

HiC Compartment analysis

Script name: HiC_Compartments.R

Purpose: Load the HiC compartment data, clean it and extract descriptive information

Email: monica.cabrera.pasadas@gmail.com

```
Compartments <- read.table(paste0(wd,"HCT116_HiC_compartments.tsv"), sep="\t", header=T)
Compartments <- Compartments[,-c(1,6:7)] # ***Specific step from my data!*** Deleting the column with
Compartments[is.na(Compartments)] <- 0 #To not delete all rows with one NA at some time point, I first
Compartments <- Compartments[rowSums(Compartments[,-c(1:3)])!=0,] # I remove all rows that have 0 (NA)

samples <- colnames(Compartments[,-c(1:3)])

## Let's categorize the compartments values in A and B according to their sign (A compartment has posit
for (i in samples) { Compartments[[paste0("Category_",i)]] <- as.character(ifelse(Compartments[i] < 0,
Compartments$combinations <- do.call(paste, c(Compartments[,grep("Category", names(Compartments)), value
Compartments$state_compartments <- as.character(ifelse(Compartments$combinations == paste(replicate(len
head(Compartments,5)
```

##	Chromosome	Start	End	WT.DMSO.0h	WT.NUT.1h	WT.NUT.4h	WT.NUT.7h
## 7	1	600001	700000	0.0000000	0.0000000	0.4791029	0.5000356
## 8	1	700001	800000	0.4077818	0.3850865	0.3949688	0.3968340
## 9	1	800001	900000	0.3519558	0.2824241	0.2965231	0.2996754
## 10	1	900001	1000000	0.0000000	0.2722165	0.2784057	0.2976727
## 11	1	1000001	1100000	0.2985673	0.3988010	0.3221267	0.3815691
##	WT.NUT.10h	WT.NUT.24h	Category_WT.DMSO.0h	Category_WT.NUT.1h			
## 7	0.0000000	0		NULL		NULL	
## 8	0.5078724	0		A		A	
## 9	0.3593087	0		A		A	
## 10	0.0000000	0		NULL		A	
## 11	0.0000000	0		A		A	
##	Category_WT.NUT.4h	Category_WT.NUT.7h	Category_WT.NUT.10h				
## 7		A		A		NULL	
## 8		A		A		A	
## 9		A		A		A	
## 10		A		A		NULL	
## 11		A		A		NULL	
##	Category_WT.NUT.24h		combinations	state_compartments			
## 7		NULL	NULL-NULL-A-A-NULL-NULL	dynamic			
## 8		NULL	A-A-A-A-A-NULL	dynamic			
## 9		NULL	A-A-A-A-A-NULL	dynamic			
## 10		NULL	NULL-A-A-A-NULL-NULL	dynamic			
## 11		NULL	A-A-A-A-NULL-NULL	dynamic			

1) How many compartments do we have for the analysis?

```
print(paste0("There are ", nrow(Compartments), " compartments with information at some time-point"))

## [1] "There are 28120 compartments with information at some time-point"

print(paste0("There are ", nrow(Compartments[- grep("NULL", Compartments$combinations),]), " compartments with information at all time-points"))

## [1] "There are 27245 compartments with information at all time-points"
```

2) How's the A/B compartment distribution?

