

Epigenomics Roadmap Hands-on

Jan Izquierdo

2024-05-24

Data preparation

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##     filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union
```

```
source("/home/jj/bedtools2/BEDtoolsR.R")
```

```
gen_path<-"/home/jj/Desktop/Bioinformatics/2nd_year/3term/"
```

```
fullpath<-paste0(gen_path,"Omics_Techniques/Seminars/3.1-Epigenomics_roadmap/metadata.roadmap_clean.txt")
```

```
metadata<-read.table(fullpath ,sep="\t")  
#View(metadata)
```

```
#Distinguish Fetal and Non-fetal(Adult) samples
```

```
metadata$PERIOD<-ifelse(grepl("fetal|Fetal", metadata$V3), "Fetal", "Adult")
```

```
#Create a new column with UniqueNames incorporating the tissue, PERIOD and ID information
```

```
metadata$UniqueName<-paste0(metadata$V2, "_", metadata$PERIOD, "_", metadata$V1)
```

```
#Choosing tissues
```

```
tissues<-c("Brain", "brain", "Muscle", "muscle", "Digestive", "digestive", "Heart", "heart")
```

```
filtered_metadata<-dplyr::filter(metadata, V2 %in% tissues)$V1 #Teacher's method
```

```
filtered_metadata<-metadata[grepl(paste(tissues, collapse="|"), metadata$V2)==TRUE,] #database of only  
#the files that interest us (all that contain any of the words of tissues in column metadata$V2)
```

Reading files and storing them in a list

```
roadmap<-list()

for (f in filtered_metadata$V1){ #first column contains the id that differs the files from each other
   #(id of entry as well)
  filename<-paste0("all.mnemonics.bedFiles/",f,"_18_core_K27ac_mnemonics.bed.gz") #f is the ids
  if (file.exists(filename)){ #in case that a file does not exist, avoid errors
    print(f)
    roadmap[[f]]<-read.table(gzfile(filename)) #all contents of file to roadmap
    #"dictionary" with the ID as key
  }
}

#Changing ID "keys" to UniqueNames that we generated before
library(plyr)
names(roadmap)<-mapvalues(names(roadmap), from=metadata$V1, to=metadata$UniqueName)
```

```
names(roadmap)
```

```
## [1] "Brain_Adult_E071"      "Brain_Adult_E074"      "Brain_Adult_E068"
## [4] "Brain_Adult_E069"      "Brain_Adult_E072"      "Brain_Adult_E067"
## [7] "Brain_Adult_E073"      "Muscle_Adult_E100"      "Muscle_Adult_E108"
## [10] "Muscle_Fetal_E089"      "Muscle_Fetal_E090"      "Heart_Adult_E104"
## [13] "Heart_Adult_E095"      "Heart_Adult_E105"      "Heart_Adult_E065"
## [16] "Sm. Muscle_Adult_E078" "Sm. Muscle_Adult_E076" "Sm. Muscle_Adult_E103"
## [19] "Sm. Muscle_Adult_E111" "Digestive_Fetal_E092"  "Digestive_Fetal_E085"
## [22] "Digestive_Fetal_E084"  "Digestive_Adult_E109"  "Digestive_Adult_E106"
## [25] "Digestive_Adult_E075"  "Digestive_Adult_E101"  "Digestive_Adult_E102"
## [28] "Digestive_Adult_E079"  "Digestive_Adult_E094"
```

```
roadmap[["Muscle_Fetal_E089"]] %>% head()
```

```
##      V1      V2      V3      V4
## 1 chr10      0 115200  18_Quies
## 2 chr10 115200 119200 17_ReprPCWk
## 3 chr10 119200 119600  16_ReprPC
## 4 chr10 119600 120200  14_TssBiv
## 5 chr10 120200 121200 17_ReprPCWk
## 6 chr10 121200 122000  16_ReprPC
```

```
#avoid executing this chunk every time by saving roadmap
#saveRDS(roadmap, "roadmap.rds")
```

1. Calculate pairwise Jaccard index

Prepare the data and create the matrix

```
#Load roadmap if the previous chunk hasn't been executed
roadmap<-readRDS("roadmap.rds")

#Doing the intersection
```

```

state="1_TssA"
b1<-dplyr::filter(roadmap[["Muscle_Fetal_E090"]], V4==state) #V4 are the promoters
 #(if we did all genomes it'd almost be the same), we choose 1 promoter
b2<-dplyr::filter(roadmap[["Digestive_Adult_E102"]], V4==state)

bedTools.2jac(bed1=b1,bed2=b2)

```

```
## /home/jj/bedtools2/bin/bedtools jaccard -a /tmp/Rtmpt0bx7Z/file6134b368b8588 -b /tmp/Rtmpt0bx7Z/file
```

