

# Four-state HMM:Analysis

Ricky Lim

May 2, 2013

Filename: 4StateAnalysis.Rnw

Working directory: /TEMP\_DDN/users/gfilion/rlim/H1\_ColorChromatin/4ColorStates

## 1 Goal

- Analyse the parameters of the four-state HMM

## 2 Load Datasets

Dataset was generated by G.Filion. In this dataset, it contains:

- Q: The transition matrix
- B: the emission matrix
- vPath: the fitted viterbi path
- emissionProb: the emission prob of each loci. In 3000bin-size, the coverage is 1031884 bins

```
load("input/fit41366702728.rda")
ls()

## [1] "HeatMapEm"    "em"           "fit4"         "get_heatMat"

print(names(fit4))

## [1] "loglik"      "Q"            "B"            "vPath"        "emissionProb"
## [6] "iterations"
```

## 3 Emission and Transition Matrix

### 3.1 Emission

```
em <- fit4$B
head(em)
colnames(em) <- c("Red", "Pink", "Yellow", "Black")

library(gplots)
get_heatMat <- function(mat_, margins, Colv, Rowv, cutZero) {
```

```

# construct heatmap, given a matrix
heatmap.2(mat_, col = cm.colors(256), Colv = Colv, Rowv = Rowv, dendrogram = "none", trace = "none",
  density.info = "none", scale = "none", keysize = 0.8, margins = margins, symbreaks = cutZero)
}

pdf("figs/em_matrix.pdf", useDingbats = FALSE)
HeatMapEm <- get_heatMat(em, margins = c(5, 10), Colv = NA, Rowv = TRUE, cutZero = FALSE)
dev.off()

```

## 3.2 Transition Matrix

```

trans <- fit4$Q
colnames(trans) <- c("Red", "Pink", "Yellow", "Black")
rownames(trans) <- c("Red", "Pink", "Yellow", "Black")

pdf("figs/trans_matrix.pdf", useDingbats = FALSE)
HeatMapTrans <- get_heatMat(trans, margins = c(5, 5), Colv = NA, Rowv = NA, cutZero = FALSE)
dev.off()

```

## 4 Enrichment over genome average

```

get_enrichmentPerGenomeAverage <- function(em_mat, fit_object) {

  # input: emmision matrix fit_object: fit object run from BaumWelch() function in
  # chromHMMatin
  averagePerState <- table(fit_object$vPath)/sum(table(fit_object$vPath))
  marksPerGenomeAverage <- em_mat %*% averagePerState
  mat_perGenomeAverage <- sweep(em_mat, 1, marksPerGenomeAverage, "/")
  return(mat_perGenomeAverage)
}

em_GenomeAverage <- get_enrichmentPerGenomeAverage(em, fit4)
get_heatMat(log(em_GenomeAverage), Colv = NA, Rowv = TRUE, margins = c(5, 10), cutZero = TRUE)

# thresholding the enrichment for 10 profiles
em_GenomeAverage_thresholded <- em_GenomeAverage
em_GenomeAverage_thresholded[em_GenomeAverage_thresholded < 1e-06] <- 1e-06

pdf("figs/enrich_matrix.pdf", useDingbats = FALSE)
get_heatMat(log(em_GenomeAverage_thresholded), Colv = NA, Rowv = TRUE, margins = c(5, 10),
  cutZero = TRUE)
dev.off()

