BIOS 7659 Homework 8

Tim Vigers

24 November 2020

# 1. Cell-Type Composition

Read in the data:

baseDir1 = "blood/plate1"  
targets1 = read.metharray.sheet(baseDir1)

## [1] "blood/plate1/selected\_plate1.csv"

baseDir2 = "blood/plate2"  
targets2 = read.metharray.sheet(baseDir2)

## [1] "blood/plate2/selected\_plate2.csv"

targets = rbind(targets1, targets2)  
rgSet = read.metharray.exp(targets=targets,extended=T)  
sampleNames(rgSet) = rgSet[[1]]  
getManifest(rgSet)

## IlluminaMethylationManifest object  
## Annotation  
## array: IlluminaHumanMethylation450k  
## Number of type I probes: 135476   
## Number of type II probes: 350036   
## Number of control probes: 850   
## Number of SNP type I probes: 25   
## Number of SNP type II probes: 40

clindat = read.table("blood/demographic.txt", sep="\t", header=T)  
pData(rgSet)$Sample\_Group = clindat$Exposure  
pData(rgSet)$child\_sex = clindat$child\_sex

## a) Find differentially methylated positions based on exposure status

# SWAN normalization  
msetSWAN = preprocessSWAN(rgSet)  
dmp = dmpFinder(msetSWAN,pheno = msetSWAN$Sample\_Group,type = "continuous")

Here the unexposed group is the reference, so a negative value of beta in the DMP table indicates that the probe was hypomethylated in the exposed group (and a positive value indicates hypermethylation). There are no probes significant at the level, but there are 4 significant at the level:

dmp %>% rownames\_to\_column(var="Probe") %>% filter(pval < 1e-5) %>% mutate(Direction = ifelse(sign(beta)==-1,"Hypomethylated","Hypermethylated")) %>% select(Probe,beta,Direction,pval) %>% rename("Beta"=beta,"P Value"=pval) %>% flextable(.) %>% footnote(.,i=1,j=3,value = as\_paragraph("With respect to exposure"),ref\_symbols = "\*",part = "header") %>% set\_table\_properties(.,layout = "autofit",width = 0.75)

| Probe | Beta | Direction\* | P Value |
| --- | --- | --- | --- |
| cg25761397 | -0.3997037 | Hypomethylated | 1.332702e-06 |
| cg00244352 | -0.3431909 | Hypomethylated | 4.254079e-06 |
| cg15614119 | -0.2460500 | Hypomethylated | 5.528681e-06 |
| cg21088165 | -0.4816779 | Hypomethylated | 7.535189e-06 |
| \*With respect to exposure | | | |

