Introduction to Unix shell - ANSWERS TO EXERCISES

- 2016/17 Part II BBS Bioinformatics
- 16 Jan 2017, 15:00-17:00
- Bioinformatics Training Room, Craik-Marshall Building, Downing Site
- Alexey Morgunov

and

- 2016/17 Part III Systems Biology SEB module
- 8 Feb 2017, 10:00-13:00
- Bioinformatics Training Room, Craik-Marshall Building, Downing Site
- Alexey Morgunov

Contents

Download and unpack the `text files` and the `human reference genome annotation`.

Answers to exercises - Shakespeare

1. Check you have the `shakespeare.txt` file (all works of Shakespeare as text). Process it to output a list of words with frequency counts. Be careful not to count capitalised and non-capitalised words separately, and take care of the apostrophe!

```
tr -sc "A-Za-z\'" '\n' < shakespeare.txt | tr '[:upper:]' '[:lower:]' | sort | uniq -c | sort
-nk1
```

2. Check how many times some country names are mentioned by Shakespeare. What are the most common words to co-occur in the same line with a country name? Filter your output for words shorter than four letters.

```
grep -i "england" shakespeare.txt | wc -l
grep -i "england" shakespeare.txt | tr -sc "A-Za-z\'" '\n' | tr '[:upper:]' '[:lower:]' |
grep -v "england" | grep '.\{4\}' | sort | uniq -c | sort -nk1
```

3. Find the most common bigrams Shakespeare uses. Trigrams?

```
#bigrams
tr -sc "A-Za-z\'" '\n' < shakespeare.txt > sh.words
tail -n +2 sh.words > sh.nextwords
paste sh.words sh.nextwords > sh.bigrams
tr 'A-Z' 'a-z' < sh.bigrams | sort | uniq -c | sort -nk1
#trigrams
tail -n +3 sh.words > sh.thirdwords
paste sh.words sh.nextwords sh.thirdwords > sh.trigrams
cat sh.trigrams | tr "[:upper:]" "[:lower:]" | sort | uniq -c | sort -nk1
```

Answers to exercises - Human genome

1. How many genes are there in the `reference genome`? Don't forget to unpack the file.

```
cut -f3  Homo_sapiens.GRCh38.82.gtf | grep -c gene
cut -f3  Homo_sapiens.GRCh38.82.gtf | sort | uniq -c #alternative
```

2. How many transcripts does your favourite gene have, e.g. ENSG00000001461?

```
grep "ENSG00000001461" Homo_sapiens.GRCh38.82.gtf | cut -f3 | grep "transcript" | wc -l
```

3. How many exons?

```
grep "ENSG00000001461" Homo_sapiens.GRCh38.82.gtf | cut -f3 | grep "exon" | wc -l
```

4. Produce a tab separated file with these columns: transcriptID, exon_number, exon_length.

```
cat Homo_sapiens.GRCh38.82.gtf | tail -n +6 | cut -f9 | cut -d";" -f3 | cut -d\" -f2 > transcriptids.txt
cat Homo_sapiens.GRCh38.82.gtf | tail -n +6 | cut -f9 | cut -d";" -f5 | cut -d\" -f2 > exon_nums.txt
paste -d- <(cut -f5 trial.txt) <(cut -f4 trial.txt) | bc > exon_lengths.txt
paste transcriptids.txt exon_nums.txt exon_lengths.txt > final_output.txt
```

5. Which exon is the longest?

```
grep "ENSG00000001461" Homo_sapiens.GRCh38.82.gtf  > gene.txt
cat gene.txt | cut -f3,4,5 > temp1.txt
cat gene.txt | cut -f9 | cut -f3,5 -d";" > temp2.txt
paste temp1.txt temp2.txt | grep ^exon  > exons.txt
paste -d- <(cut -f3 exons.txt) <(cut -f2 exons.txt) | bc > lengths.txt
paste exons.txt lengths.txt | sort -nk8
# you could do the same with awk in a much simpler way!
awk '$10 ~/ENSG00000001461/ && $3 ~/exon/ {gsub(/"|;/, "", $10); printf("%s\t%d\n", $10, ($5-$4))}' Homo_sapiens.GRCh38.82.gtf | sort -rnk2 | head -1
```

Learn more about `awk` here.

Answers to exercises - PDB

1. Extract the protein sequence from the PDB file `1A8Q.pdb` (ATOM instances).

```
cat 1A8Q.pdb | grep ^ATOM | cut -c18-26 | uniq | cut -d" " -f1 > sequence.txt
```

2. Chech if the sequence from ATOM instances matches the one in SEQRES.

```
cat 1A8Q.pdb | grep ^SEQRES | cut -c20- | tr " " "\n" | sed '/^$/d' > sequence2.txt
diff sequence.txt sequence2.txt
```

Answers to exercises - More sed

1. Write a script that would combine the split lines in `split_lines.txt` using `sed`.

```
sed 'N; s/\n / /; P; D' split_lines.txt
```

Explanation from Wikipedia.

License

Many of the shell scripting exercises are taken from Linux Shell Scripting Tutorial (LSST) v2.0 under a CC-BY-NC-SA license.