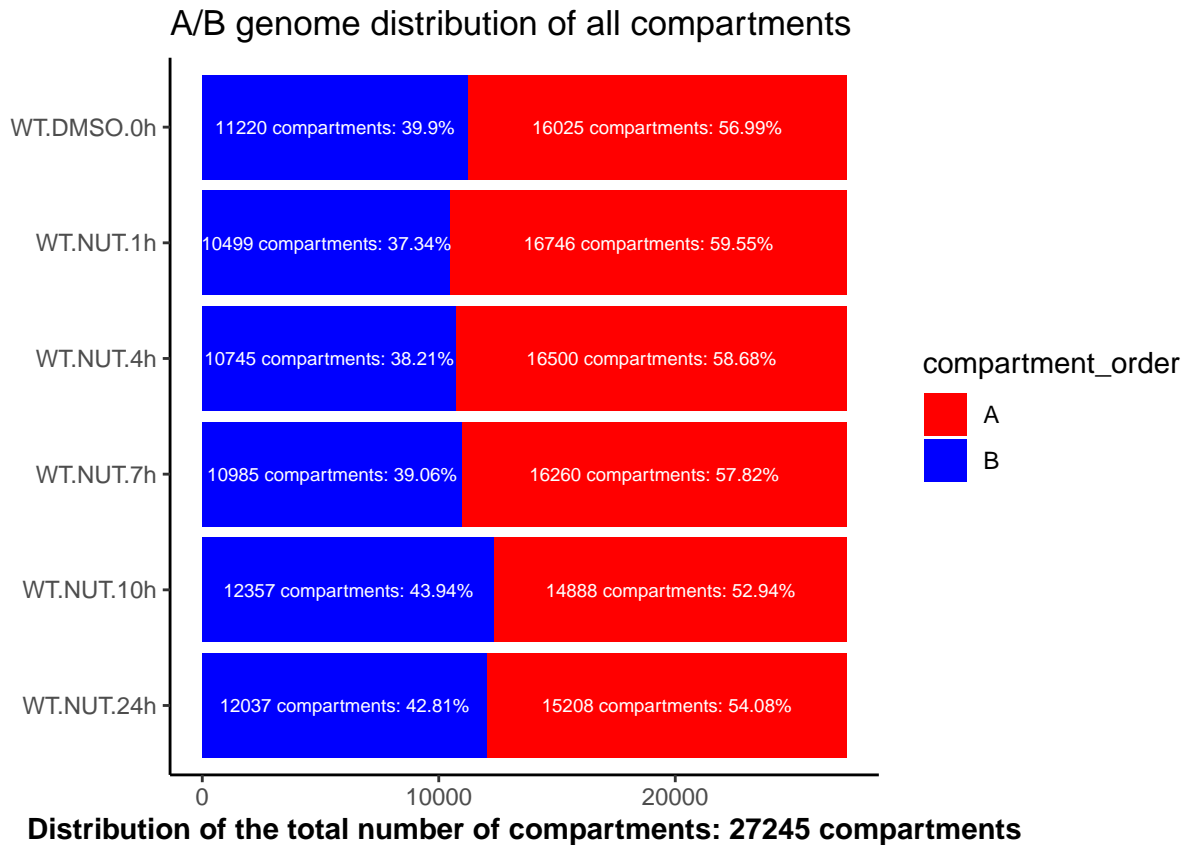
```
table_AB <- data.frame(apply(Compartments[- grep("NULL", Compartments$combinations),], c(1:length(samples)), FUN=function(x) {
  rownames(table_AB) <- c("B", "A")
  print(table_AB)
}))
```

```
##      WT.DMSO.0h WT.NUT.1h WT.NUT.4h WT.NUT.7h WT.NUT.10h WT.NUT.24h
## B          11220          10499          10745          10985          12357          12037
## A          16025          16746          16500          16260          14888          15208
```

```
# Fig 1.AB_distribution.pdf ----
# png(file="Fig1.HiC_Compartments_ABdistribution.png", width = 12, height = 8)
table_AB$compartment <- rownames(table_AB)
table_AB_melt <- melt(table_AB, id="compartment")

compartment_order <- factor(table_AB_melt$compartment, levels=c("A", "B"))

### Horizontal ----
sample_order_horizontal <- factor(table_AB_melt$variable, levels=c(rev(samples)))
print(ggplot(data = table_AB_melt, aes(x = sample_order_horizontal, y = value, fill = compartment_order)) +
  geom_col() +
  geom_text(aes(label = paste0(value, " compartments: ", round(value/nrow(Compartments)*100, digits=1)),
    y = value, fill = compartment_order)) +
  scale_fill_manual(values=c("red", "blue")) +
  # scale_y_continuous(breaks = c(0, 5000, 10000, 15000, 20000, 26973), expand = c(0, 0)) +
  theme_classic() +
  xlab("") +
  ylab(paste0("Distribution of the total number of compartments: ", nrow(Compartments[- grep("NULL", Compartments$combinations),]))) +
  ggtitle("A/B genome distribution of all compartments") +
  theme(axis.text.y=element_text(size=rel(1), angle=0),
    axis.text.x=element_text(size=rel(1)),
    axis.title=element_text(size=rel(1), face="bold")) +
  # geom_hline(yintercept=table_AB$WT.DMSO.0h[1], color="white") +
  coord_flip()
)
```