```

##           V1          V2          V3          V4
## 1 intersection  union  jaccard n_intersections
## 2      8459800 19145600 0.441867      13028

```

```

#do for all members in names(roadmap) and place in a matrix
rnames<-names(roadmap)
m<-matrix(nrow=length(rnames), ncol=length(rnames), dimnames=list(rnames)) #matrix of rnames dimensions
colnames(m)<-rnames

```

Fill the jaccard matrix

```

for (i in 1:length(rnames)){
  for (j in 1:length(rnames)){
    a<-dplyr::filter(roadmap[[i]], V4==state)
    b<-dplyr::filter(roadmap[[j]], V4==state)
    index_num<-bedTools.2jac(bed1=a,bed2=b) #MYFUNCTION #2jac for testing
    colnames(index_num)<-make.names(index_num[1,])
    index_num<-index_num[2,]
    m[i,j]<-index_num$jaccard #store jac2 indexes there #or intersection?
  }
}
saveRDS(m, "jaccard_matrix.rds")

```

Load the jaccard matrix

```
m<-readRDS("jaccard_matrix.rds")
```

2. Visualize Jaccard Index matrix in a heatmap, indicating tissue of origin and PERIOD

Prepare the data for the heatmap and create it

```

library(pheatmap)

annotation_col <- data.frame( Tissue = filtered_metadata$V2, Period = filtered_metadata$PERIOD,
                             row.names = filtered_metadata$UniqueName)

m <- as.matrix(m)
m <- apply(m, 2, as.numeric)

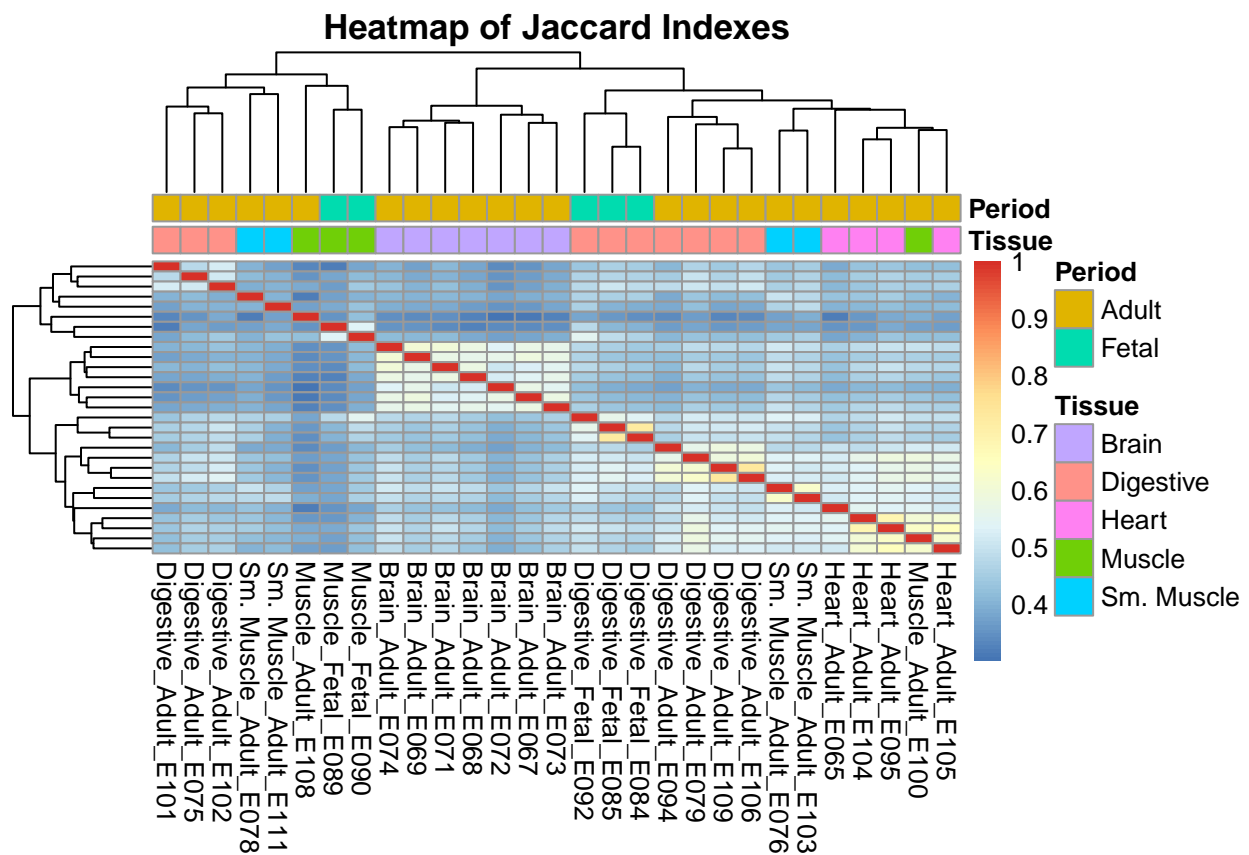
pheatmap(m,
  annotation_col = annotation_col,

```

