Cell deconvolution

Truong Nguyen

2025-06-07

# Loading packages

#BiocManager::install("FlowSorted.Blood.450k")  
#BiocManager::install("methylCC")  
library(FlowSorted.Blood.450k)

library(methylCC)  
library(minfi)  
library(tidyr)

library(dplyr)

library(tibble)  
library(ggplot2)  
library(tidyverse)

library(data.table)

library(compositions)

library(FactoMineR)   
library(factoextra)

library(vegan)

library(cluster)   
library(umap)   
library(pheatmap)  
library(RColorBrewer)  
library(gghalves)  
library(rpart)  
library(rpart.plot)  
library(devtools)

library(abind)  
library(sesame)

library(rtracklayer)

library(dplyr)  
library(magrittr)

library(S4Vectors)  
library(methylkey)

library(DT)  
library(SummarizedExperiment)  
library(limma)

library(ggrepel)  
library(GenomicRanges)  
library(ggpubr)  
library(DMRcate)

##

# Loading data

bulk\_sample <- read.csv("/Users/macos/Desktop/Job\_applications/VIB/Meth\_atlas/bulk\_samples.csv", stringsAsFactors = FALSE)  
dim(bulk\_sample)

## [1] 500 51

reference\_sample <- read.csv("/Users/macos/Desktop/Job\_applications/VIB/Meth\_atlas/reference\_samples.csv", stringsAsFactors = FALSE)  
dim(reference\_sample)

## [1] 333 8

# Loading cell deconvolution results

These downstream analyses were performed on the results of cell deconvolution using deconvolve.py (nloyfer/meth\_atlas)

fractions\_file <- "/Users/macos/Desktop/Job\_applications/VIB/Meth\_atlas/bulk\_samples\_deconv\_output.csv"   
frac <- fread(fractions\_file) |> column\_to\_rownames("V1") |> as.matrix()  
residuals\_file <- "/Users/macos/Desktop/Job\_applications/VIB/Meth\_atlas/bulk\_samples\_residuals.csv"   
resid <- fread(residuals\_file) |> column\_to\_rownames("V1") |> pull(Residuals)

# Descriptive plots

Cellular proportions (which sum to 1) were treated as compositional data and normalized by centered log-ratio (CLR) transformation. Euclidean distances on the CLR-transformed abundances were then used to cluster the samples. Dimensionality reduction by PCA and UMAP both supported two main clusters (k-silhouette analysis, k = 2): Cluster 1: immune-rich (Neutrophiles + Monocytes high prevalence) Cluster 2: non-immune or atlas others.

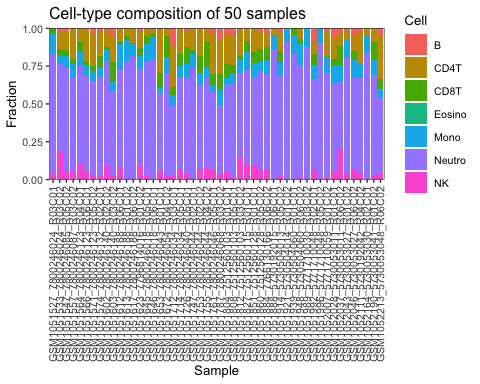
Decision tree provided a simple way to explain the clustering: IF B >= 0.0055 –> Cluster 1; else if Mono < 0.078 –> Cluster 1; else Cluster 2

Using Shannon diversity to assess the heterogeneity of 2 clusters: Cluster 1 is more heterogeneous (Median entropy = 1.9 vs 1.1 of cluster 2; two-sided Wilcoxon test, p = 0.007), suggesting mixed or inflammed samples.

#############################################################  
## 1. Cell proportions from deconvolve.py ##  
#############################################################  
  
# transposed version is convenient for sample-level ops  
frac\_t <- t(frac)   
  
# Fast check  
## 1-A: do fractions sum to ~1?  
round(rowSums(frac\_t), 3) |> summary()

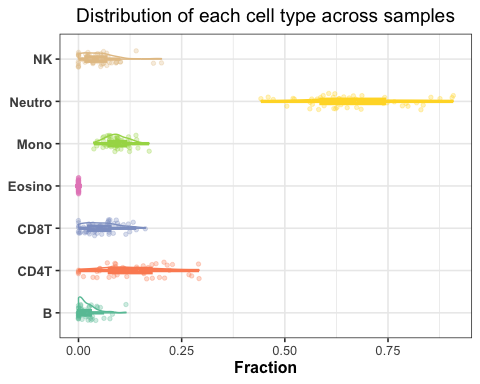
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.998 1.000 1.000 1.000 1.000 1.001

## 1-B: “other” row vs residuals  
if (exists("resid") && "other" %in% rownames(frac)) {  
 df\_tmp <- tibble(other = frac["other", ],  
 residual = resid)  
 ggplot(df\_tmp, aes(residual, other)) +  
 geom\_point() + geom\_smooth(method = "lm") +  
 ggtitle("Samples with high residuals are dominated by 'other'?")  
}  
  
# Descriptive plots  
## 2-A: stacked bar  
  
frac\_long <- frac\_t |>  
 as.data.frame() |>   
 rownames\_to\_column("Sample") |>   
 pivot\_longer(-Sample,  
 names\_to = "Cell",  
 values\_to = "Fraction")  
  
ggplot(frac\_long, aes(Sample, Fraction, fill = Cell)) +  
 geom\_bar(stat = "identity", width = 0.9) +  
 scale\_y\_continuous(expand = c(0, 0)) +  
 theme\_bw(base\_size = 10) +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1)) +  
 labs(title = "Cell-type composition of 50 samples")