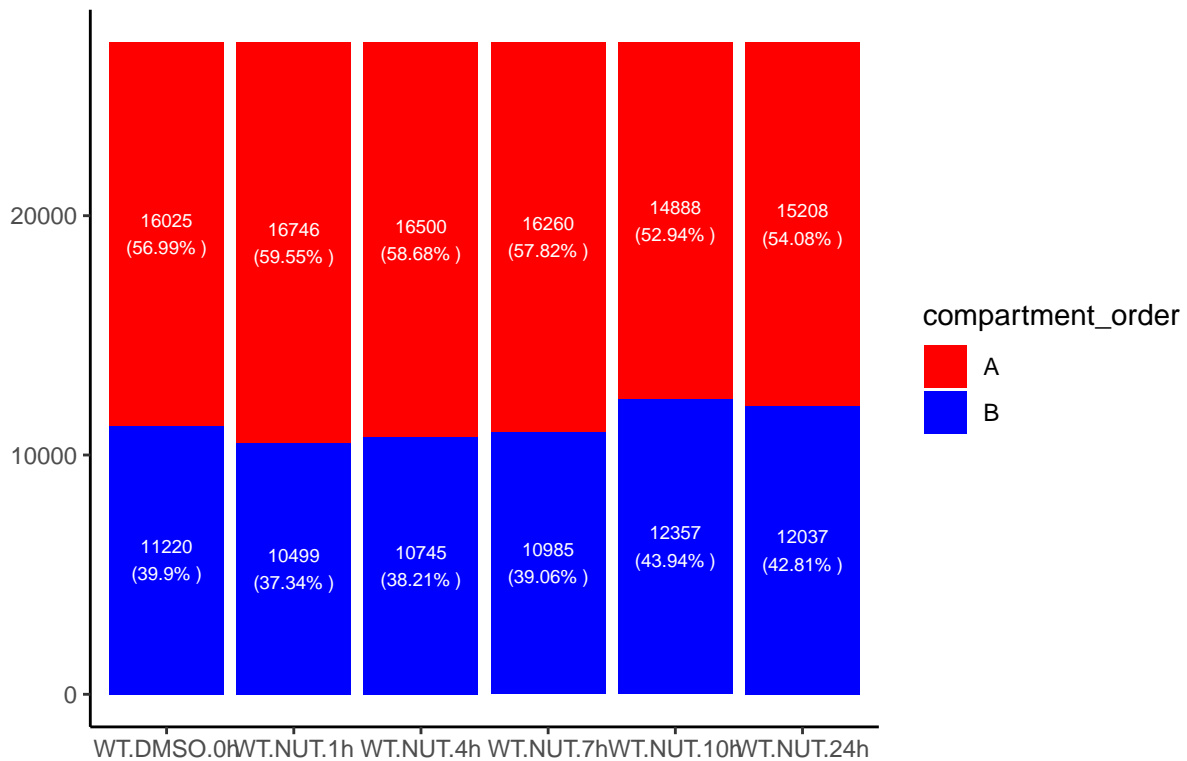


```

### Vertical ----
sample_order_vertical <- factor(table_AB_melt$variable, levels=c(samples))
print(ggplot(data = table_AB_melt, aes(x = sample_order_vertical, y = value, fill = compartment_order))
      geom_col() +
      geom_text(aes(label = paste0(value, "\n", "(", round(value/nrow(Compartments)*100, digits = 2), "% )"),
                    y = value, dx = 0, dy = -10, color = "white")) +
      scale_fill_manual(values=c("red", "blue")) +
      # scale_y_continuous(breaks = c(0, 5000, 10000, 15000, 20000, 25000, 26973), expand = c(0, 0)) +
      theme_classic() +
      ylab("") +
      xlab("") +
      ggtitle(paste0("A/B genome distribution of the total number of compartments: ", nrow(Compartments)))
      #+geom_hline(yintercept=table_AB$WT.DMSO[1], color="white"))
)

```

A/B genome distribution of the total number of compartments: 27245 cor



```
# dev.off()
```

