Markov Models part 1: nothing to hide so far. Solutions

Computational biology

Jacques van Helden

2019-12-10

Contents

```
if (!require("gplots")) {
   install.packages("gplots", dependencies = TRUE)
   library(gplots)
   }
if (!require("RColorBrewer")) {
   install.packages("RColorBrewer", dependencies = TRUE)
   library(RColorBrewer)
   }
```

CpG island length distribution

```
## Load CpG island coordinates
CpG.coord <- read.delim(file = "data/hg38_CpG_islands.bed.gz", header = FALSE)

names(CpG.coord) <- c("chr", "start", "end", "id")
## Note: bed convention : the end marks the first position
## **after** the feature, so the length is simply the difference
CpG.coord$length <- CpG.coord$end - CpG.coord$start
```

```
genome.size <-  3e+9

## Compute statistics about CpG island lengths
CpG.length.stat <- data.frame(
  nb = nrow(CpG.coord),
  min = min(CpG.coord$length),
  Q1 = quantile(CpG.coord$length, probs = 0.25),
  median = median(CpG.coord$length),
  mean = mean(CpG.coord$length),
  Q3 = quantile(CpG.coord$length, probs = 0.75),
  pc95 = quantile(CpG.coord$length, probs = 0.95),
  max = max(CpG.coord$length),
  MB =  sum(CpG.coord$length) / 1e6,
  cov.percent = sum(CpG.coord$length) / genome.size * 100
)



## Print the stats
kable(CpG.length.stat, caption = "Statistics about CpG island lengths", digits = 3, row.names = FALSE)
```

Table 1: Statistics about CpG island lengths

nb	min	Q1	median	mean	Q3	pc95	max	MB	cov.percent
31144	201	325	569	777.05	959	1947	45712	24.2	0.807

The currently annotated CpG islands cover 24.2 Mb, which represent 0.8% of the genome.

```r
par(mfrow = c(2,1))


par(mar = c(4.1, 5.1, 1, 1))

hist(CpG.coord$length,
     breaks = seq(from = 0,
                  to = CpG.length.stat$max + 50,
                  by = 50),
     main = NA, las = 1,
     xlab = "CpG length", ylab = "Number of CpGs")

## Draw the histogram with a limited X axis to see the relevant part of the distribution
hist(CpG.coord$length,
     breaks = seq(from = 0,
                  to = CpG.length.stat$max + 50,
                  by = 50),
     xlim = c(0,3000),
     main = NA, las = 1,
     xlab = "CpG length (truncated X scale)", ylab = "Number of CpGs", col = "gray")
```

```r
par(mar = c(4.1, 5.1, 4.1, 2.1))
par(mfrow = c(1,1))
```

Loading the Markov models

We start by loading the two Markov models previously computed from the RSAT tool create background model.

```r
## Load a transition matrix from the RSAT result
readMarkovModel <- function(file) {
  model <- read.delim(
    file = file,
    row.names = 1)
  names(model) <- toupper(names(model))
  rownames(model) <- toupper(rownames(model))
  prior <- model[nrow(model),1:4] ## Last row contains priors as comments
  names(prior) <- names(model)[1:4]
  transitions <- rbind(model[1:4, 1:4], "B" = prior)
  return(transitions)
}

## Load the CpG transition table from the RSAT result
transitions.CpG <- readMarkovModel(file = "results/hg38/hg38_CpG_transitions_m1.tsv")

## Load the genomic background transition table from the RSAT result
transitions.Bg <- readMarkovModel(file = "results/hg38/hg38_genomic-bg_transitions_m1.tsv")
```