## 2-B: violin per cell type  
  
ggplot(frac\_long, aes(x = Cell, y = Fraction, color = Cell)) +  
 # violins outlined in colour, no fill  
 geom\_half\_violin(side = "r", trim = TRUE, fill = NA, size = 0.5, alpha = 0.4) +  
 # boxplots outlined in same colour, no fill  
 geom\_boxplot(width = 0.1, fill = NA, size = 1.2, outlier.shape = NA, alpha =1) +  
 # optional: overlay points with matching border colour  
 geom\_jitter(width = 0.2, size = 1.2, alpha = 0.3, shape = 21, aes(fill = Cell)) +  
 coord\_flip() +  
 scale\_color\_brewer(palette = "Set2") +  
 scale\_fill\_brewer(palette = "Set2") + # for the point fills  
 labs(  
 title = "Distribution of each cell type across samples",  
 x = NULL,   
 y = "Fraction"  
 ) +  
 theme\_bw(base\_size = 12) +  
 theme(  
 legend.position = "none",  
 axis.text.y = element\_text(size = 10, face = "bold"),  
 axis.title.x = element\_text(face = "bold"),  
 plot.title = element\_text(hjust = 0.5)  
 )

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



#############################################################  
## 2. Compositional transform (CLR) & correlation heatmap ##  
#############################################################  
  
# replace zeros (needed before clr) – multiplicative replacement  
library(zCompositions)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

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

## Loading required package: NADA

## Loading required package: survival

##   
## Attaching package: 'NADA'

## The following object is masked from 'package:compositions':  
##   
## cor

## The following object is masked from 'package:IRanges':  
##   
## cor

## The following object is masked from 'package:S4Vectors':  
##   
## cor

## The following object is masked from 'package:stats':  
##   
## cor

## Loading required package: truncnorm

frac\_t\_mr <- zCompositions::cmultRepl(frac\_t, label = 0, method = "CZM")

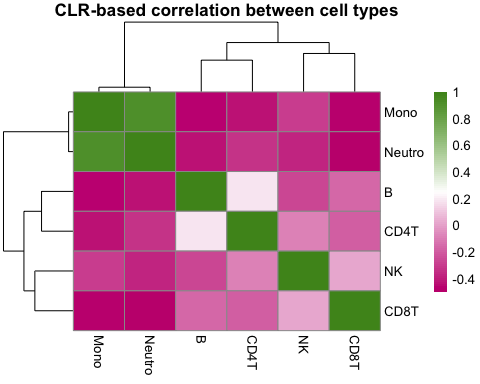
## Warning in zCompositions::cmultRepl(frac\_t, label = 0, method = "CZM"): Column no. 7 containing >80% zeros/unobserved values deleted (see arguments z.warning and z.delete).

## No. adjusted imputations: 36

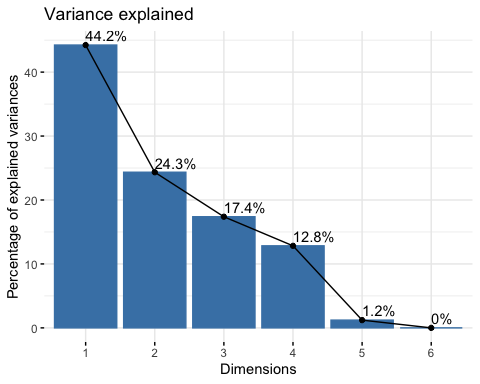
clr\_mat <- clr(frac\_t\_mr)   
  
dim(clr\_mat)

## [1] 50 6

# correlation of clr values  
corr <- cor(clr\_mat)  
pheatmap(corr,  
 clustering\_distance\_rows = "euclidean",  
 clustering\_distance\_cols = "euclidean",  
 color = colorRampPalette(c("#c51b7d", "white", "#4d9221"))(100),  
 main = "CLR-based correlation between cell types")



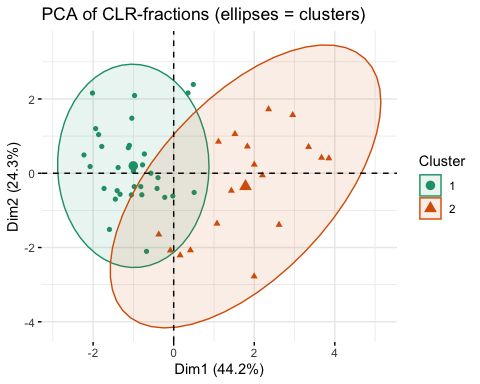
#############################################################  
## 3. PCA + unsupervised clustering ##  
#############################################################  
  
pca <- PCA(clr\_mat, graph = FALSE)  
fviz\_eig(pca, addlabels = TRUE, barfill = "steelblue") +  
 ggtitle("Variance explained")



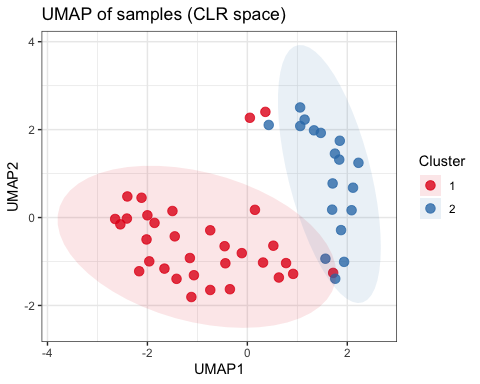
# choose k via silhouette width on first 5 PCs  
pc\_scores <- pca$ind$coord[, 1:5]  
sil\_widths <- sapply(2:7, function(k) {  
 pam(pc\_scores, k)$silinfo$avg.width  
})  
(k\_opt <- which.max(sil\_widths) + 1) # pick the best k