```

main = "Heatmap of Jaccard Indexes",
cluster_rows = TRUE,
cluster_cols = TRUE,
show_rownames = FALSE,
)

```



3. Report jaccard index between sample E071 and sample E074

Set up variables and consult the index

```

#m<-readRDS("jaccard_matrix.rds")
#o bē
rownames(m)<-rnames

#E071 on filtered -> Name in UniqueNames -> m[name1] is row of name -> m[name1, name2]
#m[name1, name2] is row and column of names
E071_Unique<-filtered_metadata$UniqueName[filtered_metadata$V1=="E071"]
E074_Unique<-filtered_metadata$UniqueName[filtered_metadata$V1=="E074"]

m[E071_Unique, E074_Unique]

```

```
## [1] 0.597558
```

4. Perform multidimensional scaling on a distance matrix based on 1-jaccardIndex.

Creation of the 1-jaccard distance matrix and performing the multidimensional scaling

```
#distance matrix is 1-jaccard distance
dm<-1-m

#Perform the multidimensional scaling, k=4 for 4 dimensions ( using 4 for the next exercise )
multi_d_scaling <- cmdscale(dm, k = 4)
```

5. Plot 1:2 and 3:4 dimensions and color by tissue, shape by fetal/adult.

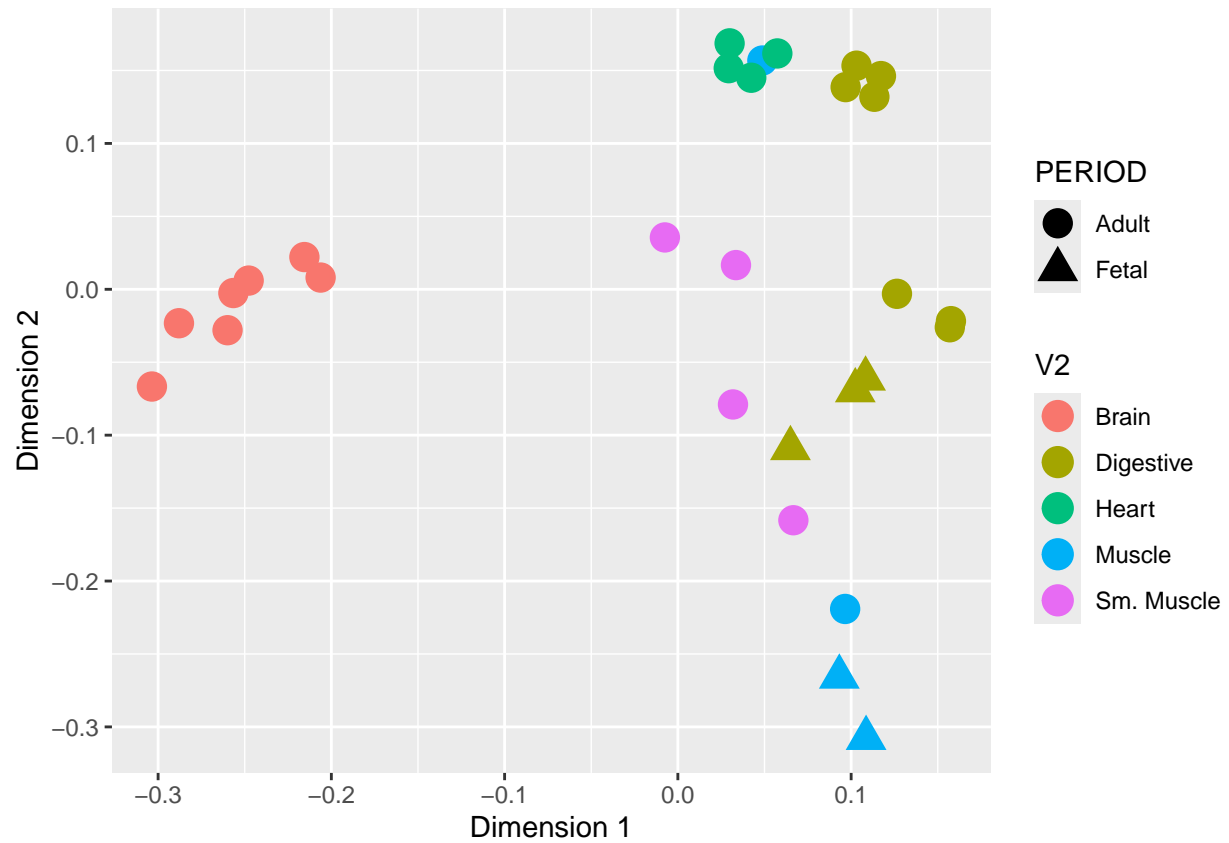
Create a new dataframe for plotting the dimensions