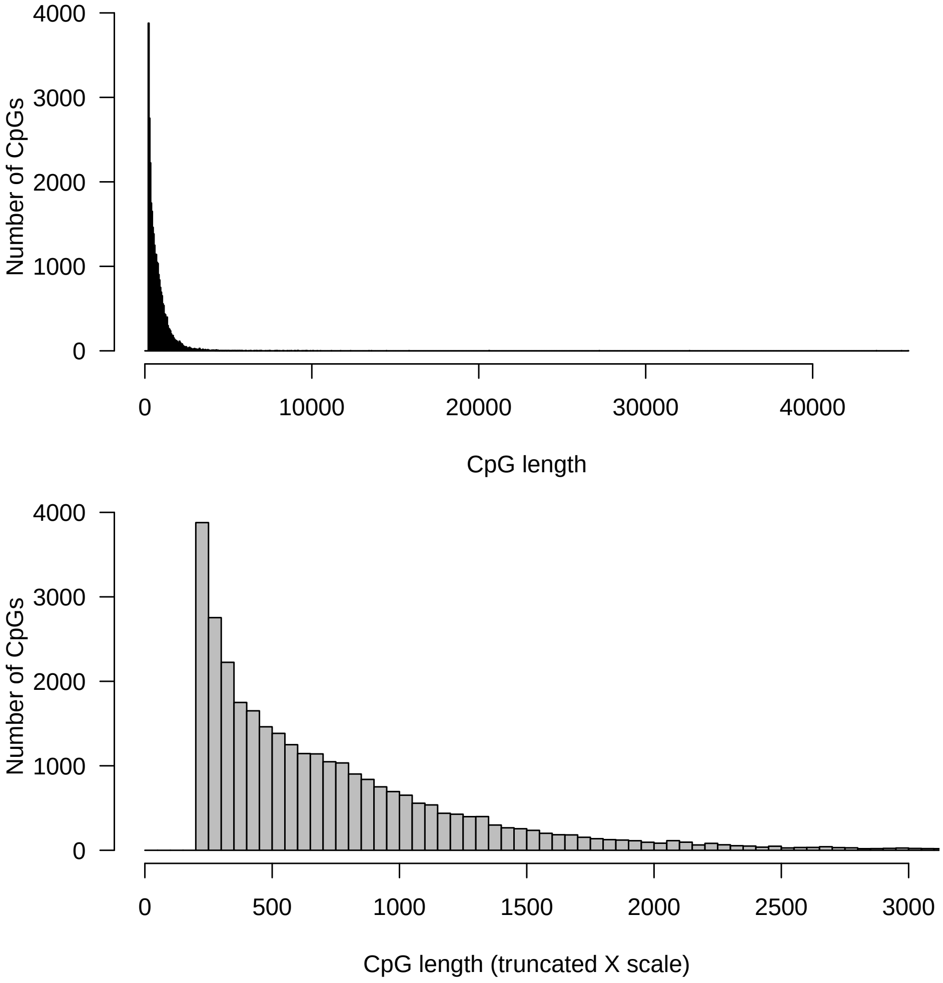


Figure 1: Distribution of lengths for all the CpG islands annotated in the Human genome.

```
## Check that the rows of the transition matrix sum to 1
## (given some rounding errors)
# apply(transitions.CpG[2:5], 1, sum)
# apply(transitions.Bg[2:5], 1, sum)
```

```
## Define a function that draws a heatmap from a transition matrix
transition.heatmap <- function(
  transitions,
  main = "Transitions",
  col.palette = gray.colors(n = 100, start = 1, end = 0),
  breaks = (0:length(col.palette))/(n.colors*2)) {


  n.colors <- length(col.palette)
  h <- heatmap.2(
    as.matrix(transitions),
    main = main,
    cellnote = round(digits = 3, as.matrix(transitions)),
    trace = "none",
    margins = c(4, 4),
    #    offsetRow = -10,
    breaks = breaks,
    key = FALSE,
    srtCol = 0,
    notecol = "black",
    notecex = 1.2,
    col = col.palette,
    scale = "none",
    las = 1,
    Rowv = FALSE, Colv = FALSE, dendrogram = "none")
  return(h)
  #  grab_grob()
}
```

```
## Draw transition heatmap
heatmap.CpG <- transition.heatmap(transitions.CpG, "CpG islands")
```

```
## Draw transition heatmap
heatmap.Bg <- transition.heatmap(transitions.Bg, "Genomic background")
```

