Programming for Bioinformatics
BIOL 8803 B
October 26th, 2015

This week we're going to do a bit of functional genomic analysis.

Commands and such from this week:

`sub` – Create a subroutine

passing scalars to subroutines

`@_` - Subroutine parameters

`references` and the \ operator

passing arrays to subroutines

`dereferencing` – Go back from a reference

`goto` – go somewhere else in the script

`eval` – execute some code on the fly

`use` – use a module

`system` and `` `` `` - make a system call

1.) Write a wrapper for finding orthologous using BLAST

Identifying orthologous genes between different genomes is a very common task and it's one of the, perhaps in fact the absolute, most important things that the comparative genomics group will do in the spring Computational Genomics class. There are a number of different ways of defining orthologous, but this week we are going to do it in a very simple way: reciprocal best BLAST hits. Reciprocal best BLAST hits are pairs of sequences where the best BLAST hit for each sequence is the other sequence.

Consider you have a sequence **A** from species sA whose best hit in species sB is the sequence **B**. **A** will be considered a reciprocal best hit of **B** if the best hit of **B** in species sA is **A.**

Example: when you BLAST the complete set of human coding sequences against the complete set of mouse coding sequences, the best hit for the human gene histone H3.1 is the mouse gene histone H3.1. These two genes would be considered reciprocal best hits and orthologous.   Please do note that this is an overly simplistic way of defining orthologous. If, for example, there had been a gene duplication event in mouse lineage yielding two copies of the histone H3 gene, then simply picking one as the ortholog for the human version would not be a rather bad idea.

Your task for this exercise is to write a script that accomplishes the above example *en masse*. You will need the makeblastdb (or formatdb depending on your BLAST installation. I am ok with your version choice if you can justify if) and the blast program. You can get them by installing the NCBI toolkit located here ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/. Your script should take in as arguments two sets of protein or nucleotide sequences, one from each genome. It should create a database from each using makeblastdb, query each set against the opposite database, and remove the databases files and any other temporary file that you created in the process.   From the results of the queries, it should find those sequence pairs which are reciprocal best hits and give them as output.

This is a small script (50ish lines) so don't kill yourself

2) Write a script for finding overlaps in a bed file in Perl. Again a ~50 lines long script, maybe.
It should find, for each base, its coverage by a given set of coordinates.

Example:
chr1   10   12
chr1   10   15
chr1   10   16

Would yield this:
chr1   10   12   3
chr1   12   15   2
chr1   15   16   1

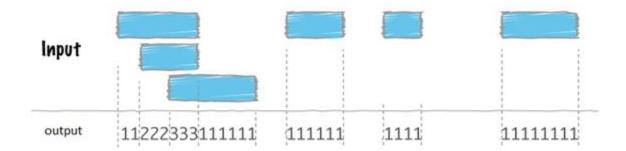Where the last column is the coverage and represents the coverage of the coordinates for that row.

Figure taken from bedtools.readthedocs.org with modifications.

This kind of analysis is important when you are trying to visualize density of specific features. Say you selectively sequenced particular part of the genome (such as in ChIP-seq) and now you want to see how the reads are distributed. You can do this by mapping the reads, taking the mapped coordinates and then computing the overlap in the bed file. The counts will be based on how many times a particular base is covered by the sequenced reads.

- Code:
    - `blast.pl`
    - `overlap.pl`

Additional instructions:

- `blast.pl`
    - blast+ executables are assumed to be under your PATH
    - Only three command line arguments are required for blast.pl, as indicated in `wrapper9.sh`. And they should be taken in the way as shown in `wrapper9.sh`.
    - If you want to implement your `blast.pl` such that it takes additional command line arguments, please make the additional arguments optional.
- `overlap.pl`
    - The input bed file may or may not be sorted. Sort the input file if needed.