3) How many compartments are static and how many compartments change residency?

```
static_dynamic_compartments <- data.frame(sort(table(Compartments[- grep("NULL", Compartments$combination)],
colnames(static_dynamic_compartments) <- c("Category", "Total_number")
print(static_dynamic_compartments)
```

```
##      Category Total_number
## 1 static in A      14078
## 2 static in B      9772
## 3      dynamic      3395
```

```
# Fig 2.Static_dynamic_distribution.pdf ----
# png(file="Fig2.HiC_Compartments_static_dynamic_distribuion.png",width = 12, height = 8)
static_dynamic_compartments$Category <- as.character(static_dynamic_compartments$Category)
static_dynamic_compartments[nrow(static_dynamic_compartments) + 1,] = c('static', sum(static_dynamic_compartments$Total_number))
static_dynamic_compartments$Category <- as.factor(static_dynamic_compartments$Category)
static_dynamic_compartments$Total_number <- as.integer(static_dynamic_compartments$Total_number)

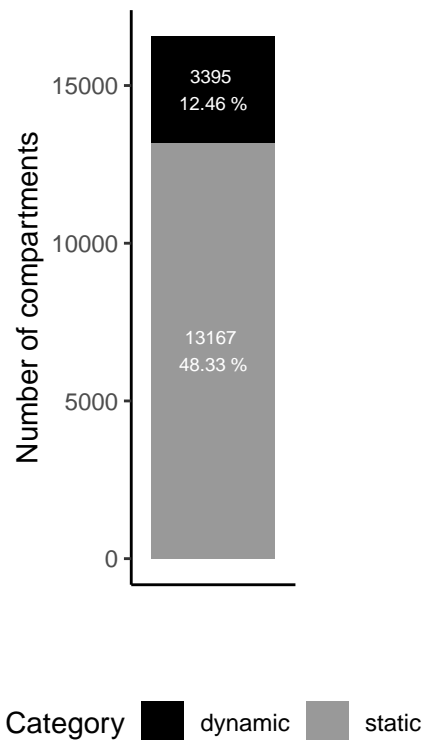
print(ggplot(static_dynamic_compartments[c(3,4),], aes(x="", y = Total_number, fill=Category)) +
  geom_bar(position="stack",stat="identity",width = 0.9) +
```

```

scale_fill_manual(values=c("black","grey60", "grey80"))+
# labs(title = "Genomic distribution of the compartments", subtitle = "Divided by stable and sw
xlab("")+
ylab("Number of compartments")+
ggtitle(paste0("Out of the total ",nrow(Compartments[- grep("NULL", Compartments$combinations),
theme_ipsum() +
theme_classic()+
theme(legend.position="bottom")+
theme(axis.ticks.x=element_blank())+
# scale_y_continuous(breaks = c(0,5000,10000, 15000,20000,23500,26973), expand = c(0,0))+
theme(aspect.ratio = 3.5)+
geom_text(aes(label = paste0(static_dynamic_compartments[c(3,4),]$Total_number," \n",round((sta
)

```

Out of the total 27245 compartments:



```

order_dynamism <- factor(static_dynamic_compartments[1:3,]$Category, levels=c("dynamic", "static in A",
print(ggplot(static_dynamic_compartments[1:3,], aes(x="", y = Total_number, fill=order_dynamism)) +
  geom_bar(position="stack",stat="identity",width = 0.9 )+
  scale_fill_manual(values=c("black","grey60", "grey80"))+
# labs(title = "Genomic distribution of the compartments", subtitle = "Divided by stable and s
xlab("")+
ylab("Number of compartments")+
ggtitle(paste0("Out of the total ",nrow(Compartments[- grep("NULL", Compartments$combinations),
theme_ipsum() +
theme_classic()+

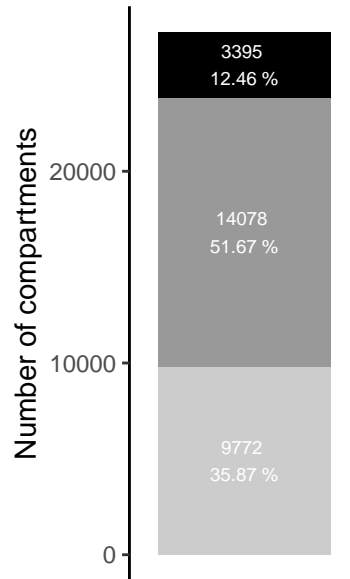
```

```

theme(legend.position="bottom")+
theme(axis.ticks.x=element_blank())+
# scale_y_continuous(breaks = c(0,5000,10000, 15000,20000,25000,26973), expand = c(0,0))+
theme(aspect.ratio = 2.5)+
geom_text(aes(label = paste0(Total_number, " \n", round((Total_number/nrow(Compartments[- grep("N
)

