Series 3

Genomics and bioinformatics - Week 3

1 Overlap graphs

Here are five reads: TCTATA, GGTCTA, TATCTA, TCTAGC and TATATC. They originate from a single longer contig.

1. Try to reassemble the contig by hand, looking at the overlaps between the reads.
2. Build an Hamiltonian overlap graph with these reads and draw the longest path. From there, reconstruct the contig.
3. Take $l = 4$ and build the corresponding "de Bruijn" graph. Find the Eulerian cycle and reconstruct the contig.
   Note: To apply Theorem 1 (from the lectures) to the "de Bruijn" graph, one has to make sure it is balanced. For this one should start with the graph containing all possible edges associated to the given set of reads and reduce the number of edges at unbalanced vertices without making the graph disconnected, and finally including the artificial edge connecting the start and end vertices. Observe that some edges of the balanced graph are doubled, reflecting the presence of repeats in the contig.

2 Getting data with UCSC

2.1 Visualizing genome data

1. Go to the UCSC Genome Browser and select the C. Elegans genome (Nematode).
2. Visualize the most recent assembly (ce10) of chromosome I.
3. Scroll down to "Mapping and Sequencing Tracks" and load the GC percent track.

2.2 Downloading genome data and look at them

Copy the "sequence" file `chrI.fa` and the "annotation" file `Caenorhabditis_elegans.WBcel235.73.gtf` for C.Elegans from our USB key, or download them from UCSC and ENSEMBL:

`http://hgdownload.cse.ucsc.edu/goldenPath/ce10/chromosomes/`
`ftp://ftp.ensembl.org/pub/release-73/gtf/caenorhabditis_elegans`

Have a look at these two files with any text editor.

3 Manipulating data with Python

1. Load the file `chrI.fa` and extract the sequence.
2. Determine the length of the sequence (see `len`).
3. Calculate the number of As, Gs, Cs and Ts in the sequence.
4. Compute the GC content of the sequence.
5. Plot the GC content using an appropriate window (bin) size (use `matplotlib`).
6. Write the start and end coordinates of each bin and its corresponding GC content to a file as follows: `binStart <tab> binEnd <tab> GC_content`

Note: If you feel confident, this is a good occasion to save time trying the Biopython library:
`from Bio import SeqIO` # then use SeqIO.read()
`from Bio.SeqUtils import GC` # then use GC()
Note: list methods such as `count`, `append`, etc. may be useful.

4 Manipulating data with R

4.1 Exons

The file `Caenorhabditis_elegans.WBcel235.73.gtf` is a tab-delimited file, with the following column headers:
`chromosome source feature start end score strand frame attributes` .

1. Load the file `Caenorhabditis_elegans.WBcel235.73.gtf` in R. It may take a few minutes.
2. Extract the rows corresponding to exons from the `feature` column to a new table.
3. Compute the exon sizes, and attach them to the table in a new column "`exonSize`".
4. Plot the exon size distribution.

4.2 GC content

1. Load the file (table) generated by your python script into R.
2. Recreate the GC content plot in R.

Note: Some useful features for this exercise:

- `which, length, hist`