```

Color Key

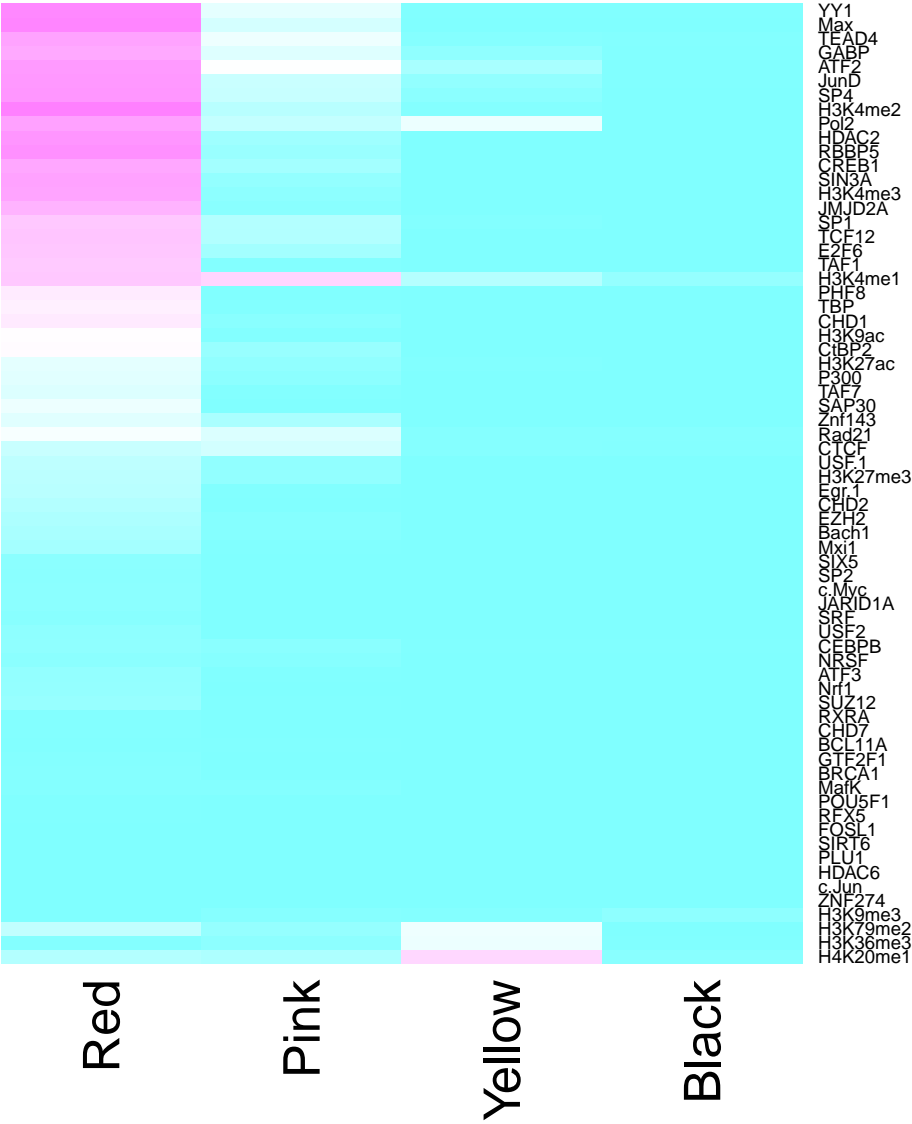
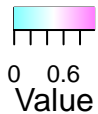