```

Out of the total 27245 compartments with va



order_dynamism dynamic static in A static in B

```
# dev.off()
```

```

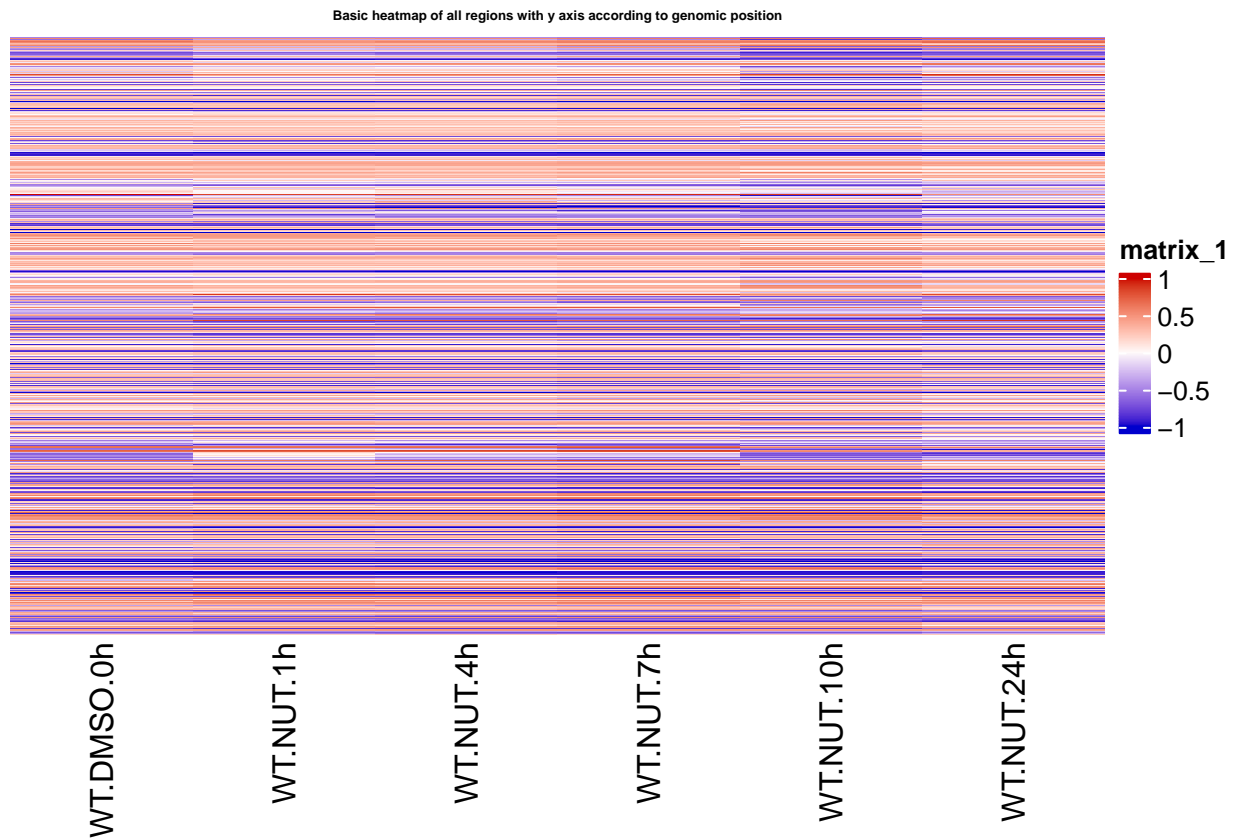
# Fig 3. Heatmap of all compartments----
# Information of the Heatmap options: https://www.quantargo.com/help/r/latest/packages/ComplexHeatmap/D
# png(file="Fig4.Heatmap.png",width = 12, height = 8)
mycols <- colorRamp2(breaks = c(-1,0,1), colors = c("blue3", "snow1","red3"))

```

```

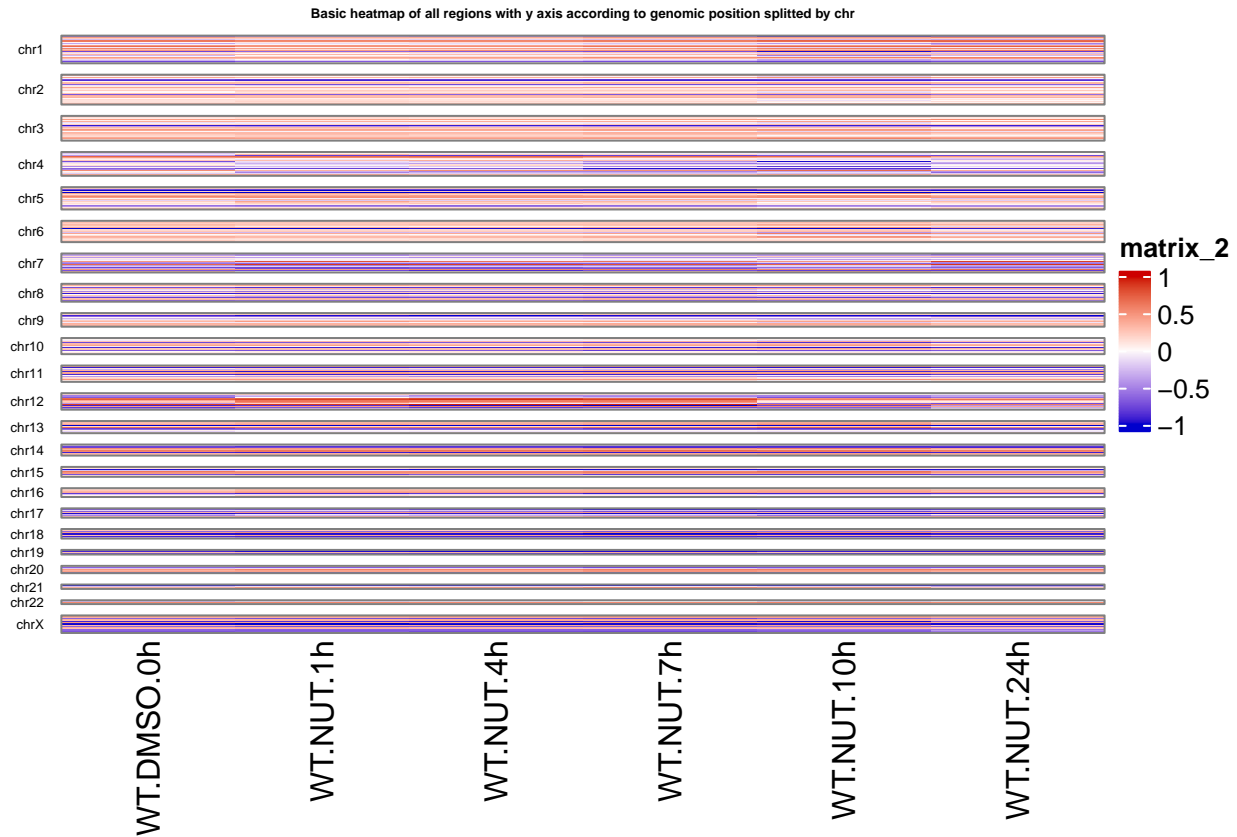
### Basic heatmap of all compartments with y axis according to genomic position ----
Heatmap(as.matrix(Compartments[!grepl("NULL",Compartments$combinations),][,c(1:length(samples)+3)]),col