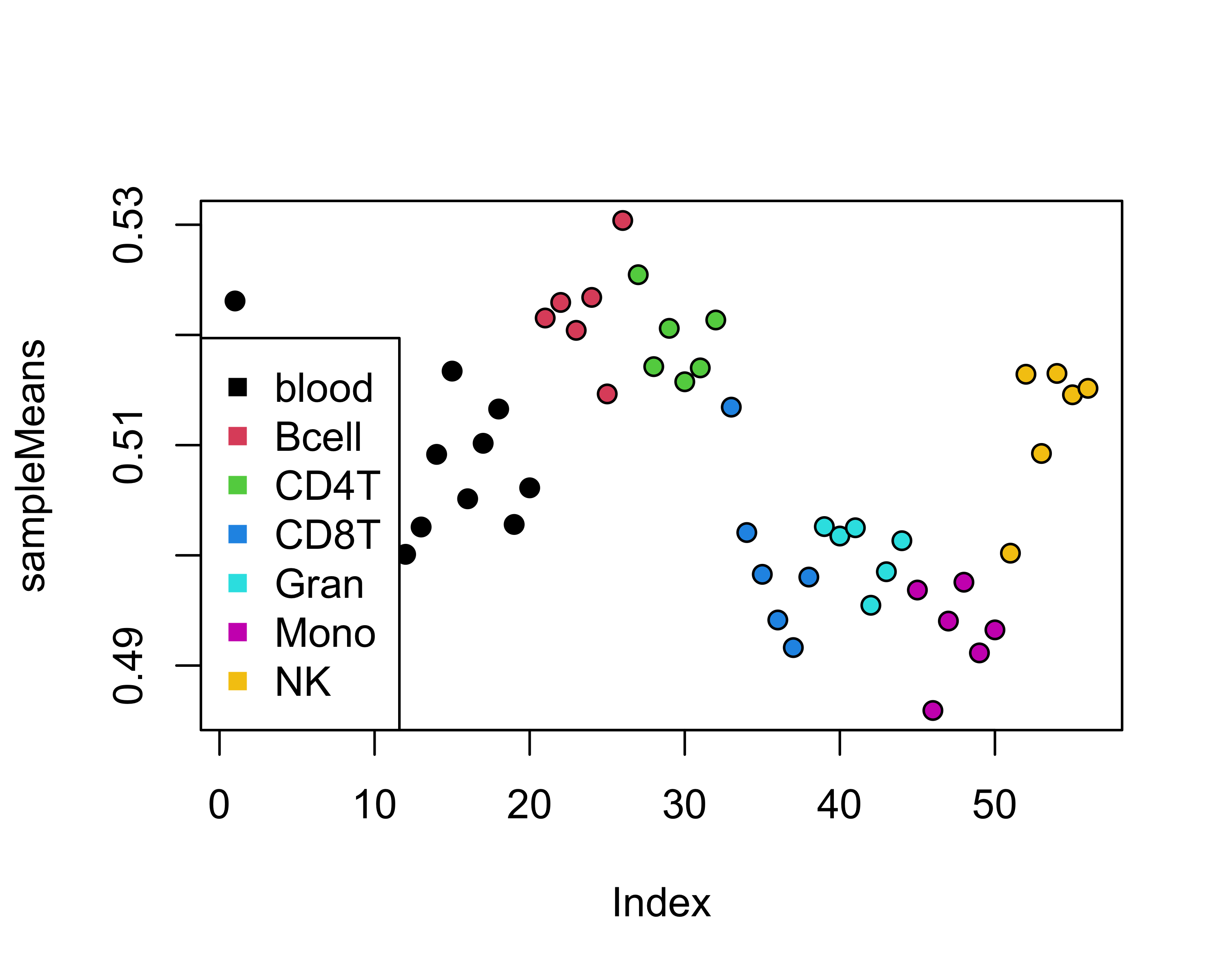
## b) Cell-type composition

Cell-type composition is important in methylation studies because methylation is a key factor in converting cells from a pluripotent progenitor to specific cell-type. As a result, differentially methylated regions (DMRs) can distinguish cell-types with high sensitivity and specificity [1]. In other words, if you are comparing two samples with different cell-type composition without some sort of adjustment, many DMRs that distinguish between them will likely be the result of cell-type composition rather than the exposure of interest. There are many ways to adjust for cell-type composition when cell-type isn’t directly measured, but the approach in Houseman et al. [1] is implemented in estimateCellCounts() in the minfi package, and that paper is frequently cited. The approach in Jaffe & Irizarry [2] is simply an adaptation of the Houseman algorithm for the 450K array. These approaches require reference data, but there are also reference-free methods such as singular value decomposition, linear models with principal components, SVA, RUV, RUVm, and others. However, the best option if possible is to measure complete blood counts rather than estimating cell types.

The Houseman algorithm takes an vector of methylation values for a homogenous cell population, and proposes the linear model where is a covariate vector indicating cell-type and is an matrix. Also, indexes the specimen and is an error vector. Next, they propose a similar model for the data without cell-type measurements: with the linking regression model where indexes samples of interest and is a matrix of errors. The math gets fairly complicated after this, so I won’t go into any more detail, but the points is that the cell-type mixture coefficients can be “recovered” from by estimating and via linear models using ordinary least squares, mixed effects models, limma, or surrogate variable analysis. The authors also recommend bootstrap procedures for estimating standard errors.

## c) Estimate cell counts

rgSet = read.metharray.exp(targets=targets,extended=F)  
cells = estimateCellCounts(rgSet,meanPlot = T,compositeCellType = "Blood",cellTypes = c("CD8T","CD4T","NK","Bcell","Mono","Gran"))



cells %>% as.data.frame(.) %>% flextable(.) %>% set\_table\_properties(.,layout = "autofit")

