SamHart_CompBio_HW2

TFCB 2018: Homework 2

Due 12pm, Oct 18, 2018

Bioconductor and sequence motifs

In this homework, we will learn to use Bioconductor functions to identify sequence motifs around well-defined transcription start sites in the human genome.

First, load the packages that we will use.

1. `tidyverse`
2. `rtracklayer`
3. `plyranges`
4. `Biostrings`

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------- tidyverse 1.2.1 -
-
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------- tidyverse_conflicts() -
-
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#source("https://bioconductor.org/biocLite.R")
#biocLite("plyranges")
#biocLite("rtracklayer")
#biocLite("Biostrings")
library(rtracklayer)
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, rename
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'
```

```
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
```

```
## The following object is masked from 'package:purrr':
##
##     reduce
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
## Loading required package: GenomeInfoDb
```

```
library(plyranges)
```

```
##
## Attaching package: 'plyranges'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, n
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(Biostrings)
```

```
## Loading required package: XVector
```

```
##
## Attaching package: 'XVector'
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
##
## Attaching package: 'Biostrings'
```

```
## The following object is masked from 'package:base':
##
##     strsplit
```

```
#In addition to above I used these:
library(GenomicAlignments)
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: DelayedArray
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
## Loading required package: BiocParallel
```

```
##
## Attaching package: 'DelayedArray'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

```
## The following object is masked from 'package:Biostrings':
##
##     type
```

```
## The following object is masked from 'package:purrr':
##
##     simplify
```

```
## The following objects are masked from 'package:base':
##
##     aperm, apply
```

```
## Loading required package: Rsamtools
```

```
##
## Attaching package: 'GenomicAlignments'
```

```
## The following object is masked from 'package:dplyr':
##
##     last
```

```
library(GenomicRanges)
library(GenomicFeatures)
```

```
## Loading required package: AnnotationDbi
```

```
##
## Attaching package: 'AnnotationDbi'
```

```
## The following object is masked from 'package:plyranges':
##
##     select
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

Problem 1

10 points

Read in the annotations of the transcription start sites identified in the FANTOM5 dataset into a `tibble`. This file is available at http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage_peak_phase1and2combined_ann.txt.gz (http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage_peak_phase1and2combined_ann.txt.gz).

Note that the above file has several “comment” lines. Look at the documentation of the function to read `TSV` files to figure out how to ignore these lines while reading the file.

Filter to all transcription start sites of the `GATA1` gene. You need to first figure out which columns contain gene names. You might then find the `str_detect` function from `tidyverse` useful for doing the filtering.

Use `print()` function to display the contents of the final `tibble` containing the transcription start sites of *GATA1*.

```
annotations <- read_tsv("http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage
_peak_phase1and2combined_ann.txt.gz", comment = "#") %>%
   #filter(str_detect(hgnc_id, "4170")) %>% #Looked up hgnc to point me in the right direction
   filter(str_detect(short_description, "GATA1")) %>%
   print()
```

```
## Parsed with column specification:
## cols(
##   `00Annotation` = col_character(),
##   short_description = col_character(),
##   description = col_character(),
##   association_with_transcript = col_character(),
##   entrezgene_id = col_character(),
##   hgnc_id = col_character(),
##   uniprot_id = col_character()
## )
```

```
## # A tibble: 4 x 7
##   `00Annotation` short_descripti~ description association_wit~
##   <chr>          <chr>            <chr>       <chr>
## 1 chrX:48644976~ p2@GATA1         CAGE_peak_~ 0bp_to_ENST0000~
## 2 chrX:48644984~ p1@GATA1         CAGE_peak_~ 2bp_to_NM_00204~
## 3 chrX:48645010~ p3@GATA1         CAGE_peak_~ 28bp_to_NM_0020~
## 4 chrX:48650688~ p5@GATA1         CAGE_peak_~ -102bp_to_ENST0~
## # ... with 3 more variables: entrezgene_id <chr>, hgnc_id <chr>,
## #   uniprot_id <chr>
```

Problem 2

10 points

Read in the coordinates of the transcription start sites identified in the FANTOM5 dataset. This file is available at http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage_peak_phase1and2combined_coord.bed.gz (http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage_peak_phase1and2combined_coord.bed.gz).

Filter for transcription start site peaks that are of just 1nt width and on the positive strand. Note that both `width` and `strand` are default columns of `GRanges` even if they are not displayed.

Stretch the resulting `GRanges` by 10nt on either side using an appropriate function from `plyranges` (https://sa-lee.github.io/plyranges/reference/index.html (https://sa-lee.github.io/plyranges/reference/index.html)).

Use `print()` function to display the contents of the final `GRanges` .