```
#rename the cols from [,1] to their corresponding dimension and transform it into a dataframe
colnames(multi_d_scaling)<-c("MDS1", "MDS2", "MDS3", "MDS4")
df_multi_d_scaling <- as.data.frame(multi_d_scaling)

#Create a column with the rownames, for easier join with filtered_metadata (needed for PERIOD and tissue)
df_multi_d_scaling$Name<-rownames(df_multi_d_scaling)
df_multi_d_scaling <- df_multi_d_scaling %>%
  left_join(filtered_metadata, by = c("Name" = "UniqueName"))
```

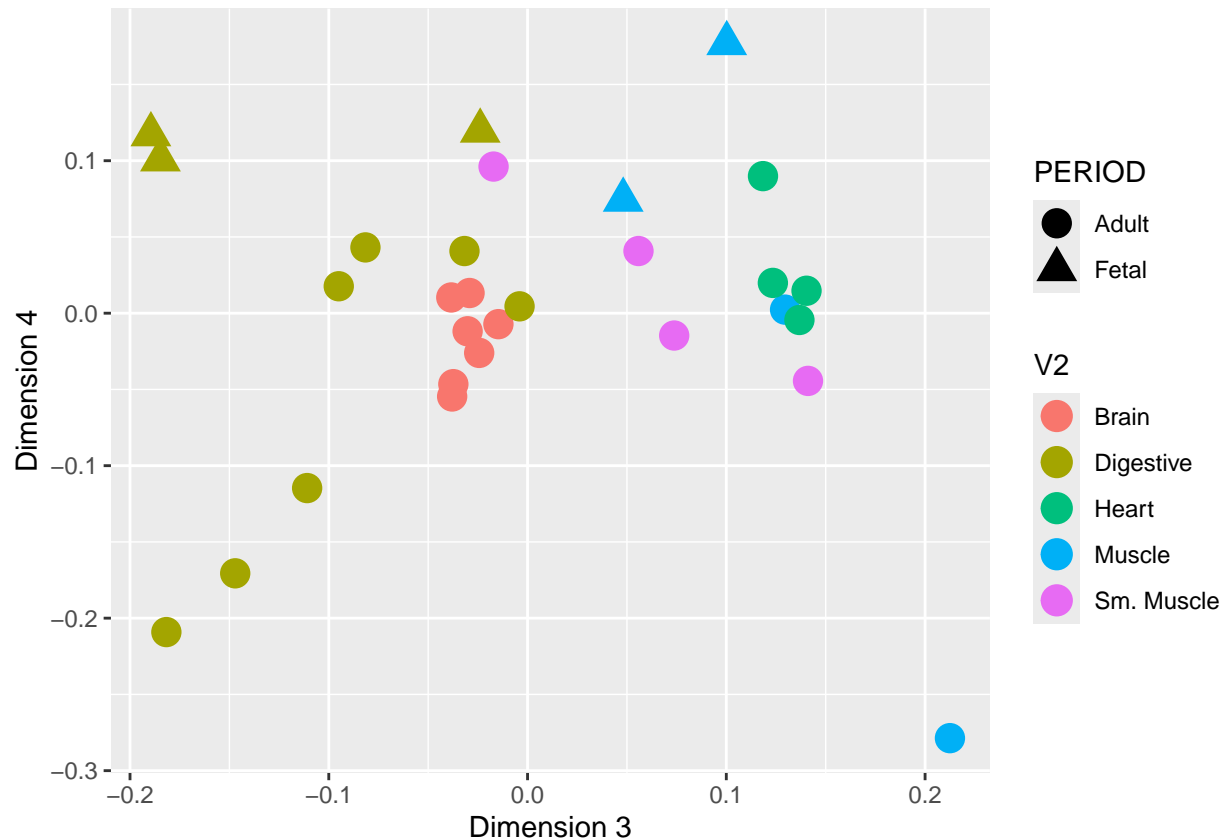
Plot for dimensions 1 and 2