## [1] 2

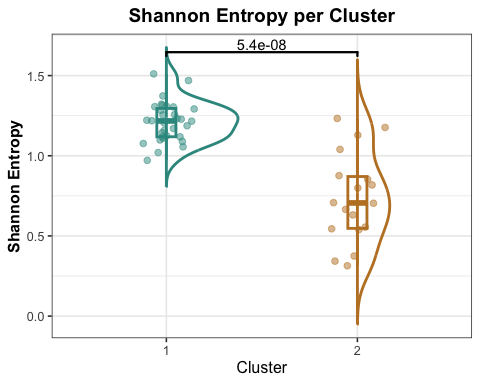
clusters <- pam(pc\_scores, k\_opt)$clustering  
  
fviz\_pca\_ind(pca,  
 geom.ind = "point",  
 col.ind = factor(clusters),  
 palette = "Dark2",  
 addEllipses = TRUE,  
 legend.title = "Cluster") +  
 ggtitle("PCA of CLR-fractions (ellipses = clusters)")



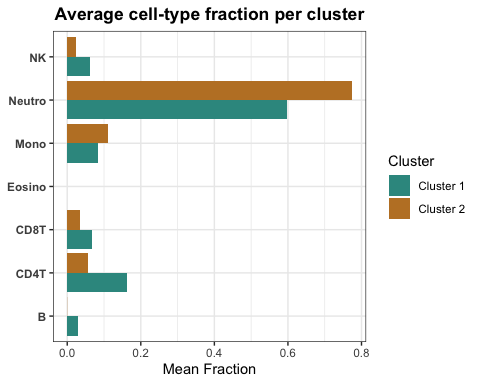
#############################################################  
## 5. UMAP (for non-linear clustering) ##  
#############################################################  
  
set.seed(123)  
Xclr\_mat <- as.matrix(clr\_mat)   
umap\_res <- umap(Xclr\_mat)  
  
df\_umap <- data.frame(UMAP1 = umap\_res$layout[, 1],  
 UMAP2 = umap\_res$layout[, 2],  
 Cluster = factor(clusters))  
  
ggplot(df\_umap, aes(UMAP1, UMAP2, color = Cluster)) +  
 geom\_point(size = 3, alpha = 0.8) +  
 stat\_ellipse(  
 aes(fill = Cluster), # also fill the ellipse  
 geom = "polygon", # draw as a filled polygon  
 alpha = 0.1, # transparency  
 color = NA, # no border  
 level = 0.95 # 95% confidence ellipse  
 ) +  
 scale\_color\_brewer(palette = "Set1") +  
 scale\_fill\_brewer(palette = "Set1") +  
 theme\_bw() +  
 labs(title = "UMAP of samples (CLR space)")



#############################################################  
## 5. Sample-level diversity indices ##  
#############################################################  
  
shannon <- vegan::diversity(frac\_t, index = "shannon") # vegan  
  
# relate entropy to clusters  
# define the pair to compare  
my\_comparisons <- list(c("1", "2"))  
  
df\_entropy <- tibble(  
 Shannon = shannon,  
 Cluster = factor(clusters)  
)  
  
ggplot(df\_entropy, aes(x = Cluster, y = Shannon, color = Cluster)) +  
 # half‐violin on right  
 geom\_half\_violin(  
 side = "r",  
 trim = FALSE,  
 fill = NA,  
 linewidth = 1,  
 alpha = 0.4  
 ) +  
 # boxplot outline  
 geom\_boxplot(  
 width = 0.1,  
 fill = NA,  
 outlier.shape = NA,  
 linewidth = 1,  
 alpha = 1  
 ) +  
 # jittered points  
 geom\_jitter(  
 width = 0.15,  
 size = 2,  
 alpha = 0.5  
 ) +  
 # custom cluster colors  
 scale\_color\_manual(values = c("1" = "#35978f", "2" = "#bf812d")) +  
 # draw the bracket and p‐value  
 stat\_compare\_means(  
 comparisons = my\_comparisons, # which groups to compare  
 method = "wilcox.test",  
 label = "p.format", # formatted p‐value  
 tip.length = 0.02, # length of the little “tees”  
 bracket.size = 0.8, # thickness of the bar  
 label.y = max(df\_entropy$Shannon) \* 1.05 # put text just above the tallest point  
 ) +  
 theme\_bw(base\_size = 12) +  
 labs(  
 title = "Shannon Entropy per Cluster",  
 x = "Cluster",  
 y = "Shannon Entropy"  
 ) +  
 theme(  
 legend.position = "none",  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 axis.title.y = element\_text(face = "bold")  
 )



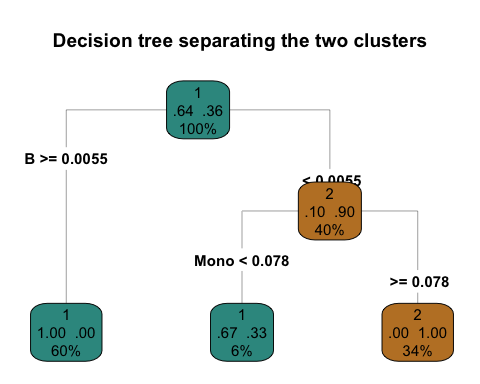
#############################################################  
## 6. Elucidating the clusters ##  
#############################################################  
  
