Project Proposal – PFB 2017

Aim: Develop an interactive, extensible, interactive, and comprehensive script for analysis of RNA-seq data in Python.

Problem Addressed: Modern RNA-seq techniques exist purely as command line tools inaccessible to the average biologist with no computational experience and even the best online tutorials struggle to clearly define good options for the average user. This proposal seeks to ameliorate this problem by wrapping one of these tools, the pseudo-aligner Salmon (Patro et al. 2017) with user-prompts to provide the key data and the option to adjust various Salmon parameters using natural language commands.

Extensions: Although beyond the scope of this course, this platform could easily be expanded to incorporate different RNA-seq alignment protocols (HISAT2, Kallisto, STAR, etc.) for various different uses. The output of this script will be a set of /sample/quant.sf files (a tab-delimited text file format output by Salmon) and is ready for downstream analysis with other tools (featureCounts, htseq-count, StringTie) and ultimately statistical analysis and visualization with common tools (DESeq2, edgeR). Many of these functions could be directly written into the script, allowing for a one-step analysis of this complex and increasingly common data type.

Benefits: This project presents a conceptually interesting and challenging task – to distill the complexities of differential expression analysis into an easy-to-use command line-based script (although future modifications could ready extend Python and wrap these tools in a platform-agnostic GUI) while ensuring immunity to user error. For instance, this tool will automatically download reference transcriptomes and genomes from Ensembl using BioMart based on user provided *Genus species* information when prompted. The tool will ensure that it has all of the necessary data to proceed before initializing commands and will provide continuous user feedback about ongoing processes. In the initial iteration this is likely to be STDOUT from Salmon, but future versions may wrap this output in an easier-to-understand format.

Reach Goals: Time permitting, read quantification with StringTie in Python using the output from Salmon would be the next logical step, enabling the user to directly use the script output for visualization and discovery of differentially expressed genes.

Sample Data: For the purposes of this exercise, we can readily make use of one of the many publically available compiled and annotated bacterial transcriptomes. *Escherichia coli* K12 is a well-characterized strain, but any species of general interest could be used so long as FASTQ files from RNA-seq runs can be found in the published literature and retrieved from GenBank. The small size of bacterial genomes (on the order of megabases) will make the already fast Salmon (aligns one 50,000,000 read FASTQ file to the 1.5 Gb zebrafish genome in less than 10 minutes) magnitudinally faster, giving our group instantaneous feedback about the functionality of our script. It is expected that a single alignment can be completed in less than 2 minutes. We can further speed our analysis through truncation of the files to a fraction of their original size, making our analysis practically instantaneous but flexible enough to accommodate far larger FASTQ files and genomes while also running on a conventional desktop computer with a recent-generation Core i-series processor and >8 GB RAM.