Series 2

Genomics and bioinformatics - Week 3 - September 27, 2012

1 Getting data with UCSC

1.1 Visualizing genome data

1. Go to the UCSC Genome Browser and select the *Mus musculus* genome.
2. Visualize the most recent assembly (mm10) of mouse chromosome 18.
3. Scroll down to "Mapping and Sequencing Tracks" and load the GC percent track.

1.2 Downloading genome data

Copy the "sequence" file `chr18.fa` and "annotation" files `chr18.gtf`, `chr18_mod.txt` for mouse chr18 from the USB keys provided by us.

Note: Actually, the `.fa` files for mouse can be downloaded from the UCSC Downloads page: `http://hgdownload.cse.ucsc.edu/goldenPath/mm9/chromosomes/`,
and the `.gtf` file for the whole mouse genome can be downloaded from ENSEMBL: `ftp://ftp.ensembl.org/pub/release-64/gtf/mus_musculus/` .

2 Overlap graphs

Here are five "reads": TCTATA, GGTCTA, TATCTA, TCTAGC, TATATC, 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) overlaph 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 an Eulerian cycle, and reconstruct the contig.
   Note: the "de Bruijn" graph must be Eulerian, thus some edges of the graph have to be doubled. Try to understand which ones you must double (look at the overlaps between reads), and think about a criterion to do it automatically.

3 Manipulating data with Python

1. Load the `.fa` file for chr18 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 chromosome.

5. Plot GC content along mouse chr18 using an appropriate window (bin) sizes (use `matplotlib`).

6. Write the start and end coordinates of each bin and it's 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 `.gtf` file is a tab-delimited file, with the following column headers:
`chromosome source feature start end score strand frame attributes` .

1. Load the `.gtf` file for chr18 in R.

2. Extract the rows corresponding to exons from the `feature` column to a new table.

3. Compute exon sizes, and attach them to the table in a new column "`exonSize`".

4. Plot the exon size distribution for chr 18.

4.2 Genes

The modified annotation file `chr18_attributes.txt` is also a tab-delimited file, with the following column headers:
`chromosome source feature start end score strand frame gene_id transcript_id exon_number gene_name gene_biotype transcript_name protein_id` .

1. Load the modified `.txt` annotation file for chr18 in R.

2. Find out the ID and name of the gene containing,
   a) the longest exon, b) most number of exons.

3. List all the intron-less genes in the chromosome.

4.3 GC content

1. Load the file (table) generated by your python script into R.

2. Recreate the GC content plot for chr18 in R.

4.4 Some useful features for this exercise

- `which, length, max, hist`
- Loops and conditions: `for, if, in`
- Conversions: `as.vector, as.numeric, as.data.frame, as.factor, float, int,`...