Universitat Autonoma de Barcelona, Module 01 Fall 2016
MSc in Bioinformatics

Linux Lab Exercises

Edgar G Nogales, 39928211-G

10/28/2016

Download data file example.bed from the data Dropbox folder to your Home/-downloads folder. Please provide full scripts for the answers

```
edgano@edgano-VirtualBox ~/Downloads $ ls -l
total 3468
-rw-r--r-- 1 edgano edgano 3543943 Oct 20 19:11 example.bed
```

1 Problem Q1

Take a look at the last 10 lines of the file. Which command are you going to use?
Solution:

```
tail -10 example.bed
```

```
edgano@edgano ~/Downloads $ tail -10 example.bed
chr2    13554893        13556977        AT2G31880.1     0       +       13554919        13556845        0       1       2084,   0,
chr3    20251840        20254199        AT3G54710.1     0       +       20251884        20254031        0       7       190,120,232,6
27,1873,2109,
chr1    471989  473160  AT1G02360.1     0       -       472137  473116  0       2       493,518,        678,0,
chr3    21701533        21703114        AT3G58660.1     0       +       21701573        21702914        0       1       1581,   0,
chr3    18709747        18710760        AT3G50410.1     0       +       18709871        18710633        0       1       1013,   0,
chr5    25645262        25646728        AT5G64080.2     0       -       25645474        25646638        0       3       457,27,355,
chr5    25645262        25646735        AT5G64080.1     0       -       25645474        25646638        0       3       464,27,367,
chr1    3055230 3056974 AT1G09470.1     0       -       3055390 3056931 0       5       214,108,462,102,328,    1530,1321,593,405,0,
chr1    28882039        28884545        AT1G76900.1     0       +       28882740        28884377        0       5       50,383,88,133
chr1    28881813        28884566        AT1G76900.2     0       +       28882740        28884377        0       5       187,383,88,13
```

Modify the command to show just the last line of the file
Solution:

```
tail -1 example.bed
```

```
edgano@edgano-VirtualBox ~/Downloads $ tail -1 example.bed
chr1    28881813        28884566        AT1G76900.2     0
```

2 Problem Q2

Extract all lines that start with "chr5" from the file and store them in a new file "chr5.bed"
Solution:

```
grep "chr5" example.bed >> chr5.bed
```

```
edgano@edgano-VirtualBox ~/Downloads $
edgano@edgano-VirtualBox ~/Downloads $ grep "chr5" example.bed  >> chr5.bed
```

First we find the chr5 string and then we send the line to chr5.bed

3 Problem Q3

Use the original example.bed file to find the mRNA entries which are lying on the - strand of the chromosome chr5
Solution:

```
awk '{if ($1 == "chr5" && $6 =="-")
        print $1"\t"$2"\t"$3"\t"$4"\t"$5"\t"$6}'
    example.bed
```

```
chr5    22612636        22613311        0       -
chr5    3015246 3018262 0       -
chr5    17589393        17591535        0       -
chr5    16516633        16522392        0       -
chr5    19908625        19909399        0       -
chr5    1182717 1184765 0       -
chr5    4726732 4728746 0       -
chr5    4726640 4728746 0       -
chr5    24250090        24251595        0       -
chr5    24250090        24251613        0       -
chr5    20270191        20272548        0       -
chr5    4543100 4545256 0       -
chr5    145635  147200  0       -
chr5    17346140        17346764        0       -
chr5    16459583        16463272        0       -
chr5    6300534 6301422 0       -
chr5    2174677 2176106 0       -
chr5    7579262 7588249 0       -
chr5    7579543 7588246 0       -
chr5    7579543 7588246 0       -
chr5    8276784 8277402 0       -
chr5    23171530        23175364        0       -
chr5    23126921        23127654        0       -
chr5    15479176        15483250        0       -
```