X <- frac\_t  
  
# after PAM or k-means:  
clusters <- pam(pc\_scores, k\_opt)$clustering  
  
mean\_by\_cl <- X %>%  
 as.data.frame() %>%  
 mutate(Cluster = clusters ) %>%  
 group\_by(Cluster) %>%  
 summarise(across(everything(), mean)) %>%  
 pivot\_longer(-Cluster, names\_to = "Cell", values\_to = "MeanFrac")  
  
my\_cols <- c(  
 "1" = "#35978f", # cluster 1  
 "2" = "#bf812d" # cluster 2  
)  
  
mean\_by\_cl\_c <- mean\_by\_cl %>%  
 mutate(Cluster = factor(Cluster, levels = c(1,2),  
 labels = c("Cluster 1","Cluster 2")))  
  
ggplot(mean\_by\_cl\_c, aes(Cell, MeanFrac, fill = Cluster)) +  
 geom\_col(position = "dodge") +  
 scale\_fill\_manual(  
 values = c("Cluster 1" = "#35978f", "Cluster 2" = "#bf812d")  
 ) +  
 coord\_flip() +  
 theme\_bw() +  
 labs(  
 title = "Average cell-type fraction per cluster",  
 x = NULL,  
 y = "Mean Fraction"  
 ) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 axis.text.y = element\_text(face = "bold"), # ← bold tick labels  
 axis.title.y = element\_text(face = "bold"), # ← bold axis title  
 legend.position = "right"  
 )



#############################################################  
## 7. Decision Tree for cluster elucidation ##  
#############################################################  
df <- cbind(X, Cluster = clusters)  
df