Figure 1: Emission Matrix

Color Key



0.2  
Value

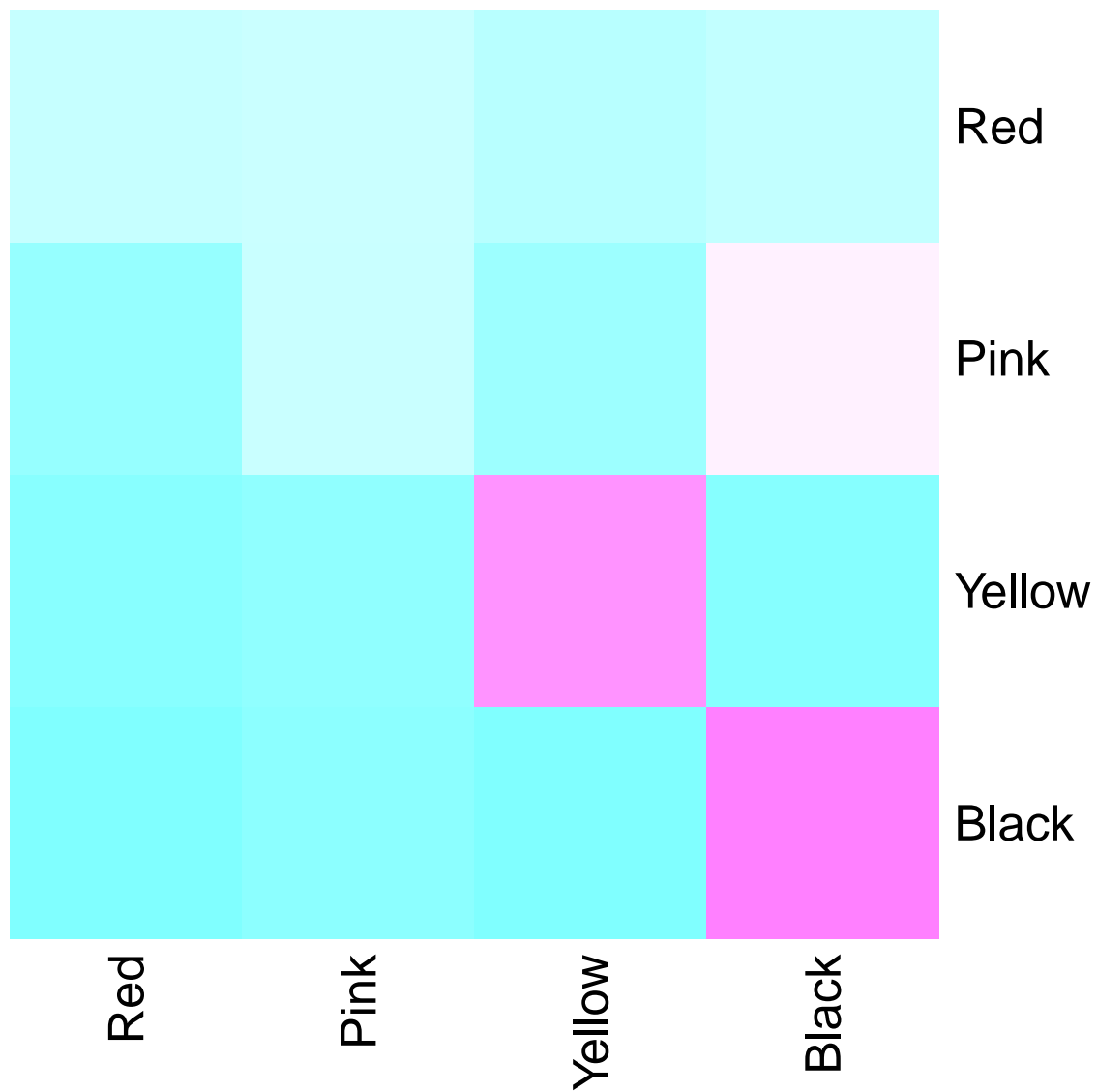


Figure 2: Transition Matrix

# Color Key

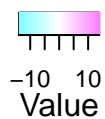


Figure 3: Enrichment Matrix

## 5 Coverage of Each Profile

Coverage of each profile was analysed as follows. The binary matrix consists of 0 and 1 that corresponds to non-target and target from all the profiles was used as input. For each profile, the coverage for each state is evaluated as the mean of the targets (1) for each state. The average genome-wide for each profile, defined as GW, was computed as the mean of the targets across the whole states or in this case along the genome.

```
coverage_data <- read.delim("input/discretized_with_NAs.txt.gz")
head(coverage_data)
coverage_data[1:5, 1:5]
dim(coverage_data)
# get the color state
state <- fit4$vPath
# exclude the chr column
coverage_data <- coverage_data[, c(2:69)]
# combine it with the color state
coverage_data <- cbind(coverage_data, state)
# remove the NAs
coverage_data <- coverage_data[complete.cases(coverage_data), ]
# get only the profiles without the color state
coverage_mat <- coverage_data[, c(1:68)]
# compute the genome-wide average of each profile
GW <- apply(coverage_mat, 2, mean)

library(plyr)

# compute the average of each state for each profile
mean_state <- aggregate(. ~ state, data = coverage_data, FUN = mean)
head(mean_state)
mean_state[1:4, 1:4]
# exclude the first column -> the state's info
coverage_state <- mean_state[, 2:ncol(mean_state)]
# combine it with the genome-wide average
coverage_profiles <- rbind(coverage_state, GW)
head(coverage_profiles)
rownames(coverage_profiles) <- c("Red", "Pink", "Yellow", "Black", "GW")
coverage_profiles <- t(coverage_profiles)

pdf("figs/coverage_matrix.pdf", useDingbats = FALSE)
get_heatMat(coverage_profiles, Colv = NA, Rowv = TRUE, margins = c(4, 10), cutZero = FALSE)
dev.off()
```

Color Key

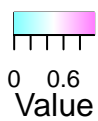


Figure 4: Coverage Matrix

## 6 Metainfo

```
sessionInfo()

## R version 2.15.0 (2012-03-30)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
## [1] C
##
## attached base packages:
## [1] grid      stats      graphics  grDevices datasets  utils      methods    base
##
## other attached packages:
## [1] Cairo_1.5-1      plyr_1.7.1      codetools_0.2-8 knitr_1.2
## [5] gplots_2.11.0    MASS_7.3-17     KernSmooth_2.23-7 caTools_1.13
## [9] bitops_1.0-4.1   gdata_2.10.6    gtools_2.6.2     vimcom_0.9-8
## [13] setwidth_1.0-3   cacheSweave_0.6-1 stashR_0.3-5      filehash_2.2-1
##
## loaded via a namespace (and not attached):
## [1] digest_0.5.2     evaluate_0.4.3  formatR_0.7      stringr_0.6      tools_2.15.0
```