```
library(ggplot2)
ggplot(df_multi_d_scaling, aes(x = MDS1, y = MDS2, color = V2, shape = PERIOD))+geom_point(size = 5)+
  labs(x="Dimension 1", y="Dimension 2")
```



Plot for dimensions 3 and 4

```
ggplot(df_multi_d_scaling, aes(x = MDS3, y = MDS4, color = V2, shape = PERIOD)) + geom_point(size = 5) +
  labs(x="Dimension 3", y="Dimension 4")
```



6. Compute hierarchical clustering in jaccard distance matrix with R function “hclust”, cut the dendrogram at different Ks using R function “cutree” and plot the dendrogram coloring the obtained clusters of samples. (You can use the R Package dendextend)

Prepare the data and compute the clustering

```
library(dendextend)

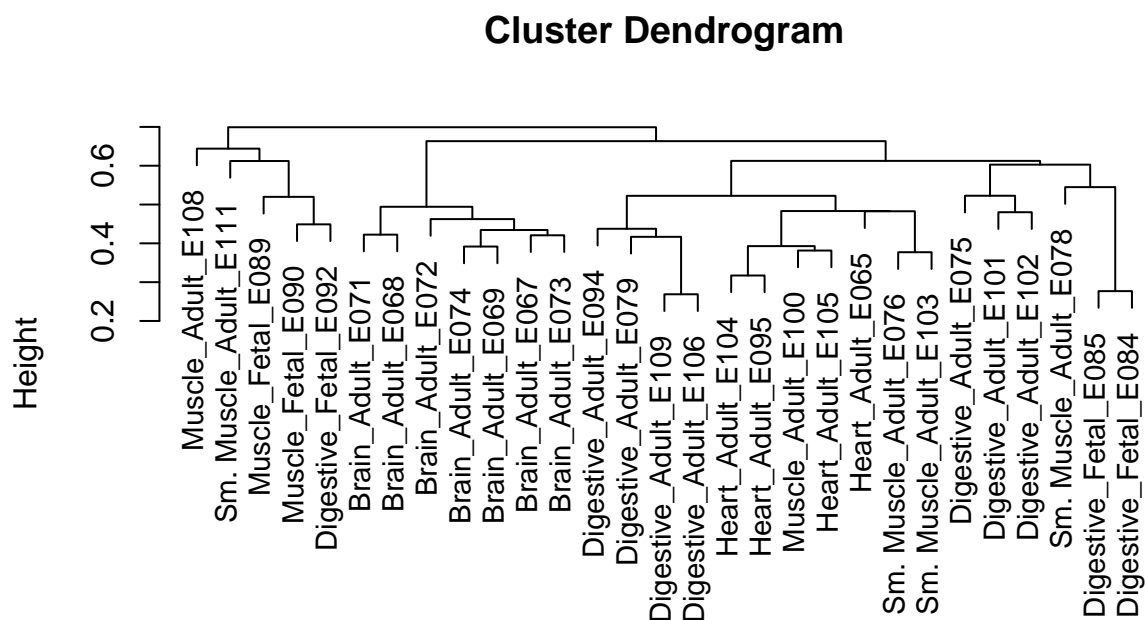
##
## -----
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
```

```
##
## Attaching package: 'dendextend'

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

dmatrix_prep<-as.dist(dm)
computed_clusters<-hclust(dmatrix_prep, method="complete")

