Programming for Bioinformatics
BIOL 7200
October 31st, 2016

This week we're going to do a bit of functional genomic analysis. Perl concepts and commands for 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

Exercises:

1) Write a wrapper for finding orthologous genes 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 genes, 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 and vice versa. 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 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` 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 on it

Deliverables

- **Code:** findOrtholog.pl
- **Output:** Your output file with orthologous genes named `findOrtholog.output`
- **A readme.txt** describing how many initial blast hits you found and how many orthologous genes you found

Additional instructions:

- blast+ executables are assumed to be under your PATH
- Only four command line arguments are required for findOrtholog.pl, as indicated in `run_week9.sh`. And they should be taken in the way as shown in `run_week9.sh`.
- You should **NOT** use Perl modules other than 'use strict' and 'use warnings'
- If you want to implement your findOrtholog.pl such that it takes additional command line arguments (so that it can work on more complex tasks than what is required here), please make the additional arguments optional.

2) Write a script for finding overlaps in a BED file in Perl. Again, a ~50 lines long script, maybe.

For each base, it should find 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
```

The last column is the coverage and represents the coverage of the coordinates for that row. This resulting file should be similar to a UCSC bedGraph track format.

Your script **should NOT** generate the results like this:

```
chrX    1    5    0
```

or this

```
chrX    6    7    1

chrX    7    8    1

chrX    8    9    1
```

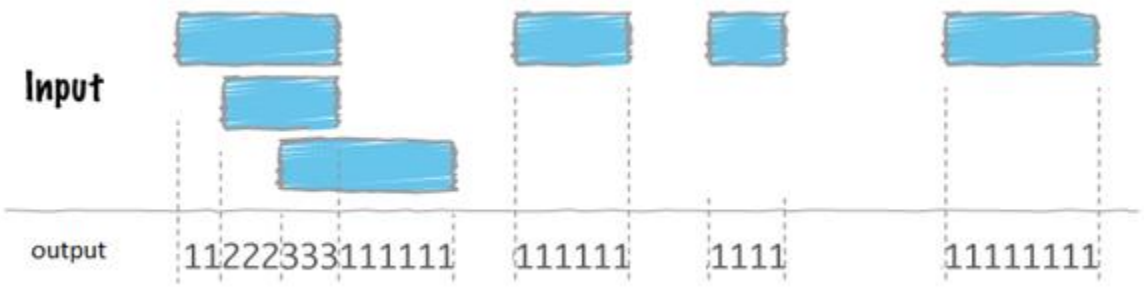

Figure taken from bedtools.readthedocs.org with modifications.

This kind of analysis is important when you are trying to visualize the density of specific features. Say you selectively sequenced particular parts 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.

Deliverables

- **Code:** coverage.pl
- **Output:** Your coverage output for bedExample.bed named `coverage.output`

Additional Instruction:

- The input bed file may or may not be sorted. Sort the input file if needed.
- You should **NOT** use Perl modules other than 'use strict' and 'use warnings'