```
TransSS <- import.bed("http://fantom.gsc.riken.jp/5/datafiles/latest/extra/CAGE_peaks/hg19.cage_peak_phase1and2combined_coord.bed.gz") %>%
   #filter (qwidth = 1) %>%
   #qnarrow(start = 1, width = 1) %>%
   filter(width == 1, strand == '+') %>%
   stretch(20) %>%
   print()
```

```
## GRanges object with 1405 ranges and 4 metadata columns:
##          seqnames              ranges strand |                         name
##             <Rle>           <IRanges>  <Rle> |                  <character>
##      [1]     chr1     1286903-1286923      + |     chr1:1286912..1286913,+
##      [2]     chr1     1615039-1615059      + |     chr1:1615048..1615049,+
##      [3]     chr1     1848600-1848620      + |     chr1:1848609..1848610,+
##      [4]     chr1     3105056-3105076      + |     chr1:3105065..3105066,+
##      [5]     chr1     5433409-5433429      + |     chr1:5433418..5433419,+
##      ...      ...                 ...    ... .                         ...
##   [1401]     chrX 152783118-152783138      + | chrX:152783127..152783128,+
##   [1402]     chrX 152927571-152927591      + | chrX:152927580..152927581,+
##   [1403]     chrX 153072579-153072599      + | chrX:153072588..153072589,+
##   [1404]     chrX 155234143-155234163      + | chrX:155234152..155234153,+
##   [1405]     chrY   21589512-21589532      + |   chrY:21589521..21589522,+
##              score     itemRgb     thick
##          <numeric> <character> <IRanges>
##      [1]        12     #FF0000   1286913
##      [2]        13     #FF0000   1615049
##      [3]        17     #FF0000   1848610
##      [4]        11     #FF0000   3105066
##      [5]        11     #FF0000   5433419
##      ...       ...         ...       ...
##   [1401]        11     #FF0000 152783128
##   [1402]      5070     #FF0000 152927581
##   [1403]       888     #FF0000 153072589
##   [1404]        14     #FF0000 155234153
##   [1405]        58     #FF0000  21589522
##   -------
##   seqinfo: 25 sequences from an unspecified genome; no seqlengths
```

Problem 3

10 points

Retrieve the DNA sequence of the 21 nt region around each transcription start site that you obtained in Problem 2. You will find the `getSeq` function in the `Biostrings` package useful.

Use the `BSgenome.Hsapiens.UCSC.hg19` package for the human genome sequence.

Use `print()` function to display the contents of the `DNAStringSet` output of `getSeq` .