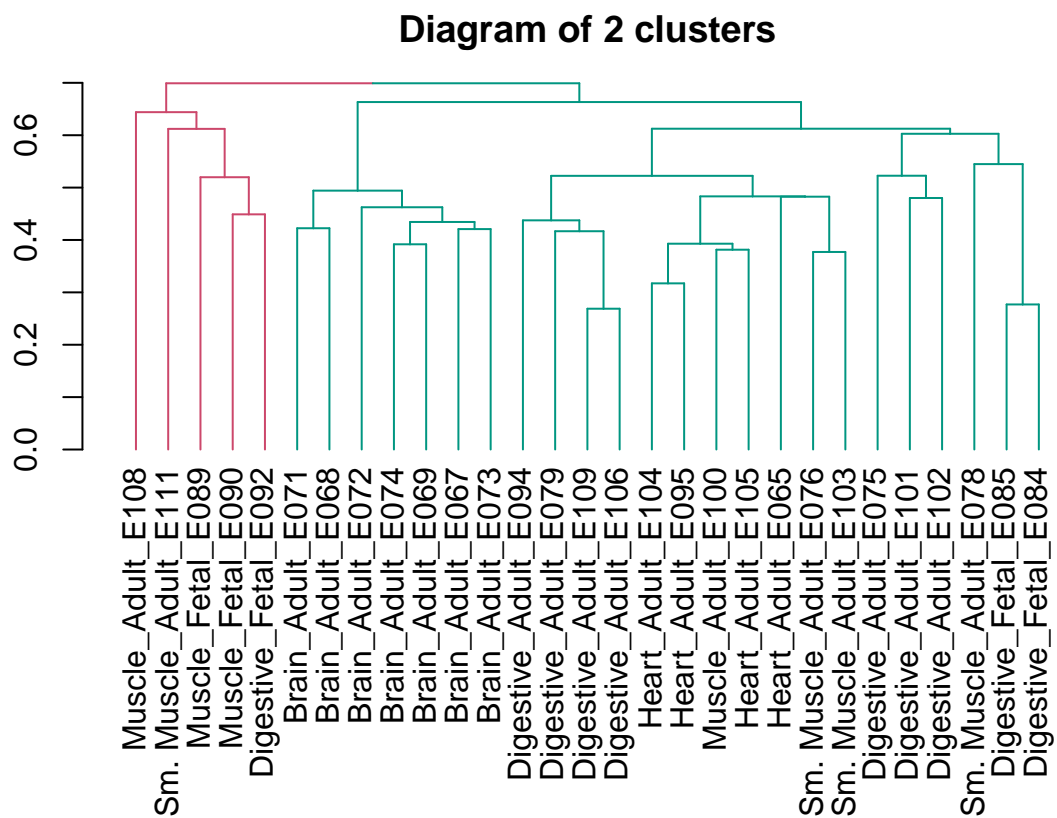
plot(computed_clusters, sub="", xlab="", cex=0.9)
```

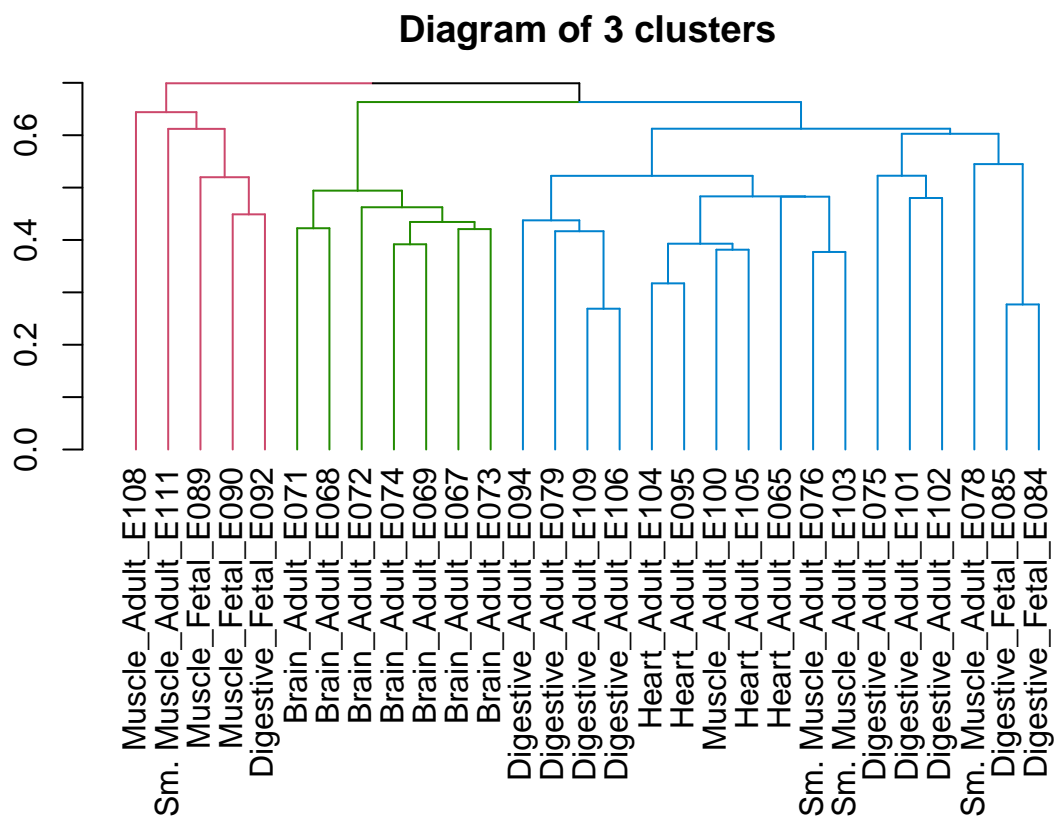


Cluster separation and plotting with cutree