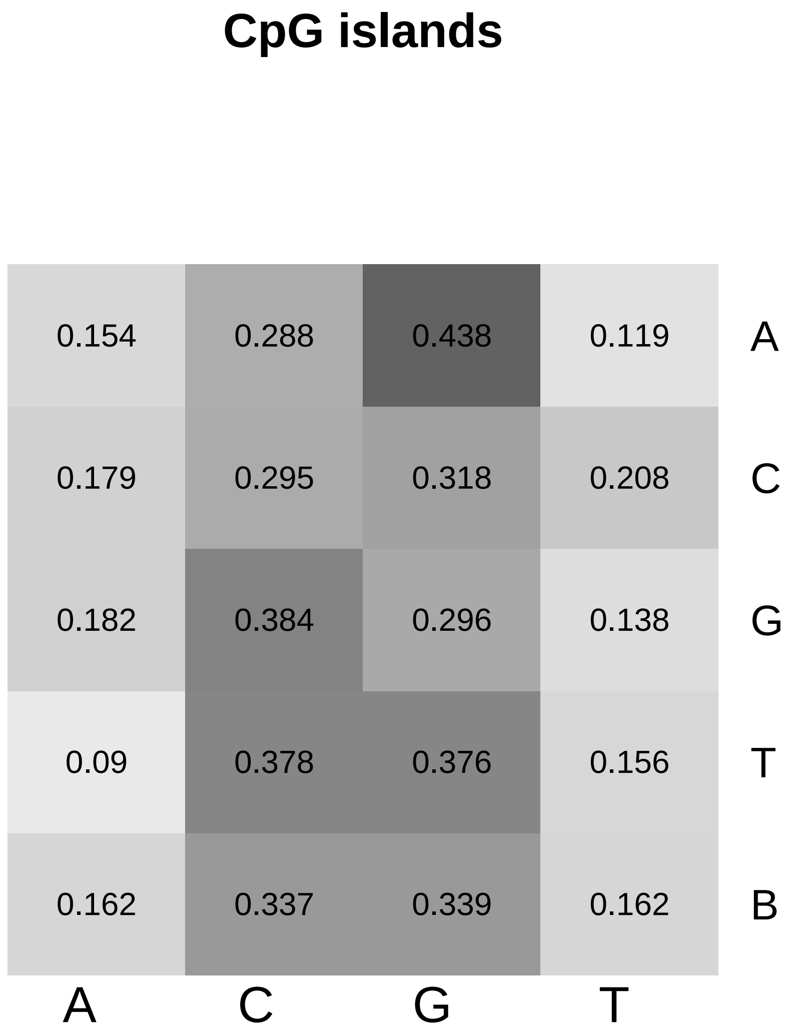
Discrimination between two models

Problem: for a given sequence of events (e.g. a nucleotidic sequence), identify the most likely Markov model.

Approach: compute the log-likelihood ratios (log-odd ratios) of the sequence probabilities computed using respectively the CpG island and genomic background models.

$$P_{\mathrm{CpG}}(S) = P_{\mathrm{CpG}}(S_1) \cdot \prod_{i=1}^{n-1} P_{\mathrm{CpG}}(S_{i+1}|S_i)$$

