Hands on Exercises for Linux: Beyond the Basics
Whitehead BaRC, October 2022

All commands are in Red.

Please log in to our tak server and create a directory, within your home directory(/home/[USERNAME]/, for the exercises in /nfs/BaRC_training
```
$ mkdir Hot_Topics
$ cd Hot_Topics
```
Copy all the files from /nfs/BaRC_training/Linux_Beyond/data_files/ to your directory
```
$ cp -r /nfs/BaRC_training/Linux_Beyond/data_files/* .
```

########## Exercises ##########

1. Sort the foo.txt file with default or -g or -n, and compare the results.
```
$ sort foo.txt
$ sort -g foo.txt
$ sort -n foo.txt
```

Note: -n is preferred over -g except for scientific notation and a leading '+'

2. Look at the top 10 lines of the sample1.txt
```
$ head sample1.txt
```

The fields of the columns are:
transcript id
gene symbol
length of this transcript
expression value in replicate 1
expression value in replicate 2
expression value in replicate 3
chromosome

3.
a) Print lines 12 to 15:
```
$ sed -n '12,15p' sample1.txt
```
b) Replace NM with RefSeq_NM
```
$ sed 's/NM/RefSeq_NM/' sample1.txt
```

4. Add line count to the sample1.txt
NR: record number
$0: all the columns
```
$ awk '{ print NR"\t"$0 }' sample1.txt
```

5. How many columns are in this file?
-F: field separator

```
$ awk '{ print NF }' sample1.txt | sort -u

# \t:tab
$ awk -F "\t" '{ print NF }' sample1.txt | sort -u
or
$ awk 'BEGIN {FS="\t"} { print NF }' sample1.txt | sort -u


6. How many genes (column 2) are in sample1.txt?
$ cut -f2 sample1.txt | sort -u | wc -l
or
$ awk '{ print $2 }' sample1.txt | sort -u | wc -l


7. Which transcript is the longest?
$ cut -f1,3 sample1.txt | sort -k 2,2nr | head -1
or
$ awk '{ print $1"\t"$3 }' sample1.txt | sort -k2,2nr | head -1


8. Display expression levels for the gene ADH1B.
$ awk '/ADH1B/' sample1.txt
or
$ awk '/ADH1B/ { print $0 }' sample1.txt
or
$ awk '{ if(/ADH1B/) print $0 }'  sample1.txt


9. Display the transcripts with max expression value above 10
$ awk '{ if( $4>10 || $5>10 || $6>10 ) print $0 }' sample1.txt


10. Substitute Chr with chr
# substitute first match on record
$ sed 's/Chr/chr/' sample1.txt
or
# OFS: out field separator
$ awk 'BEGIN {OFS="\t"}{sub(/Chr/,"chr"); print $1, $2, $7 }' sample1.txt
$ awk 'BEGIN {OFS=" "}{sub(/Chr/,"chr"); print $1, $2, $7 }' sample1.txt

# only look for specific field: the 7th field in this case
$ awk 'BEGIN {OFS="\t"} {sub(/Chr/,"chr", $7); print $0 }' sample1.txt


# Replace all matched strings in the file.
# replace C with c
$ sed 's/C/c/g' sample1.txt


11. Calculate the average expression for each transcript (and print as
new last column)
$ awk '{ total= $4 + $5 + $6; avg=total/3; print $0"\t"avg}' sample1.txt
or
```

```
$ awk '{ total=0; for (i=4; i<=6; i++) total=total+$i; avg=total/3; print $0"\t"avg }' sample1.txt
```

12. For each gene, find out the length of its longest transcript.

```
$ sort -k2,2 sample1.txt | groupBy -g 2 -c 3 -o max
```

13. Calculate the average expression for each gene.

```
# Set the number of significant figures with -prec
$ sort -k2,2 sample1.txt | groupBy -g 2 -c 4,5,6 -o mean,mean,mean -prec 10
```

14. Summarize Ensembl IDs by gene symbol and add expression info

```
$ head Ensembl_info.txt
$ sort -k3,3 Ensembl_info.txt | groupBy -g 3 -c 1,2 -o distinct,collapse > Ensembl_info.by_gene.txt
# Join Ensembl_info.by_gene.txt with expression info (in Gene_exp.txt)
$ sort -k1,1 Gene_exp.txt > Gene_exp.sorted.txt
$ join -1 1 -2 1 -t $'\t' Ensembl_info.by_gene.txt Gene_exp.sorted.txt > Ensembl_info.with_exp.txt
```

15. Find out the genes overlapping with peaks from a Chip-seq data (peaks.bed). HumanGenesPlusMinus3kb.bed contains all human genes plus regions 3kb up and down. Get the list of unique genes overlapping the peaks.

```
$ intersectBed -a HumanGenesPlusMinus3kb.bed  -b peaks.bed | cut -f4 |sort -u > overlapped_genes.txt
```

16. Try running a shell script that converts SAM files to BAM, sorts them, and then indexes them.

```
$ cd shell script

# 1 - Start by running the script by itself with no arguments
$ ./SAM_to_BAM_sort_index.sh

# 2 - Run the script, followed by the names of the SAM files you'd like to process
$ ./SAM_to_BAM_sort_index.sh *.sam

# 3 - Open the script in a text editor and change the print (printf) statements to something you'd prefer
# Run the above command again and check that it works as expected.
```