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 1.4

Tools used: faslen, fassort, fasuniq, fasfilter, fasgrep, fashead, fassub

1. Familiarize yourself with the FAST definition of FastA format using either the FAST cookbook hosted on github or the FAST publication.

2. The file 1.4.tRNA.fasta contains the reliable tRNA genes for Bacteria, Archaea, Plant and Fungi from tRNADB-CE.

3. **Use faswc to find the number of sequences and nucleotides in your downloaded dataset.**

4. Use **fashead** to look at the first few sequences of the data. It is indicated that upstream and downstream sequences are included along with the tRNA sequence. Is there a way to differentiate the upstream and downstream sequence data from the tRNA sequence data? Use **fassub** to remove the upstream and downstream sequence data from each sequence. **Report the command you used to do this.**

5. **Use faswc to find out how many nucleotides were removed by question 4.**

6. Wanting to look more closely at the alanine tRNAs use **fasgrep, faslen,** and the other Linux Coreutils to develop pipelines to find the number of alanine tRNAs, the range of lengths, and the most common length. **Report your results and the pipeline to retrieve them**.

7. Use **fasgrep**, **faslen**, and **fasfilter** to select the alanine tRNAs that have the most common length found in question 6. **Report the command pipeline used**.

8. You are interested in the unique alanine tRNA sequences, add to your pipeline from question 7 using **fassort**, **fasuniq**, and **faswc** to find the number of unique alanine tRNA sequences. **Report the number found and the command pipeline used to retrieve it**.

Tools used: gbfcut, fascut, fascodon, fascomp, faswc, fassort, fashead, fastail, fasxl

9. The file named 1.4.Penstemon.gb is PopSet Accession: 662169745 in genbank format from NCBI. We will use this dataset and FAST to analyze codon usage, base composition and amino acid composition.

10. Use **less** to quickly look at the dataset. Genbank files contain features that annotate the sequence (e.g. gene, exon, intron). **What features are available in this data?**

11. To look at codon usage, we need to extract the coding sequences from the genbank file and convert it to fastA format so it can be used by the FAST utilities. To do this we can use **gbfcut**. **Using the manpage for gbfcut what command would accomplish this? Is the data in fastA format after running this command?**

12. Pipe the results of the command used in question 11 to **faswc. How many sequences and characters are in the extracted data?**

13. Since this dataset contains different alleles for the same gene we are interested in the aggregated summary of all the data. **Using the manpage for fascodon which option will provide the aggregated summary?**

14. Pipe the results from question 11 to **fascodon** using the option from question 18. **How many sequences had a length that was non-modulo 3? What does this mean? How many stop codons were in the middle or start of the sequence? What does this indicate?**

15. The number of premature stop codons seems suspicious. **What qualifier field indicates the reading frame? Use grep to find out which reading frames occur in this data. Include the grep command and the reading frames in your answer.**

16. Using the manpage to design a **gbfcut** command that would only recover sequences with a certain reading frame. Pipe the results of the command to **faswc** for each reading frame. **Report the command pipeline used and the number of sequences in each reading frame.**

17. Since we are interested in the codon usage of all the sequences we need to modify the sequences to all start in the same reading frame. This can be accomplished by cutting nucleotides from the start of the sequence. Pipe your data from **question 16** to **fascut** for each reading frame to produce sequences that all start in the first reading frame. Redirect your output to a file using (>>) which will append to a file instead of clobbering it. **Report your command for each reading frame.**

18. Now repeat **question 14** with your newly formatted file from **question 17**.

19. Since we only want the aggregated data, design a command line pipeline to only output the lines with the aggregated data with the data sorted by amino acid. **Report the pipeline used.**

20. **Based on this data does there appear to be biased use of codons for each amino acid? Are there any amino acids that have almost equal usage of codons? Do any amino acids have a codon that is not used?**

21. You are now interested in the amino acid composition of the aggregated data. Build a pipeline including **fasxl** and **fascomp** to find the most and least abundant amino acid. **Report the most and least abundant amino acid and the pipeline used to retrieve it.**

22. You are now interested in the base composition of the CDS and introns to determine if there are any base composition biases. First calculate the normalized base composition of the aggregated CDS data. **Provide the command and base frequencies. Do the bases appear at almost equal frequencies?**

23. To further investigate the base composition of the CDS data use the optional "by" parameter of **fascut** to find the base composition of each codon position of the aggregated CDS data. **Report the normalized base composition for each codon position and the command used to retrieve it. Which codon position had the closet to equal frequencies for each nucleotide? Which codon position had the strongest bias for a single nucleotide?**

24. Based on your results from question 28 you want to sort the sequences based on the normalized thymine content of the third codon position to find the sequence with the highest thymine content in the third codon position. To do this use the default behavior of **fascomp** to annotate the sequence description with the normalized base composition and use **fassort** to sort sequences based on a tag in the description. Add **fastail** or **fashead** to the end of your pipeline to only retrieve the sequence with the highest thymine content in the the third codon position. **Report the command pipeline used and the sequence ID of the sequence with the highest thymine content in the third codon position.**

25. Now calculate the base composition of the introns. The person who submitted this data did not annotate the introns. Nevertheless, you should be able to say something about the base composition of the introns by comparing the composition of the whole sequence versus only the CDS. **Provide the command you used to calculate the aggregated base composition of the whole sequence data. Does it appear that the introns have a base composition bias? If so, what bases occur more frequently?**