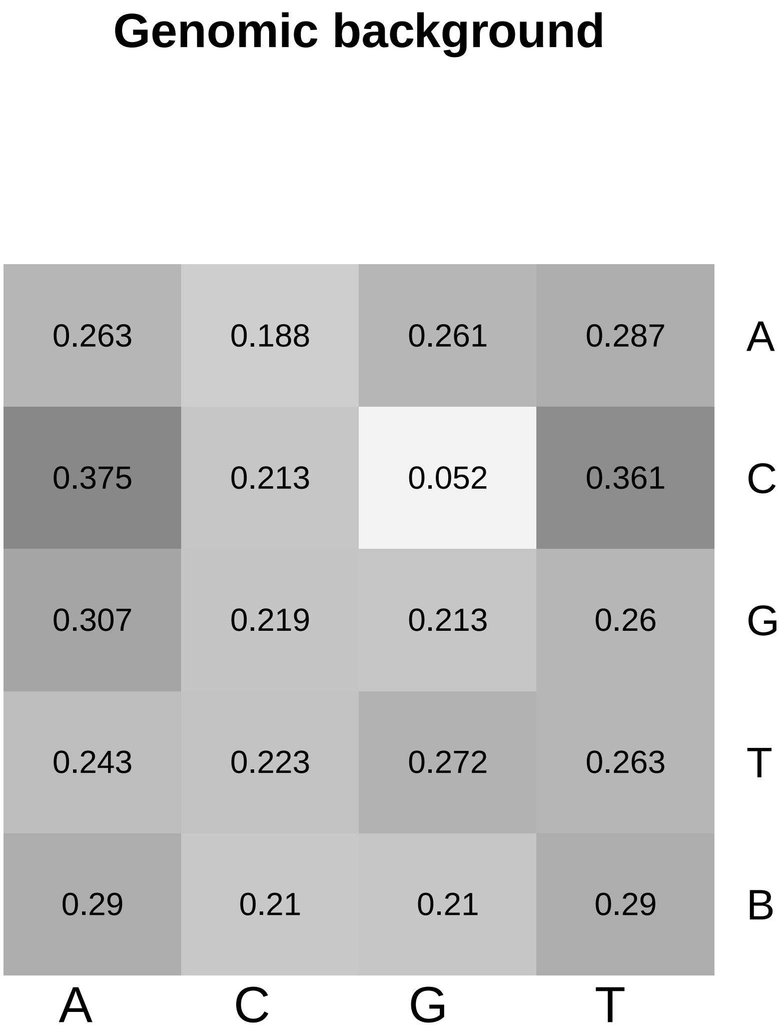
$$P_{\mathrm{Bg}}(S) = P_{\mathrm{Bg}}(S_1) \cdot \prod_{i=1}^{n-1} P_{\mathrm{Bg}}(S_{i+1}|S_i)$$



CpG islands

	A	C	G	T
A	0.154	0.288	0.438	0.119
C	0.179	0.295	0.318	0.208
G	0.182	0.384	0.296	0.138
T	0.09	0.378	0.376	0.156
B	0.162	0.337	0.339	0.162

Figure 2: Heatmap of the transition matrices from 1st order Markov models trained on CpG islands.



Genomic background

	A	C	G	T
A	0.263	0.188	0.261	0.287
C	0.375	0.213	0.052	0.361
G	0.307	0.219	0.213	0.26
T	0.243	0.223	0.272	0.263
B	0.29	0.21	0.21	0.29

Figure 3: Heatmap of the transition matrices from 1st order Markov models trained on genomic background.

$$L(S) = log\left(\frac{P_{\text{CpG}}(S)}{P_{\text{Bg}}(S)}\right)$$

where

- $L(S)$ is the log-likelihood of the sequence S,
- $P_{\text{CpG}}(S)$ the probability for this sequence to be generated by the CpG island model, and
- $P_{\text{Bg}}(S)$ its probability to be generated by the background model.

A more efficient approach : rather than computing the two sequence probabilities (as the product of transition probabilities), we can compute once and forever a matrix with tbe log-odds of residue transitions.