4 Problem Q4

Now count the number of entries of Q3 and compare with the total number of entries in example.bed
Solution:

```
awk '{if ($1 == "chr5" && $6=="-")
        print $1"\t"$1"\t"$2"\t"$3"\t"$5"\t"$6}'
        example.bed | wc -l
```

```
edgano@edgano ~/Downloads $ awk '{if ($1 == "chr5" && $6=="-") pri
3842
```

```
wc -l < example.bed
```

```
edgano@edgano-VirtualBox ~/Downloads $ wc -l < example.bed
33410
```

5 Problem Q5

Considering that the exon sizes of each entry in file example.bed are located in field number 11, get the first 10 record exon sizes
Solution:

```
cut -f11 example.bed | tr ',' '\n' | grep -v "^$" |
        sort -n -u | tail -10
```

```
edgano@edgano-VirtualBox ~/Downloads $ cut -f11 example.bed | tr ',' '\n'|grep -v "^$"|sort -n -u| tail -10
5361
5367
5538
5601
5616
5966
6041
6532
6885
7713
```

What we have done here is:
First of all we work only with the column 11, then we replace (translate) the comma for the \n to have each value in different lines.
With the grep we delete the empty lines, to be able to sort and split the string. Once we have the values for themselves, we sort as numbers and then we ignore the duplicate values, then we have the last 10 numbers (highers) and we print the result with tail command.
It's asked to return the FIRST 10 exons. But we don't know if they have to be the 10 higher or the 10 lower. You can switch "tail" and "head" if we use the -r (reverse) flag on the sort command.

6 Problem Q5 Bis

How would you remove the last comma?
Solution:

```
tr ',' '\n'
```

Using the tr command you can delete **ALL** the commas

7 Problem Q6

How would get the smallest exon size from the records? The result should provide a number for each line of the input
Solution:

```
cut -f11 example.bed | tr ',' ' ' | grep -v "^$" |
awk '{min=$1; for (i=1; i<=NF; i++) if ($i<min)min=$i;
        print "min of line ",NR": ",min}'
```

```
min of line 33389: 77
min of line 33390: 119
min of line 33391: 180
min of line 33392: 309
min of line 33393: 249
min of line 33394: 95
min of line 33395: 95
min of line 33396: 52
min of line 33397: 962
min of line 33398: 1014
min of line 33399: 103
min of line 33400: 68
min of line 33401: 2084
```

We have continued with the functionality used the last exercises, but in this case we have introduced a AWK sentence to:
Declare a MIN value and made a for loop for each value of the current line (NF) and print the minimum value and the number of line (NR)

8 Problem Q7

How would you now sort the records so that the first number shown is the smallest exon size? Again, the answer must provide a sorted list of numbers for each line of the input
Solution:

```
cut -f11 example.bed | tr ',' ' ' | grep -v "^$" | awk'
{
for (i=1; i<=NF; i++){
        arr[i]=$1;        #save the element in a list
```

```
}
asort(arr);      #sort the list
printf "line "NR" : ";
for (i=1; i<=NF; i++){
        printf arr[i]" "        #print elements
}
print ""
}' | sort -n -k4          #sort by 1st position of the line
```

```
line 14973 : 555 571
line 14974 : 562 576
line 22807 : 563 563
line 12578 : 565 609
line 17530 : 566
line 4989 : 567
line 8818 : 569 635
line 3230 : 574
line 18978 : 575 595
line 22808 : 575 598
line 14975 : 576
line 3233 : 582 634
line 22809 : 598
line 11294 : 602
line 9385 : 602
line 18379 : 607
line 7735 : 621
line 8820 : 635
line 30756 : 656 693
line 25538 : 717
line 16373 : 761 763 895
line 16374 : 763
line 9888 : 791
line 9889 : 806
line 6400 : 853 880
line 16382 : 947
edgano@edgano-VirtualBox ~/Downloads $
```