| CD8T | CD4T | NK | Bcell | Mono | Gran |
| --- | --- | --- | --- | --- | --- |
| 0.24473013 | 0.21221126 | 1.179639e-02 | 0.12211486 | 0.13014145 | 0.3087963 |
| 0.07834664 | 0.18399994 | 7.538122e-02 | 0.08690504 | 0.09264915 | 0.4918255 |
| 0.14571184 | 0.28648393 | 3.440698e-02 | 0.10895127 | 0.13164621 | 0.3002824 |
| 0.09831487 | 0.28551019 | 7.911824e-02 | 0.14558753 | 0.11676131 | 0.2611704 |
| 0.19685961 | 0.22271280 | 4.197430e-02 | 0.10295708 | 0.11272311 | 0.3369362 |
| 0.14994330 | 0.17078773 | 1.096162e-01 | 0.06853485 | 0.11621413 | 0.3868448 |
| 0.15172152 | 0.27356727 | 1.285225e-01 | 0.06991721 | 0.05034163 | 0.3301254 |
| 0.22450473 | 0.34554033 | 9.895694e-02 | 0.14560668 | 0.07658245 | 0.1192818 |
| 0.07678494 | 0.20818682 | 4.740060e-02 | 0.10282402 | 0.10651698 | 0.4705302 |
| 0.10358519 | 0.25831494 | 8.881471e-02 | 0.09553022 | 0.08252781 | 0.3840780 |
| 0.10510257 | 0.30098768 | -3.469447e-18 | 0.11750532 | 0.06812916 | 0.4073822 |
| 0.11391563 | 0.21834668 | 2.652012e-01 | 0.08757173 | 0.16612150 | 0.1376624 |
| 0.02934473 | 0.11473780 | 1.011562e-01 | 0.07603434 | 0.10906073 | 0.5655494 |
| 0.02019483 | 0.07077094 | 7.098415e-02 | 0.08893996 | 0.11387411 | 0.6443090 |
| 0.18778138 | 0.14126929 | 1.800243e-01 | 0.07174439 | 0.07516621 | 0.3642867 |
| 0.12504738 | 0.12085375 | 3.051989e-02 | 0.08876738 | 0.10867516 | 0.5284240 |
| 0.18194673 | 0.20379202 | 5.891531e-02 | 0.09304232 | 0.11067444 | 0.3619681 |
| 0.09117649 | 0.11894417 | 1.483735e-01 | 0.11561809 | 0.20884701 | 0.3316215 |
| 0.13787875 | 0.15443145 | 1.035689e-01 | 0.09733477 | 0.11691844 | 0.3874561 |
| 0.12538598 | 0.20456790 | 5.828551e-03 | 0.12964674 | 0.10794735 | 0.4276791 |

The plot above shows the average methylation across cell-type differentiating probes, for our data and the reference data set (“FlowSorted.Blood.450k”). The means of the two datasets should be in a similar range. If not, this suggests large batch effects in the data and the composition estimates should not be trusted (the function automatically performs quantile normalization between the two datasets to reduce potential batch effects). This plot looks reasonable, so we can probably trust these cell-type estimates and repeat part a) with adjustment for cell-type composition. Because the cell proportions necessarily sum to 1, we technically should not be treating them as independent variables. One option to get around this issue is to simply drop one of the cell types from the model, so for the purposes of this homework I’ve decided not to include the NK cells because these generally had very small estimated proportions.

# Format data  
m = getM(msetSWAN)  
cells = cbind(msetSWAN$Sample\_Group,cells)  
colnames(cells)[1] = "Sample\_Group"  
cells = cells[,-which(colnames(cells)=="NK")]  
# Fit models with limma  
d = model.matrix(~ Sample\_Group + CD8T + CD4T + Bcell + Mono + Gran,as.data.frame(cells))  
fit = lmFit(m,d)  
sd = fit$stdev.unscaled  
beta = fit$coefficients[,"Sample\_Group"]  
# Calculate p values  
pval = 2\*pt(abs(beta / (sd[,"Sample\_Group"] \* fit$sigma)),  
 df = fit$df.residual,lower.tail = F)  
qval = p.adjust(pval,"fdr")  
dmp = cbind(fit$coefficients,pval)  
dmp = cbind(dmp,qval)

One thing I noticed when fitting these models is that many of the regression estimates for cell type are really large with correspondingly large standard errors. When I removed them from the model and re-ran the code I got the same results as dmpFinder() in part a), so I don’t think it’s a programming issue. These estimates may not be a problem, but with more time I would probably investigate them a little. However, for this homework I am assuming that the estimates are okay given that we aren’t interested in inference for cell type. Also, I did copy the idea of using limma::lmFit() from Karen Kanaster because it is so much faster than lm(), but the code above is my own.