```



Basic heatmap of all compartments with y axis according to genomic position ----

```
Heatmap(as.matrix(Compartments[!grepl("NULL",Compartments$combinations),][,c(1:length(samples)+3)]),col
```



4) How is the compartment change residency happening in the dynamic regions?

```
compartments_Freq <- data.frame(sort(table(Compartments$combinations),decreasing = T)) #First we check
compartments_Freq$order_combinations <- seq(1:nrow(compartments_Freq)) # we add the frequency rank as I

Compartments$Frequency_combinations = compartments_Freq$Freq[match(Compartments$combinations,compartments_Freq$order_combinations)]
Compartments$order_combinations = compartments_Freq$order_combinations[match(Compartments$combinations,compartments_Freq$order_combinations)]

dynamic_compartments_Freq <- data.frame(sort(table(Compartments[! grepl("NULL", Compartments$combinations)],decreasing = T))
print(dynamic_compartments_Freq)
```

```
##          Var1 Freq
## 1  A-A-A-A-B-A 527
## 2  A-A-A-A-A-B 428
## 3  A-A-A-A-B-B 386
## 4  B-A-A-A-B-B 332
## 5  B-A-B-B-B-B 307
## 6  B-A-A-B-B-B 149
## 7  B-B-B-B-B-A 118
## 8  B-A-A-A-B-A 117
## 9  A-B-B-B-B-B 108
## 10 A-B-A-B-B-B  88
## 11 B-A-A-A-A-A  60
```



```

## 12 B-A-B-A-B-B 57
## 13 A-A-A-B-B-B 51
## 14 B-B-B-B-A-B 51
## 15 B-B-A-B-B-B 47
## 16 A-B-A-A-B-B 37
## 17 B-A-B-B-B-A 33
## 18 B-A-A-B-B-A 32
## 19 A-B-B-B-A-B 31
## 20 A-A-B-A-B-B 26
## 21 A-B-A-A-A-A 25
## 22 A-A-B-B-B-B 24
## 23 B-B-B-A-B-B 23
## 24 B-B-B-B-A-A 21
## 25 A-B-B-B-B-A 20
## 26 A-B-B-B-A-A 17
## 27 A-A-B-A-A-A 16
## 28 A-B-B-A-B-B 16
## 29 A-A-A-B-A-B 15
## 30 A-A-B-A-A-B 15
## 31 A-B-A-A-A-B 14
## 32 A-B-A-B-A-A 14
## 33 A-B-B-A-A-A 13
## 34 A-A-A-B-B-A 12
## 35 B-A-B-A-B-A 12
## 36 B-B-A-A-B-B 12
## 37 A-A-A-B-A-A 11
## 38 B-A-A-A-A-B 10
## 39 B-B-A-B-B-A 9
## 40 B-B-B-A-B-A 9
## 41 A-A-B-B-A-B 8
## 42 A-B-A-A-B-A 8
## 43 A-B-A-B-A-B 8
## 44 B-A-A-B-A-A 7
## 45 A-B-A-B-B-A 6
## 46 B-A-B-A-A-A 6
## 47 A-A-B-A-B-A 5
## 48 A-A-B-B-B-A 5
## 49 A-B-B-A-A-B 5
## 50 B-A-B-B-A-A 5
## 51 B-A-B-B-A-B 5
## 52 B-B-A-A-B-A 5
## 53 B-B-A-B-A-B 5
## 54 A-A-B-B-A-A 4
## 55 A-B-B-A-B-A 4
## 56 B-B-B-A-A-A 4
## 57 B-A-B-A-A-B 3
## 58 B-B-A-A-A-A 3
## 59 B-B-A-A-A-B 2
## 60 B-B-A-B-A-A 2
## 61 B-B-B-A-A-B 2