```
library(BSgenome.Hsapiens.UCSC.hg19)
```

```
## Loading required package: BSgenome
```

```
TSSseq <- getSeq(BSgenome.Hsapiens.UCSC.hg19, TransSS) %>%
  print()
```

```
##   A DNAStringSet instance of length 1405
##        width seq
##    [1]    21 TATCGGGCCCTGACCGTGCTG
##    [2]    21 GTTTAATCTCCACCTTCGCTC
##    [3]    21 ATGACGGGGGAGTCCTTCAAG
##    [4]    21 TACGTGAGGGAACGCGCTCTC
##    [5]    21 CATTAGCCAGGCAGACACCGG
##    ...   ... ...
## [1401]    21 GAGCGCGCCGCGTCGCCCGCC
## [1402]    21 ATTTAAAACAGTCCTTTTGCG
## [1403]    21 ACATGGACGGAACACGTAACC
## [1404]    21 GACCTGGAGCATCAGTCCTGC
## [1405]    21 ATCTCCCTTTACTGACTCTCT
```

Problem 4

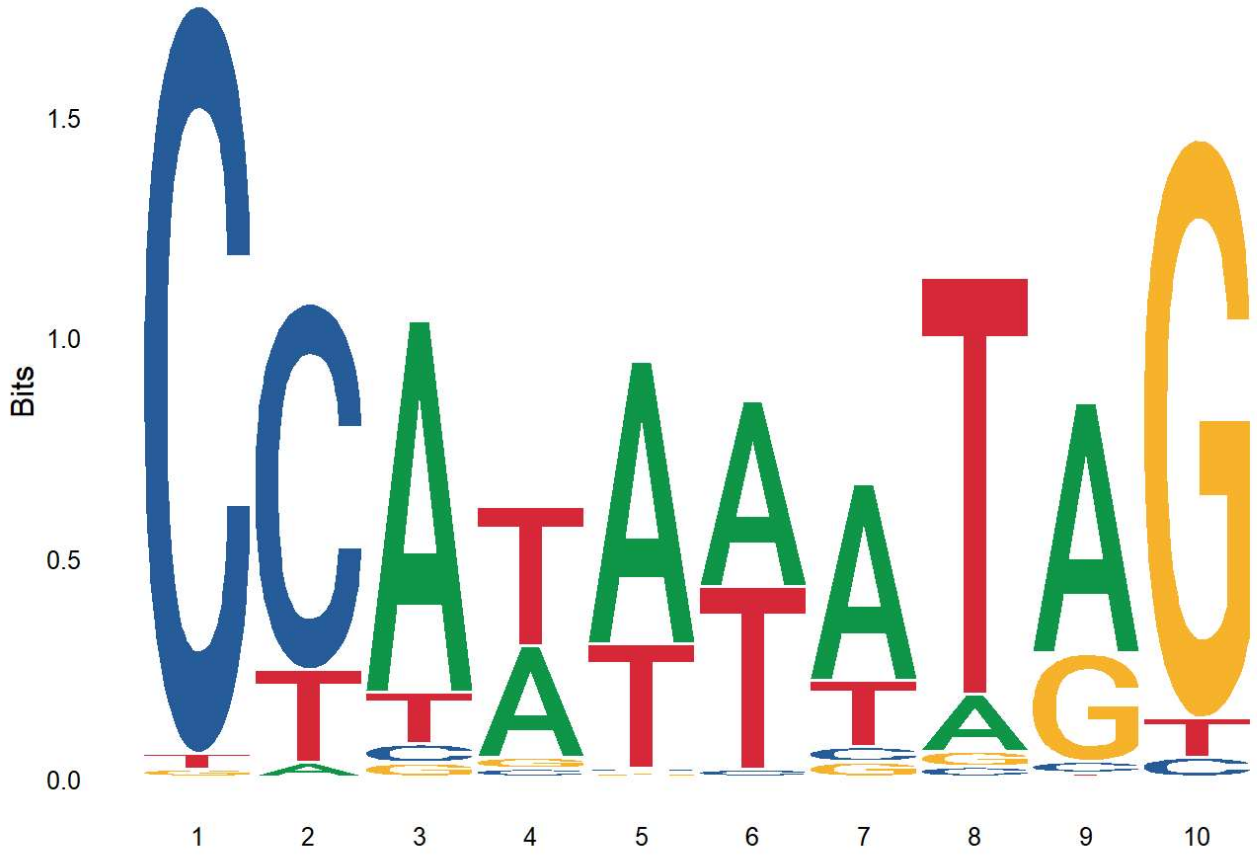
10 points

We will make “logo” plots of the above sequences using a new package called `ggseqlogo`.

Figure out how to install `ggseqlogo` and show that you installed it correctly by making an example sequence logo plot using `ggseqlogo`. You are allowed to use any example from the web, but cite your source as a comment.

```
#install.packages("ggseqlogo")
library(ggseqlogo)

