. The extracted files will be in the same directory as
the fastqc.gz files. Then move them to the ~fastqc_results_WT_rep1
# move to the ~/WT_rep1 directory where the downloaded fastq.gz files are stored before
applying fastqc
[ngscls33@scu-node02 WT_rep1]$ ~/mat/software/FastQC/fastqc WT_rep1/ERR458495.fastq.gz –extract

3.1.3 Send the html document to email address
# move to the fastqc_results directory
[ngscls33@scu-node02 fastqc_results]$ echo "here are the FastQC results" | mailx -s "FastQC
results" -a WT_rep1/ERR458493_fastqc.html fta2001@med.cornell.edu
# “echo” is the content of the mail; “-s" followed by the email theme name; “-a" followed by
the attachment; ended with the email address


3.2 do FastQC on multiple files on multiple fastq.gz files

3.2.1 Method 1 with *
Do multiQC on multiple files and output them into the ~/fastqc_results/WT_rep1
[ngscls33@scu-node02 ~]$ ~/mat/software/FastQC/fastqc WT_rep1/*fastq.gz --outdir
fastqc_results/WT_rep1/


3.2.2 Method 2 with multiqc
# check how to sue the multiqc software:
[ngscls33@scu-node02 ~]$ ~/mat/software/anaconda2/bin/multiqc --help

# generate multiQC result
[ngscls33@scu-node02 fastqc_results]$ ~/mat/software/anaconda2/bin/multiqc WT_rep1/ --dirs
```

```
[ngscls33@scu-node02 fastqc_results]$ ~/mat/software/anaconda2/bin/multiqc WT_re
p1/ --dirs
[WARNING]         multiqc : MultiQC Version v1.2 now available!
[INFO   ]         multiqc : This is MultiQC v1.1
[INFO   ]         multiqc : Template    : default
[INFO   ]         multiqc : Prepending directory to sample names
[INFO   ]         multiqc : Searching 'WT_rep1/'
Searching 68 files..  [####################################]  100%
[INFO   ]          fastqc : Found 10 reports
[INFO   ]         multiqc : Compressing plot data
[INFO   ]         multiqc : Report      : multiqc_report.html
[INFO   ]         multiqc : Data        : multiqc_data
[INFO   ]         multiqc : MultiQC complete
[ngscls33@scu-node02 fastqc_results]$ ls
multiqc_data  multiqc_report.html  WT_rep1
```

3.2.3 send multiple QC results to email with multiQC

```
# move to the folder where multiqc_report.html is stored (I moved them to WT_rep1 in advance)
[ngscls33@scu-node02 WT_rep1]$ echo "here are the results of multiQC" | mailx -s "MultiQC
results" -a multiqc_report.html fta2001@med.cornell.edu
```