## B NK CD4T CD8T Mono Neutro Eosino  
## GSM1052046\_5730053027\_R06C02 0.042 0.061 0.107 0.064 0.107 0.619 0  
## GSM1051753\_7800246044\_R02C02 0.015 0.061 0.169 0.115 0.080 0.561 0  
## GSM1052111\_5730192042\_R04C02 0.016 0.046 0.220 0.000 0.093 0.625 0  
## GSM1052007\_5771710059\_R01C02 0.023 0.016 0.165 0.005 0.105 0.687 0  
## GSM1051948\_5730504060\_R06C02 0.000 0.017 0.070 0.027 0.144 0.742 0  
## GSM1052043\_5730053027\_R01C02 0.004 0.040 0.074 0.000 0.117 0.765 0  
## GSM1051566\_7800246123\_R06C01 0.007 0.056 0.207 0.028 0.070 0.632 0  
## GSM1051939\_5730504060\_R01C01 0.000 0.000 0.000 0.046 0.124 0.830 0  
## GSM1051533\_7800246024\_R03C02 0.024 0.182 0.113 0.039 0.060 0.582 0  
## GSM1051613\_7800246188\_R01C02 0.000 0.000 0.052 0.024 0.139 0.785 0  
## GSM1051547\_7800246085\_R05C02 0.026 0.055 0.117 0.038 0.079 0.685 0  
## GSM1052164\_5730053030\_R02C01 0.000 0.000 0.045 0.000 0.120 0.835 0  
## GSM1051860\_7512560128\_R05C02 0.025 0.072 0.112 0.086 0.059 0.646 0  
## GSM1051726\_7800246040\_R05C01 0.018 0.036 0.107 0.109 0.088 0.643 0  
## GSM1051827\_7512560115\_R01C01 0.000 0.102 0.080 0.073 0.121 0.623 0  
## GSM1051653\_7800246043\_R01C01 0.006 0.047 0.292 0.095 0.037 0.523 0  
## GSM1051787\_7800246068\_R03C02 0.076 0.035 0.228 0.098 0.103 0.460 0  
## GSM1051564\_7800246123\_R04C01 0.000 0.101 0.090 0.070 0.089 0.651 0  
## GSM1051557\_7800246087\_R03C02 0.007 0.054 0.196 0.033 0.093 0.617 0  
## GSM1051914\_5730504013\_R05C02 0.027 0.022 0.122 0.076 0.088 0.666 0  
## GSM1051623\_7796814016\_R01C02 0.018 0.092 0.111 0.052 0.096 0.631 0  
## GSM1051714\_7800246034\_R05C01 0.012 0.069 0.159 0.076 0.078 0.606 0  
## GSM1051645\_7800246018\_R05C01 0.000 0.026 0.051 0.053 0.122 0.748 0  
## GSM1051603\_7800246140\_R03C02 0.044 0.023 0.179 0.133 0.059 0.561 0  
## GSM1051617\_7800246188\_R06C02 0.000 0.000 0.094 0.012 0.074 0.819 0  
## GSM1051884\_7766130166\_R06C02 0.029 0.060 0.097 0.084 0.094 0.636 0  
## GSM1051612\_7800246188\_R06C01 0.010 0.080 0.078 0.019 0.171 0.642 0  
## GSM1051712\_7800246034\_R03C01 0.115 0.021 0.274 0.051 0.068 0.472 0  
## GSM1051996\_5771710048\_R05C02 0.000 0.000 0.000 0.000 0.095 0.905 0  
## GSM1052028\_5730053011\_R01C01 0.062 0.054 0.142 0.045 0.118 0.579 0  
## GSM1051571\_7800246123\_R05C02 0.034 0.022 0.215 0.035 0.069 0.626 0  
## GSM1051672\_7800246054\_R04C02 0.000 0.033 0.051 0.140 0.103 0.674 0  
## GSM1051888\_5730192015\_R06C01 0.005 0.000 0.045 0.003 0.093 0.854 0  
## GSM1051921\_5730504014\_R01C02 0.000 0.000 0.012 0.000 0.080 0.908 0  
## GSM1051812\_7512560103\_R03C02 0.007 0.140 0.178 0.015 0.092 0.567 0  
## GSM1051646\_7800246018\_R06C01 0.000 0.000 0.035 0.025 0.141 0.798 0  
## GSM1051808\_7512560103\_R05C01 0.053 0.008 0.188 0.056 0.074 0.621 0  
## GSM1052190\_5730053041\_R01C02 0.009 0.022 0.083 0.121 0.093 0.672 0  
## GSM1051574\_7800246132\_R02C01 0.027 0.023 0.157 0.074 0.063 0.656 0  
## GSM1051992\_5771710048\_R02C01 0.082 0.068 0.086 0.073 0.094 0.595 0  
## GSM1051804\_7512560103\_R01C01 0.031 0.059 0.209 0.092 0.045 0.565 0  
## GSM1052213\_5730053048\_R06C02 0.046 0.035 0.291 0.062 0.061 0.505 0  
## GSM1051527\_7800246024\_R03C01 0.000 0.042 0.000 0.040 0.124 0.794 0  
## GSM1051602\_7800246140\_R02C02 0.005 0.092 0.122 0.017 0.074 0.691 0  
## GSM1051747\_7800246044\_R02C01 0.000 0.000 0.141 0.027 0.093 0.739 0  
## GSM1051755\_7800246044\_R04C02 0.004 0.075 0.089 0.162 0.076 0.593 0  
## GSM1051851\_7512560128\_R01C01 0.000 0.091 0.158 0.071 0.094 0.586 0  
## GSM1052037\_5730053011\_R06C02 0.030 0.201 0.088 0.124 0.116 0.442 0  
## GSM1051989\_5771710030\_R03C02 0.000 0.000 0.000 0.000 0.124 0.876 0  
## GSM1051761\_7800246046\_R05C01 0.033 0.060 0.268 0.074 0.044 0.520 0  
## Cluster  
## GSM1052046\_5730053027\_R06C02 1  
## GSM1051753\_7800246044\_R02C02 1  
## GSM1052111\_5730192042\_R04C02 1  
## GSM1052007\_5771710059\_R01C02 1  
## GSM1051948\_5730504060\_R06C02 2  
## GSM1052043\_5730053027\_R01C02 2  
## GSM1051566\_7800246123\_R06C01 1  
## GSM1051939\_5730504060\_R01C01 2  
## GSM1051533\_7800246024\_R03C02 1  
## GSM1051613\_7800246188\_R01C02 2  
## GSM1051547\_7800246085\_R05C02 1  
## GSM1052164\_5730053030\_R02C01 2  
## GSM1051860\_7512560128\_R05C02 1  
## GSM1051726\_7800246040\_R05C01 1  
## GSM1051827\_7512560115\_R01C01 2  
## GSM1051653\_7800246043\_R01C01 1  
## GSM1051787\_7800246068\_R03C02 1  
## GSM1051564\_7800246123\_R04C01 2  
## GSM1051557\_7800246087\_R03C02 1  
## GSM1051914\_5730504013\_R05C02 1  
## GSM1051623\_7796814016\_R01C02 1  
## GSM1051714\_7800246034\_R05C01 1  
## GSM1051645\_7800246018\_R05C01 2  
## GSM1051603\_7800246140\_R03C02 1  
## GSM1051617\_7800246188\_R06C02 2  
## GSM1051884\_7766130166\_R06C02 1  
## GSM1051612\_7800246188\_R06C01 1  
## GSM1051712\_7800246034\_R03C01 1  
## GSM1051996\_5771710048\_R05C02 2  
## GSM1052028\_5730053011\_R01C01 1  
## GSM1051571\_7800246123\_R05C02 1  
## GSM1051672\_7800246054\_R04C02 2  
## GSM1051888\_5730192015\_R06C01 2  
## GSM1051921\_5730504014\_R01C02 2  
## GSM1051812\_7512560103\_R03C02 1  
## GSM1051646\_7800246018\_R06C01 2  
## GSM1051808\_7512560103\_R05C01 1  
## GSM1052190\_5730053041\_R01C02 1  
## GSM1051574\_7800246132\_R02C01 1  
## GSM1051992\_5771710048\_R02C01 1  
## GSM1051804\_7512560103\_R01C01 1  
## GSM1052213\_5730053048\_R06C02 1  
## GSM1051527\_7800246024\_R03C01 2  
## GSM1051602\_7800246140\_R02C02 1  
## GSM1051747\_7800246044\_R02C01 2  
## GSM1051755\_7800246044\_R04C02 1  
## GSM1051851\_7512560128\_R01C01 2  
## GSM1052037\_5730053011\_R06C02 1  
## GSM1051989\_5771710030\_R03C02 2  
## GSM1051761\_7800246046\_R05C01 1

# X = samples × cell-types (still a matrix)  
# cluster\_id = vector of length nSamples with the PAM/k-means labels  
  
tree\_df <- X |>  
 as.data.frame() |> # <- matrix ➜ data-frame  
 mutate(Cluster = factor(clusters)) # add the label column  
  
# now run rpart on the data-frame  
library(rpart)  
tree <- rpart(  
 Cluster ~ .,   
 data = tree\_df,  
 method = "class",  
 parms = list(split = "information"),  
 control = rpart.control(cp = 0.01, minsplit = 5)  
)  
  
  
# Define a two-colour palette (one per cluster)  
my\_palette <- c("#35978f", "#bf812d")  
  
rpart.plot(  
 tree,  
 type = 4, # label all nodes  
 extra = 104, # show class and prob info  
 fallen.leaves = TRUE,  
   
 # --- NEW COLOUR OPTIONS ---  
 box.palette = my\_palette, # fill colours for the node boxes  
 branch.col = "darkgrey", # branch line colour  
 shadow.col = NA, # remove box shadow  
 nn = FALSE, # display node numbers  
 tweak = 1.2, # scale up text slightly  
   
 main = "Decision tree separating the two clusters"  
)