#example data from: https://omarwagih.github.io/ggseqlogo/
data(ggseqlogo_sample)
ggplot() + geom_logo( seqs_dna$MA0001.1 ) + theme_logo()
```



Problem 5

10 points

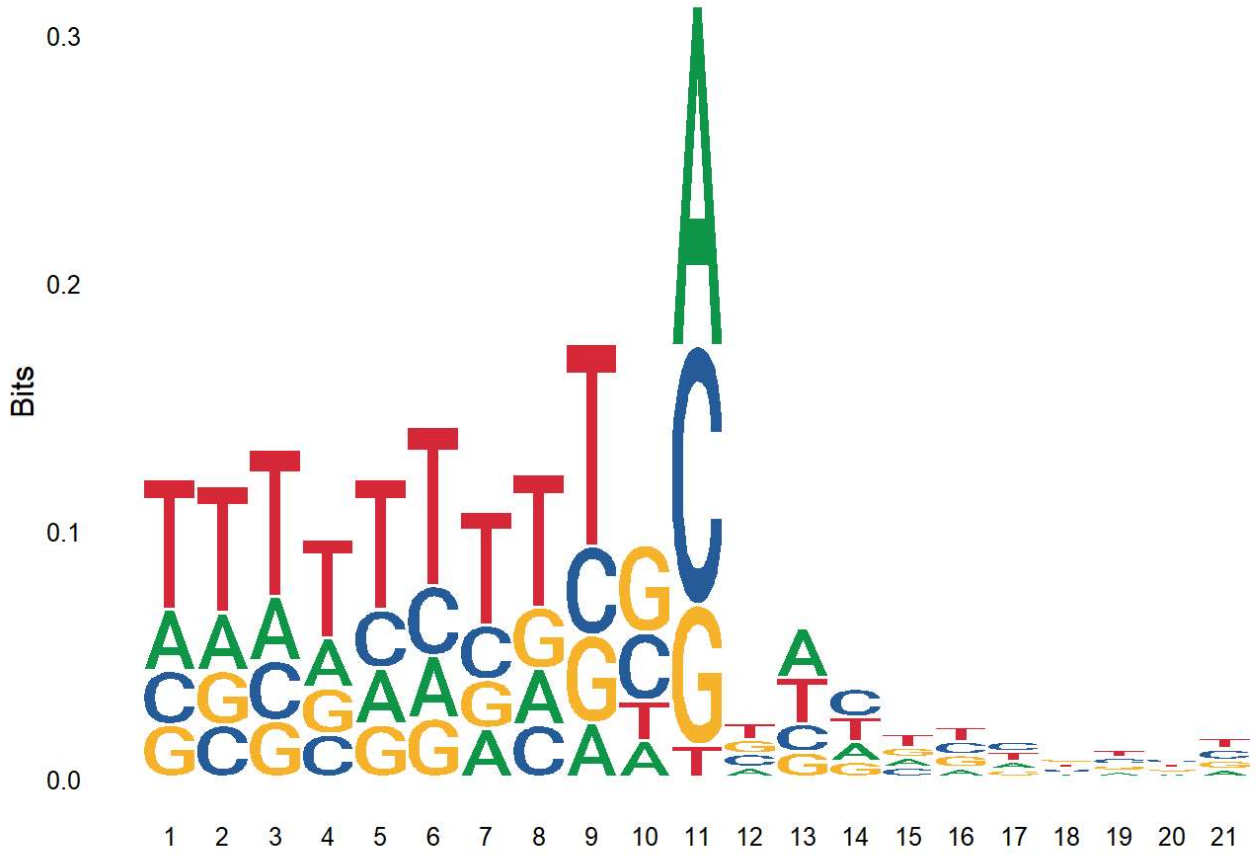
Make and display a sequence logo plot of the sequences around transcription start sites.

First convert the `DNAStringSet` from Problem 3 to R `character` using the function `as.character()`.

Then use this for plotting in `ggseqlogo`.

5 bonus points if you use can adjust the X tick labels to go from -10 to +10!

```
TSSseq2 <- as.character(TSSseq)
ggplot() + geom_logo(TSSseq2) + theme_logo()
```



Git and GitHub

Problem 6

10 points

Make a GitHub account and populate your bio. Here's an example github.com/trvrb/ (https://github.com/trvrb/). Please provide the link to yours.

Sam's Github profile: github.com/sfhart33/ (https://github.com/sfhart33/)

Problem 7

10 points

Make a new project directory using the material from lecture 3 (../../lectures/lecture3) as basis. Call this directory `tfcb-homework2`. Take files in `lecture3/tables/` and move them to a `data/` directory under `tfcb-homework2`. Include a `README.md` under `tfcb-homework2/` that briefly describes this as homework 2 from TFCB and gives your name.

Make this directory a Git repository, commit the readme file as well as the `data/` files and push it to your GitHub account.

Problem 8

10 points

Make an `analysis/` directory under `tfcb-homework2/` and include an R Markdown script to read and operate on these tables following step-by-step instructions from lecture 4 (../../lectures/lecture4/lecture4.pdf). Make the `analysis/` directory your working directory. You'll need to change calls to `read_tsv` in the PDF from:

```
data <- read_tsv("tables/example_dataset_2.tsv")
```

to

```
data <- read_tsv("../data/example_dataset_2.tsv")
```

The file should look something like:

````
---
title: "Homework 2"
output: github_document
---


```r
library(tidyverse)
```


```r
#data <- read_tsv("tables/example_dataset_2.tsv") %>% print()
```
````

Stage and commit this `.Rmd` script and then push changes to your GitHub account.

Problem 9

10 points

Additionally commit and push the resulting knitted version of the Rmd analysis. It will be a file called `.md`.