Regular expressions: Text editing and Advanced manipulation

HORT 530

Lab 4

Instructor: Kranthi Varala

Regular expressions in grep

- Use the `-E` option with grep to make sure it processes regular expressions correctly.
- The `-E` option ensures the use of Extended regular expression which allows more complex patterns.
- Regex provided to grep is not enclosed within the // borders. Instead it is enclosed in single quotes ‘’

```
$ grep -E '[0-9][5-9]\.' GSE49418_series_matrix.txt
```

Regular expressions in grep

- Character sets supported by grep:
- [[:alpha:]] ==> Alphabets
- [[:alnum:]] ==>Alphabets and digits
- [[:digit:]] ==> digits
- [[:space:]] ==> space, tab, newline etc.

See full list here:

https://www.gnu.org/software/grep/manual/html_node/Character-Classes-and-Bracket-Expressions.html

Regular expressions in grep

- Special characters supported by grep:
- \b ==> End of a word. Any character that is not [a-z],[A-Z],[0-9] or _
- \s ==> Whitespace, similar to [[:space:]]
- \S ==> Not whitespace

Regular expressions in sed

- Supports the same regular expressions as grep.
- `$ sed 's/^\sMy\b/Our/' <File>` will replace the word My with Our, only when the word My is the first word in the line. The '\b' in the pattern ensures that there is no letter immediately after My.
- Therefore, words like Myself, Mystery, Myth etc. do not match.

tr command

- 'tr' is a command to translate i.e., replace one character with another.

`$ tr 'A' 'G' <File>` will replace all occurrences of A with G in the specified file.

- If tr is given the -d option it simply removes all instances of the character.

`$ tr -d '\t' <File>` will remove all tabs from the specified file.

Today's pairs

	Pair#1	Pair#2	Pair#3	Pair#4	Pair#5	Pair#6
Week4	Meredith, Freddie	Sharlene, Chris	Xiaohui, Mithila	Brenden, Emily	Rachel, Maria	Hui, Scott

Exercises

- We will work with grep, sed, awk and tr to solve these text manipulation problems.

1. Copy the following files from /home/kvarala/Files/ to your working directory.
    1. North_of_Boston.txt
    2. GSE49418_series_matrix.txt
    3. SRR6473489.fastq and
    4. At_Promoters_1KB.fasta
    5. bZIP1_TargetIDs.txt

2. Count the number of lines in each file.

Exercises contd...

3. Find all lines in North_of_Boston.txt that start with the word 'I'

NOTE: The grep command should be invoked with the -E option to allow use of regular expressions.

4. Replace all occurrences of the word ‘My’ with ‘Our’ in the file North_of_Boston.txt and save this output to the file New_Boston.txt
NOTE: Use the sed command to replace ‘My’ with ‘Our’.

5. Retrieve the first four lines of the poem ‘Mending Wall’ that is in the file North_of_Boston.txt **HINT**: grep can return a given number of lines before (-B) or after (-A) the match.

Exercises contd...

6. Create two files: 1. Contains ONLY the comment lines from the GSE49418_series_matrix.txt file and 2. Contains ONLY the non comment lines from GSE49418_series_matrix.txt. HINT: All the comment lines start with !

7. Find all lines from GSE49418_series_matrix.txt where the gene expression in the first data column is >=10 HINT: Use the awk command to check the value of first column ($1) is >=10.

Exercises contd...

At_Promoters_1KB.fasta contains the 1Kb sequence immediately upstream of every gene in the Arabidopsis thaliana genome. The file bZIP1_TargetIDs.txt lists the 470 genes up-regulated by the bZIP1 transcription factor Para et. Al., 2014 (PNAS July 15, 2014 111 (28) 10371-10376;). The bZIP1 TF is known to bind a DNA Motif: G[AC]CACGT Using the grep and sed tools identify how many of the 470 genes listed in bZIP1_TargetIDs.txt contain the motif G[AC]CACGT in their promoter.

Exercises contd...

Here are the steps to achieve this:

1. Use grep to extract the promoter sequences for these 470 genes from the full set of promoters. You can give list a file of IDs to search for using the -f option. Since each promoter sequence is 1000 bases and there are 80 bases per line, each promoter sequence is spread over 1000/80 = 12.5 lines. So, after each gene ID the next 13 lines contain the promoter of that gene. Redirect these lines to a file (Step1.file).

Exercises contd...

2. When searching for multiple IDs from the file grep will insert a line containing only ‘--’ between each match. So, use grep again to remove all lines that do not contain an alphabet from the file generated in step 1 and redirect to a new file (Step2.file).

Exercises contd...

3. Step2.file should contain 470 promoter IDs and their sequences. But, if the motif occurs at the end of the line it will be hard to find. So, we want all of the promoter sequence in one line. We can achieve that as follows:

a. Remove all new line characters from this file using the tr command:

$ tr -d ‘\n’ Step2.file >OneLine.file

b. Reintroduce new line characters after the string ‘1000’ and before the character ‘>’. You will have to use two sed commands and use the s///g syntax to achieve this. For example: $ sed ‘s/A/Z/g’ Example.fasta will replace ALL occurrences of A with Z on EVERY line in Example.fasta. Without the ‘g’ at the end only the first occurrence of A will be replaced on every line.

c. Redirect the output of the second sed command to a new file: Step3.file

Exercises contd...

4. Now use grep -c to find the number of promoters that have the motif G[AC]CACGT in Step3.file.