There are 0 probes significant at the level, and 21 probes significant at the level:

dmp %>% rownames\_to\_column(var="Probe") %>% filter(pval < 1e-5) %>% mutate(Direction = ifelse(sign(beta)==-1,"Hypomethylated","Hypermethylated")) %>% select(Probe,beta,Direction,pval) %>% rename("Beta"=beta,"P Value"=pval) %>% flextable(.) %>% footnote(.,i=1,j=3,value = as\_paragraph("With respect to exposure"),ref\_symbols = "\*",part = "header") %>% set\_table\_properties(.,layout = "autofit",width = 0.75)

# INTERPRET RESULTS HERE

# 2. ChIP-Seq

Read in the data:

data(yeastCDS)  
allData = dataReading("./chipseq/tup1\_IP.txt", "./chipseq/mock\_IP.txt",  
 yeastSpecies = yeastCDS$Saccharomyces.cerevisiae)

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  
## [1] "Reading IP and control datasets... "  
## [1] "... done"  
## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  
## [1] ""

## a) What methods were used?

## b) Average number of sequences mapped

baseLineIP = baseLineCalc(allData$IPdata[,ncol(allData$IPdata)])  
baseLineControl = baseLineCalc(allData$controlData[,ncol(allData$controlData)])

The average number of sequences mapped in the Tup1 ChIP sample is 65.5266955 and the average number in the mock sample is 14.4172567, so the Tup1 ChIP sample appears to have a lot more reads.

## c) Peak detection

# Subset the data  
ip = allData$IPdata[allData$IPdata$V1 == "chrV",]  
control = allData$controlData[allData$controlData$V1 == "chrV",]  
peaks = peakDetection(ip$V3,control$V3,"chrV",baseLineIP = baseLineIP,baseLineControl = baseLineControl)

## [1] "1% of windows were analyzed"  
## [1] "5% of windows were analyzed"  
## [1] "10% of windows were analyzed"  
## [1] "20% of windows were analyzed"  
## [1] "50% of windows were analyzed"  
## [1] "60% of windows were analyzed"  
## [1] "70% of windows were analyzed"  
## [1] "80% of windows were analyzed"  
## [1] "90% of windows were analyzed"  
## [1] "100% of windows were analyzed"  
## [1] "66 significant window(s) were detected..."  
## [1] "... starting merging procedure"  
## [1] ""  
## [1] "# of detected basic peaks (bPeaks) : 17"  
## [1] ""  
## [1] "\*\* Saving chromosome information in PDF file:"  
## [1] "bPeaks\_results\_dataSummary.pdf"  
## [1] ""

## [1] "\*\* Bed file saving in:"  
## [1] "bPeaks\_results.bed"  
## [1] ""  
## [1] "\*\* Peak drawing in PDF file:"  
## [1] "bPeaks\_results\_bPeaksDrawing.pdf"

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

## Warning in (vecIP + 1)/(vecINPUT + 1): longer object length is not a multiple of  
## shorter object length

## Warning in log2(vecIP + 1) + log2(vecINPUT + 1): longer object length is not a  
## multiple of shorter object length

The peakDetection() function found 17 peaks.

## d)

## e) Peak location

# Find peaks  
peak\_location = peakLocation("./bPeaks\_results.bed",yeastCDS$Saccharomyces.cerevisiae)

## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  
## [1] "Opening BED file with peak information:"  
## [1] "./bPeaks\_results.bed"  
## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  
## [1] ""  
## [1] "Starting peak location regarding ORF/CDS positions..."  
## [1] ""  
## [1] "# of analyzed peaks: 17"  
## [1] "# of peaks UPSTREAM annotated CDS : 15"  
## [1] "# of peaks IN annotated CDS : 2"  
## [1] ""

## [1] "Saving the results in:"  
## [1] "bPeaksLocation\_peakLocation\_inPromoters.txt"  
## [1] "bPeaksLocation\_peakLocation\_inCDS.txt"  
## [1] "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

# Read results back in  
promoters = read.delim("./bPeaksLocation\_peakLocation\_inPromoters.txt",header = F)  
cds = read.delim("./bPeaksLocation\_peakLocation\_inCDS.txt",header = F)

Out of the 17 peaks detected, 15 are in promoters and 2 are in CDS (genes).

# References

1. Houseman EA, Accomando WP, Koestler DC, et al. DNA methylation arrays as surrogate measures of cell mixture distribution. BMC Bioinformatics. 2012;13(1):86. <doi:10.1186/1471-2105-13-86>
2. Jaffe AE, Irizarry RA. Accounting for cellular heterogeneity is critical in epigenome-wide association studies. Genome Biol. 2014;15(2):R31. <doi:10.1186/gb-2014-15-2-r31>