UNIX Data Tools

Buffalo Chapter 7

Overview

In Chapter 3 we learned the basic operations within the Unix shell:

- standard out and standard error streams of data
- how to redirect our data streams
- how to efficiently run a series of commands using pipes
- how to manage command processes

Here, we'll learn a number of UNIX tools that will allow us to inspect and manipulate data

Inspecting a data file for the first time: `head`

- Use the `cd` command to navigate into the `chapter-07-unix-data-tools` folder in the Buffalo online resources
- We can inspect a file by using the `cat` command to print its contents to the screen:

```
$ cat Mus_musculus.GRCm38.75_chr1.bed
```

- That's a little unwieldly...perhaps we just want to see the first few lines of a file to see how it's formatted. Let's try:

```
$ head Mus_musculus.GRCm38.75_chr1.bed
```

- If we want to see less or more of a given file, we can specify the number of lines using the `-n` option:

```
$ head -n 3 Mus_musculus.GRCm38.75_chr1.bed
```

Inspecting a data file for the first time: `tail`

- Similar to `head`, you can use the `tail` command to inspect the end of a file:

```
$ tail -n 3 Mus_musculus.GRCm38.75_chr1.bed
```

- `tail` can also be useful for removing the header of a file; this is particularly useful when concatenating files for an analysis:

```
$ tail -n +2 genotypes.txt
```

- And here's a handy trick for inspecting both the head and tail of a file simultaneously:

```
$ (head -n 2; tail -n 2) < Mus_musculus.GRCm38.75_chr1.bed
1     3054233    3054733
1     3054233    3054733
1     195240910    195241007
1     195240910    195241007
```

Additional uses of `head`

- We can also use `head` to inspect the first bit of output of a UNIX pipeline:

```
$ grep 'gene_id "ENSMUSG00000025907"' Mus_musculus.GRCm38.75_chr1.gtf | head -n 1
```

- When including head at the end of a complex UNIX pipeline, the pipeline will only run until it produces the number of lines dictated by `head`
- Why is this important or useful? This dummy pipeline may help:

```
$ grep "some_string" huge_file.txt | program1 | program2 | head -n 5
```

Inspecting files and pipes using `less`

- `less` is what is known as a "terminal pager"; it allows us to view large amounts of text in our terminal
- Whereas with `cat` the contents of our file flash before our eyes, with `less` we can view and scroll through the file's contents
- Let's observe the difference between `cat` and `less` using a file from the Buffalo Chapter 7 materials:

Try:

```
$ cat contaminated.fastq
```

Then try:

```
$ less contaminated.fastq
```

- While viewing the file in `less` try navigating with the space bar and the `b`, `j`, `k`, `g`, and `G` keys. To exit the file, press `q`

Using `less` to highlight text matches and check pipes

- Highlighting text matches can allow us to search for potential problems in data
- For example, imagine we download useful Illumina data from another study and it's not clear from the documentation whether adapter sequence has been trimmed
- We can search for a known 3' adapter sequence using `less`:

```
$ less contaminated.fastq

# then press / and enter AGATCGG
```

- `less` can also be used to check the individual components of a pipe under construction:

```
$ step1 input.txt | less
$ step1 input.txt | step2 | less
$ step1 input.txt | step2 | step3 | less
```

- The commands will only run until a page of your terminal is full, limiting computation time