9 Problem Q8

Get the 10 largest exons of chr1 stored in example.bed
Solution:

```
awk '{if ($1=="chr1") print $11}' example.bed
| tr ',' '\n' | grep -v "^$"| sort -n -u |tail -10
```

```
edgano@edgano-VirtualBox ~/Downloads $ awk '{if ($1=="chr1") print $11}' example.bed| tr ',' '\n'|grep -v "^$"|sort -n -u| tail -10
3762
3875
3882
3897
4075
4154
4755
5239
5616
7713
```

In this case we have used AWK to filter only the exons from chr1

10 Problem Q9

Now modify Q8 script to receive as a parameter the number of exons to search for
Solution:

```
#!/bin/bash
awk '{if ($1=="chr1") print $11}' example.bed
| tr ',' '\n' | grep -v "^$"| sort -n -u |tail -$1
```

```
edgano@edgano-VirtualBox ~/Downloads $ ./script.sh 10
3762
3875
3882
3897
4075
4154
4755
5239
5616
7713
```

In that case, we just copied the command to the script and we have used $1 (first parameter received) as value for TAIL command

11 Problem Q10

Uncompress a compressed fastq file, example.fastq.gz, take the first 400 lines, and convert from FASTQ to FASTA format
Solution:

```
#!/bin/bash
gunzip $1
head -400 ${1:0:-3}| awk 'NR%4 ==1{print ">" substr($0,2)}
        NR%4 ==2 {print}' > ${1:0:-9}.fasta
```

```
190  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
191  >SRR000001.96 3060N:7:1:998:361 length=36
192  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
193  >SRR000001.97 3060N:7:1:329:1098 length=36
194  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
195  >SRR000001.98 3060N:7:1:378:1025 length=36
196  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
197  >SRR000001.99 3060N:7:1:1162:61 length=36
198  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
199  >SRR000001.100 3060N:7:1:1033:147 length=36
200  TTCTGTCTAGTTAATTTGGGAAGATATTTCCTTTTT
```

In this exercise we will receive a .gz file by console. We will uncompress it and we will work with the first 400 lines of the file name (without the extension .gz). The we pipe the result to the AWK command and we save just the lines we need (we keep the size of the sequence because it's useful in some case). Finally we save the results in a file with ".fasta" extension. Our result is the .fastq file, and the .fasta archive with 200 lines (400 /2)

12 Problem Q11

How would you modify Q10 script to be able to accept any number of fastQ compressed files in a folder? Please provide the whole script
Solution:

```
#!/bin/bash
ext='.fasta'
for filename in 'ls *.fastq.gz'; do
    for fil in ${filename:2}; do
        filQ="${fil%.*}"
        filA="${filQ%.*}"
        filA=$fil_$ext

        gunzip $fil
        head -400 $filQ |awk 'NR%4 ==1{print ">"
        substr($0,2)} NR%4 ==2 {print}' > $filA
        done
done
```

```
#!/bin/bash
ext='.fasta'
for filename in 'ls *.fastq.gz'; do
    echo $filename
    for fil in ${filename:2}; do # :2 to start at the second element - 1st is "ls"
        filQ="${fil%.*}"    #remove .GZ extension -> filQ will be: name.fastq
        fil_="${filQ%.*}"   #remove .fastq extencion -> fil_ will be the "naked" name : name
        filA=$fil_$ext      #concatenate variable: name+extension(.fasta)

        gunzip $fil         #extract
        head -400 $filQ | awk 'NR%4 == 1 {print ">" substr($0, 2)}NR%4 == 2 {print}' $filQ > $filA
    done
done
```

In this case we have used % and * to get the file names. We can do the same of Q10