```

```

# Fig 4.Dynamic_compartment_distribution.pdf ----
# png(file="Fig3.HiC_Compartments_dynamic_compartments_distribuition.png",width = 12, height = 8)
dynamic_compartments_Freq$Percentage <- round(dynamic_compartments_Freq$Freq/nrow(Compartments[(! grepl

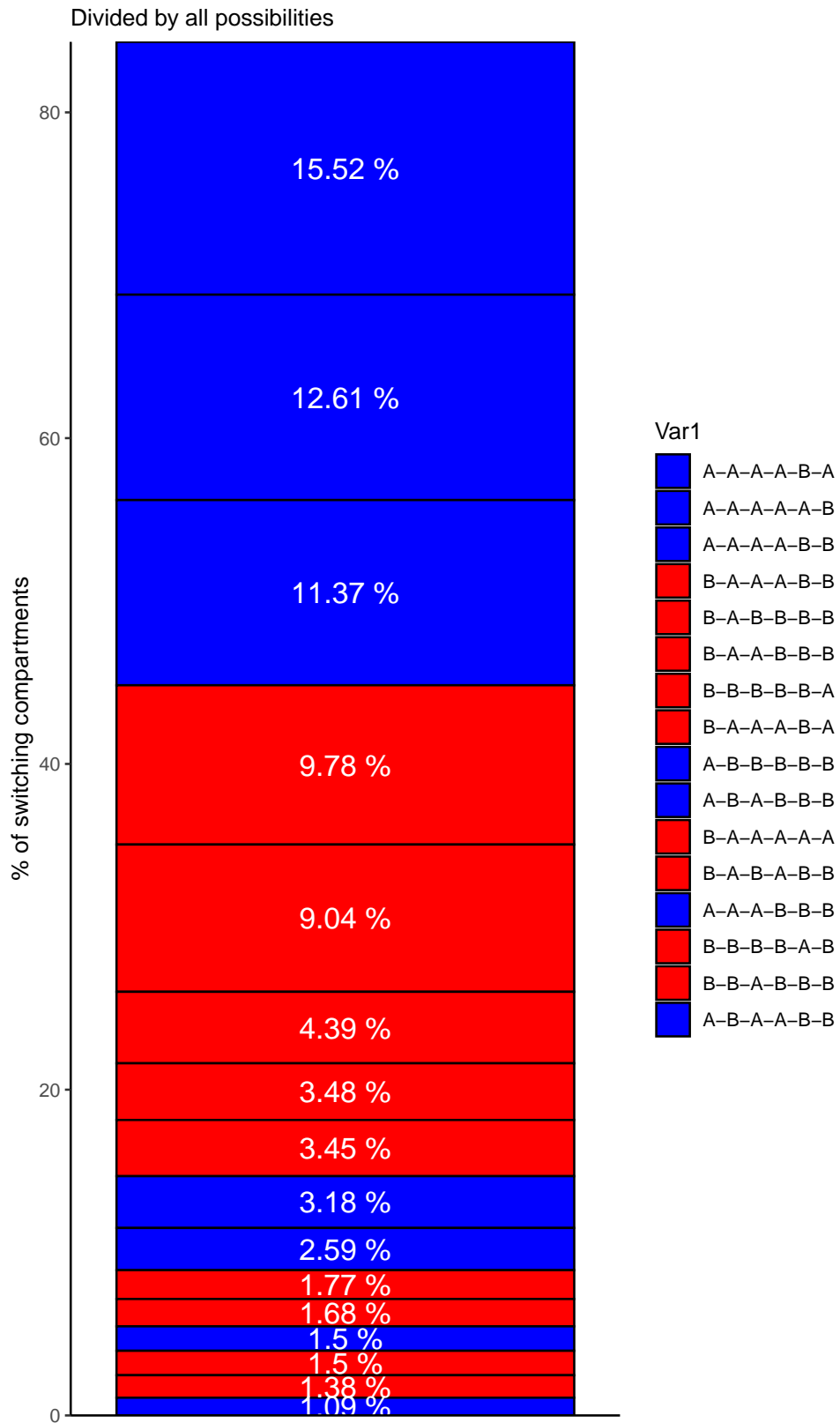
```

```

dynamic_compartments_Freq <- dynamic_compartments_Freq %>% mutate(colors = case_when(str_detect(Var1, "
dynamic_compartments_Freq_1 <- dynamic_compartments_Freq[dynamic_compartments_Freq$Percentage >= 1,]

print(ggplot(dynamic_compartments_Freq_1, aes(x="",y=Percentage,fill=Var1)) +
  geom_bar(position="stack",stat="identity",width = 1 ,color="black") +
  scale_fill_manual(values = dynamic_compartments_Freq_1$colors) +
  labs(title = "Switching compartments distribution of all compartments",subtitle = "Divided by a
  xlab("")+
  ylab("% of switching compartments")+
  theme_ipsum() +
  theme_classic()+
  theme(axis.ticks.x=element_blank())+
  theme(aspect.ratio = 2.5)+
  scale_y_continuous(expand = c(0,0))+
  geom_text(aes(label = paste0(Percentage," %")),position = position_stack(vjust = .5), color="wh
  ggtitle("")
)