# Differential analysis

## Gene annotation

Two annotation manifests—HM450.hg38.manifest.gencode.v37.tsv.gz and EPIC.hg38.manifest.gencode.v37.tsv.gz—were used for gene annotation. Of the CpG sites in the bulk-sample dataset, 421 overlapped with the HM450 manifest and 444 overlapped with the EPIC manifest.

annot<-readr::read\_tsv("https://github.com/zhou-lab/InfiniumAnnotationV1/raw/main/Anno/HM450/HM450.hg38.manifest.gencode.v37.tsv.gz") %>%  
 dplyr::rename(Probe\_ID=probeID)

## Rows: 423764 Columns: 10  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: "\t"  
## chr (8): CpG\_chrm, probe\_strand, probeID, genesUniq, geneNames, transcriptTy...  
## dbl (2): CpG\_beg, CpG\_end  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

annot2<-readr::read\_tsv("https://github.com/zhou-lab/InfiniumAnnotationV1/raw/main/Anno/EPIC/EPIC.hg38.manifest.gencode.v37.tsv.gz") %>%  
 dplyr::rename(Probe\_ID=probeID)

## Rows: 731491 Columns: 10  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: "\t"  
## chr (8): CpG\_chrm, probe\_strand, probeID, genesUniq, geneNames, transcriptTy...  
## dbl (2): CpG\_beg, CpG\_end  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# checking the overlapping CpGs  
intersect <- intersect(bulk\_sample$CpGs, annot$Probe\_ID)  
length(intersect)

## [1] 444

## Density plot - QC

The β-value density plots across samples showed no obvious outliers, although a few curves deviated from the expected bimodal distribution. The M-value density plots exhibited unusually peaks in some samples, which could reflect batch effects, different tissue origins, or variability in experimental quality or preprocessing.

names(bulk\_sample)[names(bulk\_sample) == "CpGs"] <- "Probe\_ID"  
meth <- bulk\_sample  
rownames(meth) <- meth$Probe\_ID  
meth <- meth[ , -1]  
dim(meth)

## [1] 500 50

meth <- as.matrix(meth)  
  
n <- ncol(meth)   
  
densityPlot(  
 meth,  
 sampGroups = colnames(meth), # one group label per sample  
 sampNames = colnames(meth), # put sample names in the legend  
 pal = rainbow(n), # <- now a colour \*vector\*  
 legend = FALSE, # draw your own if you wish  
 xlab = expression(beta~value),  
 main = "Density of β values per sample"  
)

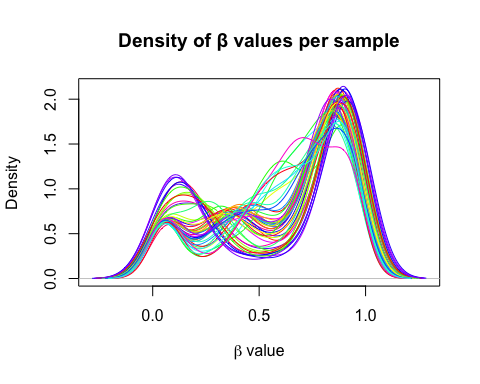
## Warning in plot.window(...): "sampNames" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "sampNames" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "sampNames" is not  
## a graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "sampNames" is not  
## a graphical parameter

## Warning in box(...): "sampNames" is not a graphical parameter

## Warning in title(...): "sampNames" is not a graphical parameter



## Filter NA CpG

To improve the power of downstream analyses, CpG probes with missing data in ≥ 20% of samples were filtered out.

# 1. choose the NA-threshold  
na\_cutoff <- 0.20   
  
# 2. calculate the fraction of NA in each probe (row)  
na\_frac <- rowMeans(is.na(meth))  
  
# 3. keep rows where that fraction ≤ threshold  
beta\_filt <- meth[ na\_frac <= na\_cutoff , ]  
  
# 4. (optional) report how many probes you kept / removed  
cat(  
 sprintf("Kept %s of %s probes (%.1f %%); removed %s (%.1f %%)\n",  
 nrow(beta\_filt), length(na\_frac),  
 100 \* nrow(beta\_filt) / length(na\_frac),  
 sum(na\_frac > na\_cutoff),  
 100 \* sum(na\_frac > na\_cutoff) / length(na\_frac))  
)

## Kept 500 of 500 probes (100.0 %); removed 0 (0.0 %)

## Convert to M-value

methM <- beta2m(meth)   
n <- ncol(methM)   
  
densityPlot(  
 methM,  
 sampGroups = colnames(methM),   
 sampNames = colnames(methM),   
 pal = rainbow(n),   
 legend = FALSE,   
 xlab = "M value",  
 main = "Density of M values per sample"  
)

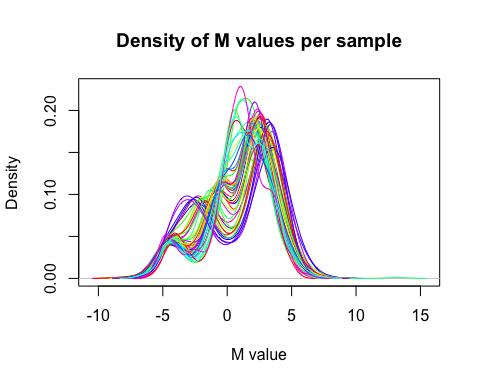
## Warning in plot.window(...): "sampNames" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "sampNames" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "sampNames" is not  
## a graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "sampNames" is not  
## a graphical parameter