13 Problem Q12

Open variants.bed file and count variants for exons of chomosome 22, sort exons by number of variants, give the top 10 exon by number of variant.
Solution:

```
cut -f4 variants.bed|sort|uniq -c| sort -nr| head -10
```

```
edgano@edgano-VirtualBox ~/Downloads $ cut -f4 variants.bed | sort | uniq -c | sort -nr | head -10
     40 uc010gqp.3_cds_0_0_chr22_15690078_f
     31 uc003bhh.4_cds_0_0_chr22_46256561_r
     29 uc062bek.1_cds_0_0_chr22_15690246_f
     25 uc062cbs.1_cds_1_0_chr22_22376183_f
     25 uc011agd.3_cds_0_0_chr22_15528159_f
     19 uc062bej.1_cds_1_0_chr22_15690426_f
     18 uc062cbp.1_cds_1_0_chr22_22353111_f
     15 uc062bej.1_cds_0_0_chr22_15690078_f
     15 uc003alp.5_cds_5_0_chr22_31712083_r
     14 uc003atr.4_cds_6_0_chr22_37723185_f
```

First we will work just with column 4 where we have the names of the exons, we count it using uniq c, finally we sort as numbers in a reverse way. We just print the first 10 values.

14 Problem Q13

Modify your previous script to receive a number as a parameter N and then show the top N exons
Solution:

```
cut -f4 variants.bed | sort | uniq -c | sort -nr | head -$1
```

```
40 uc010gqp.3_cds_0_0_chr22_15690078_f
31 uc003bhh.4_cds_0_0_chr22_46256561_r
29 uc062bek.1_cds_0_0_chr22_15690246_f
25 uc062cbs.1_cds_1_0_chr22_22376183_f
25 uc011agd.3_cds_0_0_chr22_15528159_f
19 uc062bej.1_cds_1_0_chr22_15690426_f
18 uc062cbp.1_cds_1_0_chr22_22353111_f
15 uc062bej.1_cds_0_0_chr22_15690078_f
15 uc003alp.5_cds_5_0_chr22_31712083_r
14 uc003atr.4_cds_6_0_chr22_37723185_f
13 uc032qhw.1_cds_0_0_chr22_22514002_r
11 uc003bix.3_cds_1_0_chr22_49883663_f
11 uc003bck.3_cds_3_0_chr22_42209651_r
11 uc003bcj.3_cds_4_0_chr22_42209651_r
10 uc062ccv.1_cds_1_0_chr22_22704524_f
10 uc003bhw.1_cds_34_0_chr22_46533627_r
10 uc002zuq.5_cds_0_0_chr22_21383859_f
10 uc002zsd.5_cds_0_0_chr22_18606515_r
 9 uc062bxq.1_cds_0_0_chr22_21126248_r
 9 uc062bfr.1_cds_11_0_chr22_17108307_f
```

15 Problem Q14

Convert a gff file to bed format using awk. Receive input file as an argument of the script and provide an error message when the file is not provided by the user
Solution:

```
#!/bin/bash
if [ -z "$1" ]
        then
        echo "File is not provided"
else
        awk 'BEGIN {print "chr\t chr_start\t chr_end\t
        name\t score\t strand\t"} NR>= 2{split($9,a,"=");
        split(a[2],b,";"); print $1"\t"$4"\t"$5"\t"b[1]
        "\t"$6"\t"$7}' $1 > nuevo.bed
fi
```

