Series 3

Genomics and bioinformatics - Week 3

October 3, 2011

1 Learning objectives

The final goal of the programming exercises you receive each week is for you to be able to download some interesting data files from UCSC, format them, extract the relevant data from them and apply some statistics that you have seen in this course. At the end of the fourteen weeks, hopefully, you will be able to answer many different types of biological questions by fetching the data needed and using some existing software necessary for the analysis or writing the code necessary for the analysis yourself.

2 Today's exercise

We are going to create a contig from a bunch of reads. The reads have been specially created by the assistants for this problem and will be given to you on the Moodle website. Your goal is to try to align them to generate the largest contig possible. We will consider a simplified ideal case where:

1. The reads don't contain any errors.
2. The reads are all of the same length.
3. The reads are all on the same strand (and necessarily in the same direction).
4. The full chromosome of our imaginary specie is very very small.

2.1 Load the reads

Read the file `reads.fastq` and load all the reads it contains inside a list. Of course, you should probably take a look at `reads.fastq` in your text editor first, this will help you when writing the parsing function.

2.2 Find the overlaps

Write a function that takes two reads as input and returns True if they overlap, false otherwise. The function should also have a `min_overlap` parameter that specifies how many base pairs must match between both reads at a minimum for it to be considered an overlap. If read1 ends with a "C" and read2 starts with a "C", you can't consider that an overlap.

2.3 Generate the edges

Using the the list of reads you just loaded and the function you just wrote you can now generate a list of edges. Indeed, in the graph we are going to create, the reads will act as the vertices and the overlaps as the edges. As you know, an edge exists if a vertex connects to an other vertex. In our case, a link between two reads exists if a read overlaps an other read.

3 Python tricks

1. You don't need to use the indexes of a vector to iterate over its elements.
2. Use `;` to put several statements on the same line. Else it is not needed.
3. dir(object) tells you all the existing methods for this object. If you are using `ipython`, you can type `object.` and press Tab to display the same information.

4 R tricks