## Warning in box(...): "sampNames" is not a graphical parameter

## Warning in title(...): "sampNames" is not a graphical parameter



## Create SummarizedExperiment object

# probes:  
  
key <- "Probe\_ID"   
anno\_df <- annot[, c(key, setdiff(names(annot), key))]  
  
## --- 3. produce a complete row-annotation table (same order as methM) ------  
all\_probes <- rownames(methM)  
  
row\_df <- anno\_df[ match(all\_probes, annot[[key]]), ] # keeps order of methM  
  
row\_df <- as.data.frame(row\_df)  
  
rownames(row\_df) <- all\_probes   
  
table(complete.cases(row\_df))

##   
## FALSE TRUE   
## 56 444

# samples:  
stopifnot(all(colnames(methM) %in% rownames(df)))  
col\_df <- df[ match(colnames(methM), rownames(df)), ]  
  
se <- SummarizedExperiment(  
 assays = list(Mvalue = methM),  
 colData = DataFrame(col\_df), # your sample metadata (clusters, etc.)  
 rowData = DataFrame(row\_df) # annotation + NA rows  
)  
  
se

## class: SummarizedExperiment   
## dim: 500 50   
## metadata(0):  
## assays(1): Mvalue  
## rownames(500): cg26930596 cg08884752 ... cg08314679 cg09113560  
## rowData names(10): Probe\_ID CpG\_chrm ... transcriptIDs distToTSS  
## colnames(50): GSM1052046\_5730053027\_R06C02 GSM1051753\_7800246044\_R02C02  
## ... GSM1051989\_5771710030\_R03C02 GSM1051761\_7800246046\_R05C01  
## colData names(8): B NK ... Eosino Cluster

## DMP findings

To investigate the differences of DNA methylation profiles between 2 clusters, linear regression model with limma was fitted with empirical Bayes method to moderate the standard errors of the estimated log-fold changes.

The top differentially methylated CpG sites (FDR-adjusted) were then visualized with volcano plots.

#############################################################  
## 0. Setup ##  
#############################################################  
  
m\_mat <- assay(se, "Mvalue")   
clusters <- factor(se$Cluster)   
table(clusters)

## clusters  
## 1 2   
## 32 18

#############################################################  
## 1. Design & contrasts ##  
#############################################################  
  
design <- model.matrix(~ 0 + clusters) # no intercept: one column per group  
colnames(design) <- levels(clusters) # e.g. "1" "2"  
  
colnames(design) <- c("Cluster1", "Cluster2")  
contrast.mat <- makeContrasts(  
 diff = Cluster2 - Cluster1, # Cluster 2 minus Cluster 1  
 levels = design  
)  
  
#############################################################  
## 2. Fit the linear model ##  
#############################################################  
  
fit <- lmFit(m\_mat, design)  
fit2 <- contrasts.fit(fit, contrast.mat)  
fit2 <- eBayes(fit2)  
  
#############################################################  
## 3. Extract results ##  
#############################################################  
  
top <- topTable(  
 fit2,  
 coef = "diff", # the contrast we named above  
 number = Inf, # all probes  
 sort.by = "P",  
 adjust = "BH" # FDR  
)  
  
# quick overview  
head(top[, c("logFC", "AveExpr", "t", "P.Value", "adj.P.Val")])

## logFC AveExpr t P.Value adj.P.Val  
## cg25605731 -1.171941 -1.0212304 -9.392382 6.007415e-13 3.003708e-10  
## cg21341487 -1.099075 -1.2880548 -9.126169 1.573542e-12 3.933855e-10  
## cg14059339 -1.372633 -0.9458381 -8.858620 4.169332e-12 5.364141e-10  
## cg14985891 -1.370708 -1.1506689 -8.850728 4.291313e-12 5.364141e-10  
## cg23363263 -1.342355 -0.4374194 -8.744296 6.334651e-12 5.739236e-10  
## cg19082496 -1.192959 -0.7805113 -8.624455 9.832121e-12 5.739236e-10

# how many genome-wide significant?  
sum(top$adj.P.Val < 0.05)

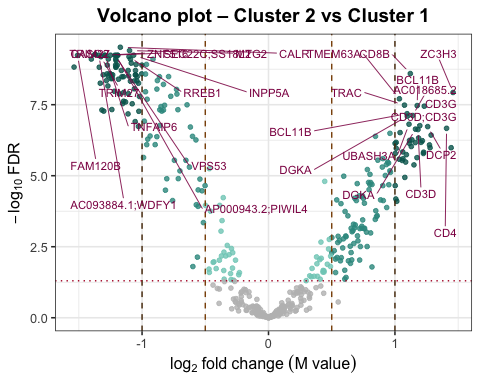
## [1] 302

# add annotation columns if you stored them in rowData(se)  
top\_anno <- cbind(  
 rowData(se)[rownames(top), ], # annotation  
 top # statistics  
)  
  
dim(top\_anno)

## [1] 500 16

#############################################################  
## 4. Save & plot ##  
#############################################################  
  
write.csv(top\_anno, "DMPs\_Cluster2\_vs\_Cluster1.csv", row.names = FALSE)  
  
# Volcano  
top\_anno <- as.data.frame(top\_anno)  
## 1 ──────────────────────────────────────────────────────────  
## Classify each CpG for colour & label  
## ────────────────────────────────────────────────────────────  
volc <- top\_anno %>%  
 mutate(  
 negLog10FDR = -log10(adj.P.Val),  
 absLFC = abs(logFC),  
 class = case\_when(  
 adj.P.Val < 0.05 & absLFC >= 1 ~ "FDR<0.05 & |LFC|≥1",  
 adj.P.Val < 0.05 & absLFC >= 0.5 ~ "FDR<0.05 & 0.5≤|LFC|<1",  
 adj.P.Val < 0.05 & absLFC < 0.5 ~ "FDR>0.05",  
 TRUE ~ "NS"  
 )  
 )  
  
