Introduction to Scientific Computing: A Crash Course

Presented by Travis J Lawrence and Dana L Carper
Quantitative and Systems Biology
University of California, Merced

Worksheet 1.3.1

Exploring genome annotations

Three genome annotation files are available for this assignment: 1.) `Arabidopsis thaliana`, 2.) `Hordeum vulgare` and 3.) `Zea mays`. Unless indicated you only need to answer the question for one genome. Nonetheless, since `gff` formatting is mostly standardized for genome annotations your answers should work on any genome annotation with slight modifications.

1. Change to the directory containing the genome annotation files. Use `ls` to list all the files in the directory. What is the file extension for the genome annotation files? What command would you use to list only the genome annotation files in the directory? You will need to use a wildcard (`*`).

2. Using `less` explore one of the annotation files. Some lines start with `#`. Do these appear to be genome annotation records? Remember in a `flat file` database each line is a record, and each delimited column is a field. However, most `flat file` formats have comment lines that allow the inclusion of additional information not intended to be processed as records. Comment lines are prepended by a character, usually `#` or `!`.

3. Use `head` with the `-n` option to determine the number of comment lines at the beginning of one of the annotation files.

4. To get practice with piping the output of one command to the input of another command using the `|` character redirect the output of your command from `question 3` to `wc`. Does the output of `wc` make sense when taking into account the value you provided the `-n` option for `head`?

5. Using `grep` how would you select lines that contain `#`?

6. How would you select lines with `grep` that begin with `#`? Your answers for `question 5` and `question 6` should be different.

7. How would you select lines that don't start with `#`? Use the `manpage` for `grep` to find an option to do this.

8. Using the output from the command in `question 7` figure out the number of annotations each genome contains. Remember that you can use the output from one command as the input to another command using the `|` character.

9. Using `cut` and the `-f` option incrementally increase the value of the `-f` option to determine

the number of fields each annotation line contains.

10. What information does each field contain? Some fields might not be clear, however, you can quickly lookup the `gff` format online.

11. Which field contains the `seqname` information?

12. Use `cut`, `sort`, and `uniq` to find the values contained in the `seqname` field. Do these values have a biological representation?

13. Develop a command pipeline to determine the number of features annotated for each chromosome in the `seqname` field. Use `grep`, and `wc` to develop this pipeline. You will need to run this pipeline for each chromosome separately.

14. Using `head` or `less` what is the first annotation record? Should we include this `feature type` in our counts for each chromosome? Modify the pipeline from `question 13` to exclude this `feature type` from your counts.

15. The solutions to `question 13 and 14` require that you run the pipeline for each chromosome individually. Modify the pipeline from `question 14` using `cut`, `sort`, `uniq` and an option for `uniq` to reproduce the results from `question 14` but in one command.

16. Based on how `uniq` functions did you need to use the `sort` command in the pipeline for `question 15`?

17. Sort the results from `question 16` in descending order by the number of features.

18. What type of `features` are annotated in the genome? Develop a pipeline that reports each `feature type` once.

19. Develop a pipeline to count the number of times each `feature type` occurs.

20. Sort the results from `question 19` in descending order by the number of times each feature occurs.

21. Develop a pipeline that gives the distribution of `feature type` on each chromosome. You will need to use `grep`, `cut`, `sort`, and `uniq`.

22. Wanting to use the results from `question 21` in a later analysis redirect the output using `>` to a file with a descriptive name.

23. Looking at the results from `question 21` does the ratio of `genes` to `mRNA` have a biological explanation?

Preparing RNA-seq annotation file

Several RNA-seq pipelines require annotation files in a format that differs from `gff`. Even when a pipeline accepts `gff` formatted annotations there are multiple reasons you might want to modify the available annotation. The questions below will lead you through producing a `gene` level annotation in another popular format called `SAF`. `SAF` format starts with a header line with the names of the five required columns separated by tabs. The column names are `GeneID`, `Chr`, `Start`, `End`, `Strand` and additional columns with supplemental annotation information may be added.

24. Using a text editor open a new file and write the header line with column names separated by `tabs` followed by a `hard return` then save and exit your text editor.
25. How would you select only the `gene` `feature type` from the `gff` annotation file? You will likely want to include a `tab` character in your search pattern. Since a `tab` character is easily confused with multiple spaces a special notation is used `\t` to indicate a `tab`.
26. The `GeneID` column in a `SAF` file is a unique identifier for each record. The gene accession tends to be a good choice for this field because it makes down stream analyses easier. Which field in the `gff` annotation contains this information? (Hint: you are looking for something similar to this `ID=gene:ATCG01310`)
27. What delimiter is used to separate the meta data fields in the answer to `question 26`?
28. Develop a command pipeline that extracts the gene accession number for each gene in the `gff` annotation file. You only want the portion after `ID=gene:`. You will need to us `grep`, and `cut` with the `–c` option. Redirect the output (`>`) of this command pipeline to a new file.
29. Next you need to extract the `chromosome`, `start`, `end`, and `strand` information from the `gff` file. Using `cut` you should be able to get this information in one command. Redirect the output (`>`) of this command to a new file.
30. You now have all the information you need for your `SAF` annotation file, but in three separate files. First combine the `GeneID` and the `chromosome`, `start`, `end`, and `strand` information. You can do this using the `paste` command. We did not cover this command in the lecture. Read the `manpage` and experiment with the command to see if you can get the result you want. Once you are getting the results you want redirect (`>`) the output to a new file.
31. Finally, combine the header file you made in `question 24` and file you produced in `question 30`. You can do this by using the `cat` command and the append redirect (`>>`). Before attempting to do this make backup copies of your original files from `questions 24 and 30` using the `cp` command. Again we did not lecture on the `cat` command, but quick experimentation should reveal the function of this command.