```
cluster_numbers<-c(2, 3, 4, 6, 8)

#par(mfrow = c(1, length(cluster_numbers)/2))
for (cl in cluster_numbers){
  clusters <- cutree(computed_clusters, k = cl)
  dend <- color_branches(as.dendrogram(computed_clusters), k = cl)
  par(mar = c(10, 4, 2, 2)+0.1)
  plot(dend, main=paste0("Diagram of ", cl, " clusters"), cex=0.9)
}
```



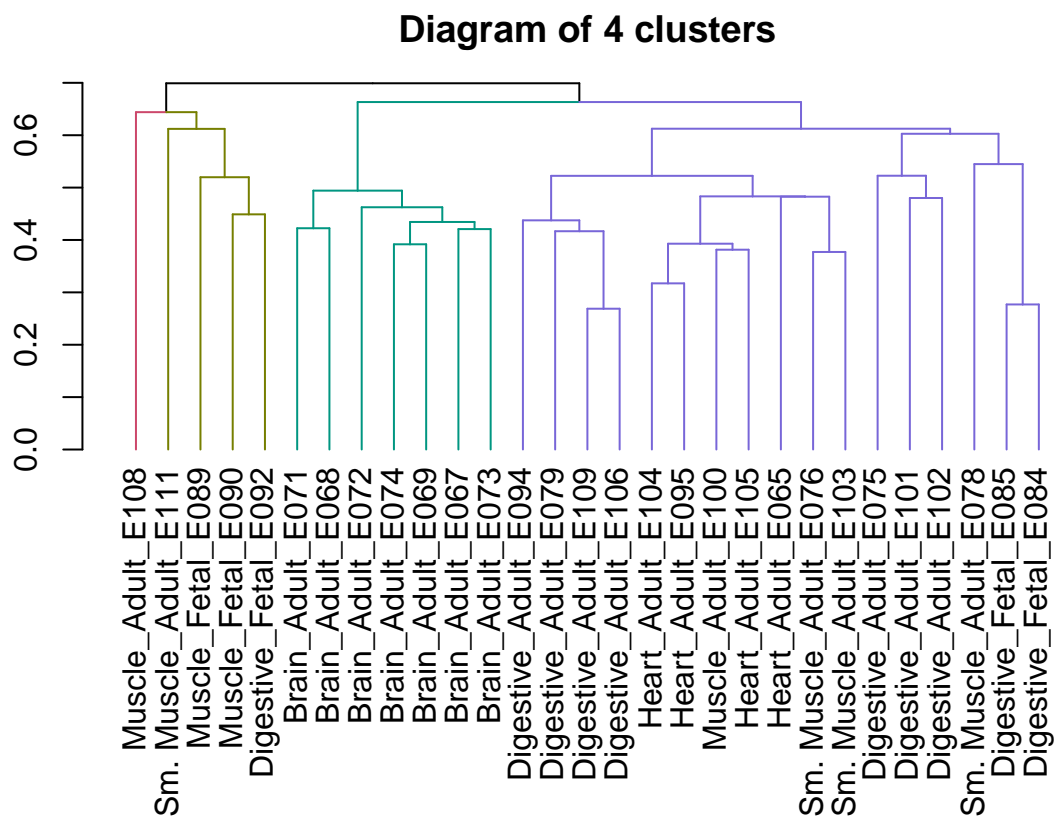
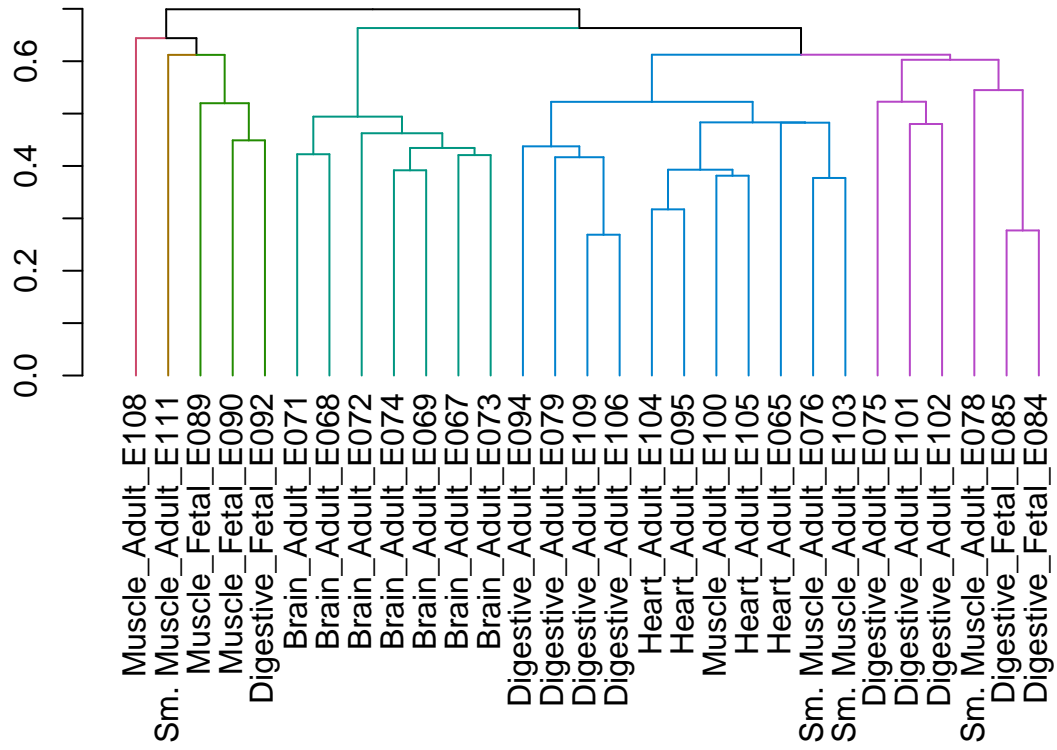
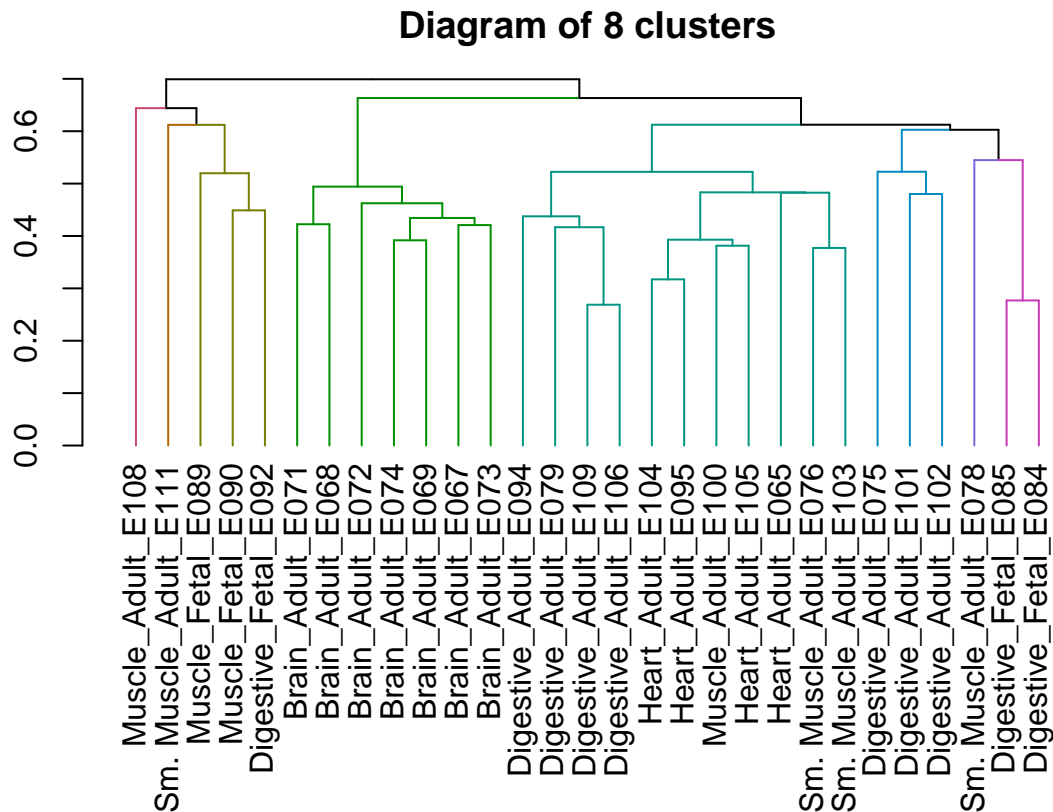


Diagram of 6 clusters





Do you see a logic in the clusters using epigenomes?

Most clusters seem to be separated by tissue and the separation occurs by earliest branch division (On 2 clusters, it divides by red and blue instead of creating a division out of green, which is the new cluster created in the 3 clusters plot)

Which major sub-divisions can you see?

The main subdivisions on 3 clusters are Muscle+Small muscle, Brain and Digestive+Heart, which we can observe that also get separated when we increase the number of clusters in later plots

All states convey similar information?

The further away 2 samples are, the more epigenomic differences they present.