## colour palette  
my\_cols <- c(  
 "FDR<0.05 & |LFC|≥1" = "#01665e", # red  
 "FDR<0.05 & 0.5≤|LFC|<1" = "#35978f", # blue  
 "FDR>0.05" = "#80cdc1",  
 "NS" = "grey"  
)  
  
label\_df <- volc %>%  
 filter(adj.P.Val < 0.05) %>% # only significant ones  
 group\_by(direction = sign(logFC)) %>% # -1 = hypo, 1 = hyper  
 slice\_min(order\_by = adj.P.Val, n = 15) %>% # 5 per direction  
 ungroup()  
  
## 2 ──────────────────────────────────────────────────────────  
## Volcano plot  
## ────────────────────────────────────────────────────────────  
library(ggplot2)  
library(ggrepel)  
library(dplyr)  
ggplot(volc, aes(x = logFC, y = negLog10FDR, colour = class)) +  
 geom\_point(size = 1.3, alpha = 0.8) +  
 scale\_colour\_manual(values = my\_cols, name = NULL) +  
   
 ## threshold lines  
 geom\_vline(xintercept = c(-1, 1), linetype = "dashed", colour = "#543005") +  
 geom\_vline(xintercept = c(-0.5, 0.5), linetype = "dashed", colour = "#8c510a") +  
 geom\_hline(yintercept = -log10(0.05), linetype = "dotted", colour = "#b2182b") +  
 geom\_text\_repel(  
 data = label\_df,  
 aes(label = genesUniq),   
 colour = "#8e0152",   
 max.overlaps = Inf,  
 size = 3,  
 segment.size = 0.3,  
 box.padding = 0.8,  
 point.padding= 0.6,  
 force = 1.5,  
 seed = 123  
 ) +  
   
 ## aesthetics  
 labs(  
 title = "Volcano plot – Cluster 2 vs Cluster 1",  
 x = expression(log[2]~fold~change~(M~value)),  
 y = expression(-log[10]~FDR)  
 ) +  
 theme\_bw(base\_size = 12) +  
 theme(  
 plot.title = element\_text(hjust = .5, face = "bold"),  
 legend.position = "none"  
 )

## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom\_text\_repel()`).



## DMR findings

Differentially methylated regions between Cluster 1 and Cluster 2 were identified using the DMRcate package.

# DMR finding with DMRcate  
## DMR analysis manually  
   
contrast.mat <- limma::makeContrasts(Cluster2 - Cluster1, levels = design)  
myannotation <- cpg.annotate(  
 datatype = "array",   
 what = "M",   
 object = methM, # M-value matrix (CpGs × samples)  
 design = design, # model matrix with your Cluster column  
 analysis.type = "differential",  
 contrasts = TRUE,  
 cont.matrix = contrast.mat,   
 coef = "Cluster2 - Cluster1",  
 arraytype = "450K" # or "450K"  
)

## Loading required package: IlluminaHumanMethylation450kanno.ilmn12.hg19

## Your contrast returned 302 individually significant probes. We recommend the default setting of pcutoff in dmrcate().

dmrcoutput<- DMRcate::dmrcate(myannotation,C=2, pcutoff=0.05, lambda = 1000)

## Fitting chr1...

## Fitting chr2...

## Fitting chr3...

## Fitting chr4...

## Fitting chr5...

## Fitting chr6...

## Fitting chr7...

## Fitting chr8...

## Fitting chr9...

## Fitting chr10...

## Fitting chr11...

## Fitting chr12...

## Fitting chr13...

## Fitting chr14...

## Fitting chr15...

## Fitting chr16...

## Fitting chr17...

## Fitting chr18...

## Fitting chr19...

## Fitting chr20...

## Fitting chr21...

## Fitting chr22...

## Demarcating regions...

## Done!

dmr\_table <- DMRcate::extractRanges(dmrcoutput, genome = "hg38")

## see ?DMRcatedata and browseVignettes('DMRcatedata') for documentation

## loading from cache

## Circosplot

A circos plot was used to visualize the chromosomal distribution of differentially methylated positions (DMPs).

results <- top\_anno %>%  
 mutate(deltaBeta = 2^(logFC/2) / (1 + 2^(logFC/2)) -  
 2^(-logFC/2) / (1 + 2^(-logFC/2)))  
  
## 2. merge so we have chr, pos, deltaBeta  
res\_annot <- results %>%  
 filter(!is.na(Probe\_ID)) # keep only mapped probes  
res\_annot <- res\_annot[res\_annot$adj.P.Val < 0.05, ]  
dim(res\_annot)

## [1] 270 17

## 3. build GRanges with the required metadata slots ------------------------  
ranges <- GRanges(  
 seqnames = res\_annot$CpG\_chrm,  
 ranges = IRanges(start = res\_annot$CpG\_beg,  
 end = res\_annot$CpG\_end), # CpGs: single-bp  
 deltabetas = res\_annot$deltaBeta,  
 midpoint = res\_annot$CpG\_beg # single-point so start==midpoint  
)  
  
## 4. draw the circos plot ---------------------------------------------------  
p <- circosplot(ranges, genome = "hg38")

## Loading required package: BSgenome.Hsapiens.UCSC.hg38

## Loading required package: BSgenome

## Loading required package: BiocIO

##   
## Attaching package: 'BiocIO'

## The following object is masked from 'package:rtracklayer':  
##   
## FileForFormat

print(p)