chr	chr_start	chr_end	name	score	strand
Chr1	2903	10817	LOC_Os01g01010	.	+
Chr1	2903	10817	LOC_Os01g01010.1	.	+
Chr1	2903	3268	LOC_Os01g01010.1:exon_1	.	+
Chr1	3354	3616	LOC_Os01g01010.1:exon_2	.	+
Chr1	4357	4455	LOC_Os01g01010.1:exon_3	.	+
Chr1	5457	5560	LOC_Os01g01010.1:exon_4	.	+
Chr1	7136	7944	LOC_Os01g01010.1:exon_5	.	+
Chr1	8028	8150	LOC_Os01g01010.1:exon_6	.	+
Chr1	8232	8320	LOC_Os01g01010.1:exon_7	.	+
Chr1	8408	8608	LOC_Os01g01010.1:exon_8	.	+
Chr1	9210	9617	LOC_Os01g01010.1:exon_9	.	+
Chr1	10104	10187	LOC_Os01g01010.1:exon_10	.	+
Chr1	10274	10430	LOC_Os01g01010.1:exon_11	.	+
Chr1	10504	10817	LOC_Os01g01010.1:exon_12	.	+
Chr1	2903	3268	LOC_Os01g01010.1:utr_1	.	+
Chr1	3354	3448	LOC_Os01g01010.1:utr_2	.	+
Chr1	3449	3616	LOC_Os01g01010.1:cds_1	.	+
Chr1	4357	4455	LOC_Os01g01010.1:cds_2	.	+
Chr1	5457	5560	LOC_Os01g01010.1:cds_3	.	+
Chr1	7136	7944	LOC_Os01g01010.1:cds_4	.	+
Chr1	8028	8150	LOC_Os01g01010.1:cds_5	.	+
Chr1	8232	8320	LOC_Os01g01010.1:cds_6	.	+
Chr1	8408	8608	LOC_Os01g01010.1:cds_7	.	+

In this case we will work with $1 and we will not manipulate it. The final file will be "nuevo.bed".
First of all we will print the header using the BEGIN section of AWK. Then we will print the values for the bed format, but we need to manipulate the column 9 to "clean" the name. For this we use 2 splits to keep whatever we have between the "=" and the ";".

16 Problem Q15

Using the Plant genome data set, find the shortest gene available at peach.genes. Then extract the corresponding sequence from scaffoldX in a gene.fasta file. For example, if the gene positions were (32,114), the script should print my gene output:
¿scaffold1 AGACTAGGAGTTTCCAATTTGGGTGGAAAAAGTATGAGAATTCA-CATTAGCCAT TGAGGATTCGAATGGACACAAAATAACTGGCAAAACAT-GAATCAATAAAACTTAACCGATTCAACAA TACATGTCTGTATCGTGAGC-TAGGGTTGGCTACATAAATG

¿my_gene TATGAGAATTCACATTAGCCATTGAGGATTCGAATGGACACAAAATAACTG-GCAA AACATGAATCAATAAAACTTAACCGAT
Solution:

```
#!/bin/bash
vari=$(cut -f3-5 peach.genes |awk
'{print $1"\t"$2"\t"$3"\t"$3-$2}' | sort -k4 -n |head -1)

scaffold=$(echo $vari |cut -d"␣" -f1)
schIni=$(echo $vari | cut -d "␣" -f2)
schSize=$(echo $vari |cut -d "␣" -f4)

cat peach.genome | tr -d "\n" | awk -v scaffold=$scaffold
        -v schIni=$schIni -v schSize=$schSize 'BEGIN
    {print ">"scaffold"\t"} s=index($0,scaffold){print
    substr($0, s+10+schIni, schSize)}'
```

```
edgano@edgano-VirtualBox ~/Downloads $ ./test.sh
>scaffold_6     ATCCAACGGTGAGTGACCACAAATCTCTTAGACCCCTGTAGTCCAAGAGATCAGAACTATATAG
ATATACTTGTATAAAATATTGAGAATATTCTTAATCTCGAAATAAAAATAATAAAAGGATATTCTCTAGCAAAGAATATT
TGGTCATGAGAATACTTTTTCTTCCATTGTGAGATAAATCTTATCCATTTTTCTCATTAAAAAGAAGAAGAAAAACTCGG
GCCTTATAAAATGAAATATGAAATTTCATAGTAACAAAATGATTAGAGGAAACTAAAGAGAAAGTTTGCTTGCTTTGAGT
AAGTATGAGGAAACTTGTTGAGGTAATAATTCTAGTTGTAGTGAAAGCAAATATTAAAGCTTCTTTTCTCTCTCTCTCCC
TCCTCAGCTCTCACCC
```