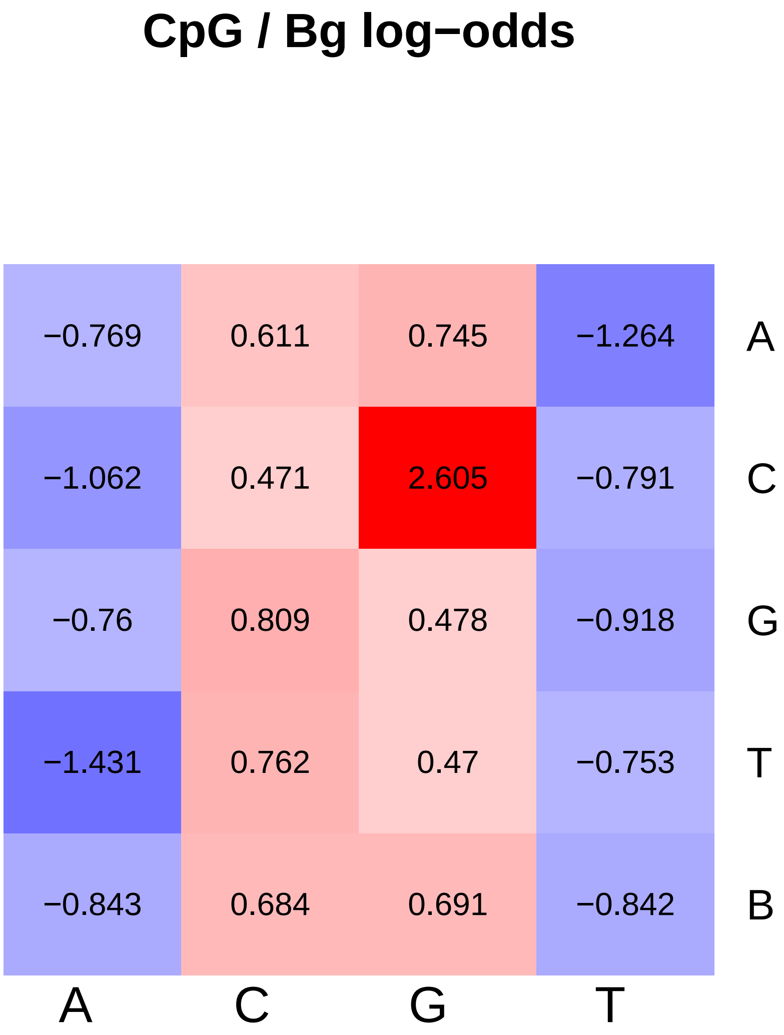
$$L(r_j|r_i) = log\left(\frac{P_{\text{CpG}}(r_j|r_i)}{P_{\text{Bg}}(r_j|r_i)}\right)$$

```r
my.palette <- colorRampPalette(c("blue", "white", "red"))(n = 100)

## Compute log-odds of transition frequencies
transition.log.odds <- log2(transitions.CpG / transitions.Bg)

max.lor <- ceiling(max(abs(range(transition.log.odds))))

heatmap.logodds <- transition.heatmap(
  transition.log.odds, "CpG / Bg log-odds",
  col.palette = my.palette,
#  breaks = seq(from = -max.lor, to = +max.lor, length.out = length(my.palette) + 1)
  breaks = length(my.palette) + 1
)
```

```r
# library(gridGraphics)
# library(grid)
# library(gplots)
# library(gridExtra)
#
# grab_grob <- function(){
#   grid.echo()
#   grid.grab()
# }
# heatmaps <- list(
#   CpG = one.heatmap(transitions.CpG, "CpG islands"),
#   Bg = one.heatmap(transitions.Bg, "Genomic background")
# )
# grid.newpage()
# grid.arrange(grobs = heatmaps, ncol = 2, clip = TRUE)
```

We can then use this log-odds matrix to compute the log-odds of a sequence as the sum of the transition log-odds.

$$L(S) = L_{\text{B}}(S_1) \cdot \sum_{i=1}^{n-1} L(S_{i+1}|S_i)$$



A	C	G	T	
−0.769	0.611	0.745	−1.264	A
−1.062	0.471	2.605	−0.791	C
−0.76	0.809	0.478	−0.918	G
−1.431	0.762	0.47	−0.753	T
−0.843	0.684	0.691	−0.842	B

Figure 4: Heatmap of the log-odds between CpG and genomic background.

Hidden Markov Model

We already dispose of the emission probabilities of the residues in the two respective states (CpG islands and genomic background). We now need to compute the transition probabilities between the states.

We can estimate these probabilities based on the following parameters

- genome size: $S_{Bg} = 3 \times 10^9$
- total size of the annotated CpG islands: $S_{CpG} = 24,200,434$
- number of CpG islands: $N_{CpG} = 31,144$