```

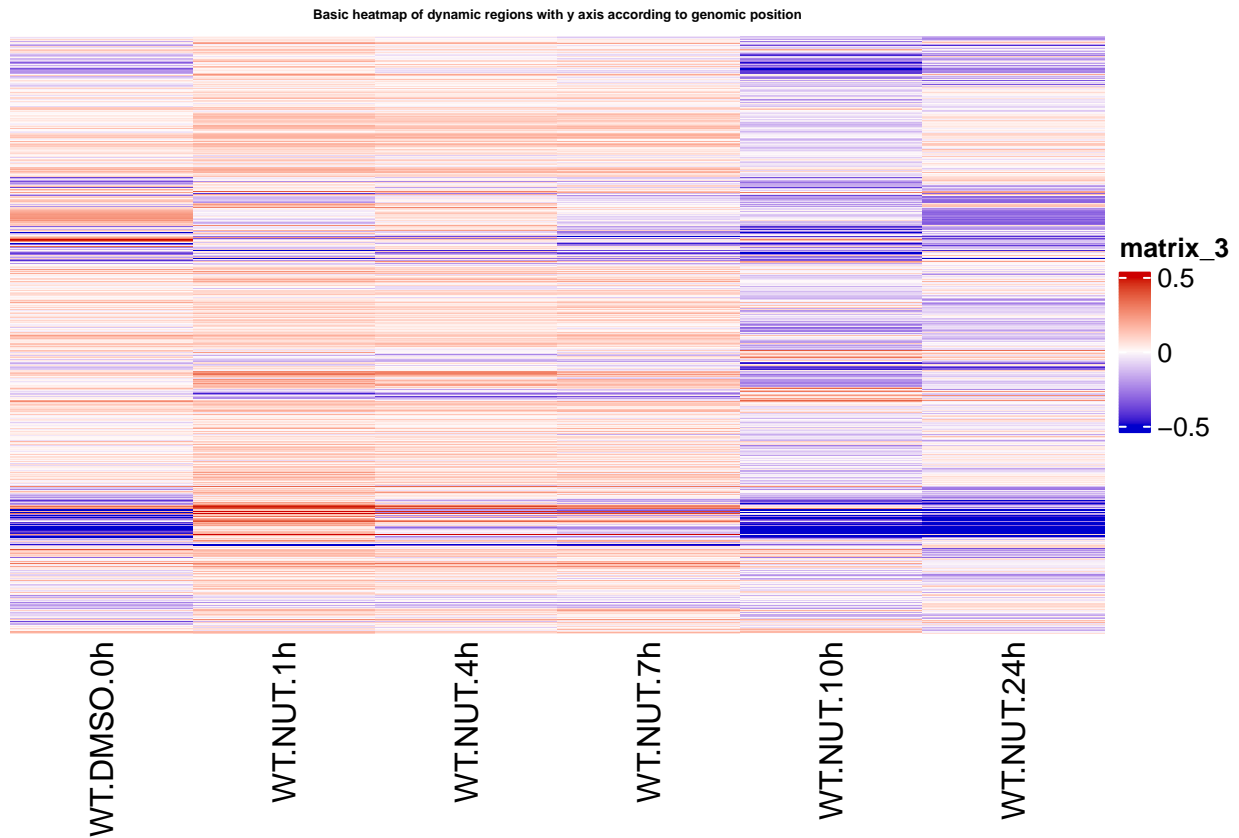


```
# dev.off()
```

```
mycols_dynamic <- colorRamp2(breaks = c(-0.5,0,0.5), colors = c("blue3", "snow1","red3"))
```

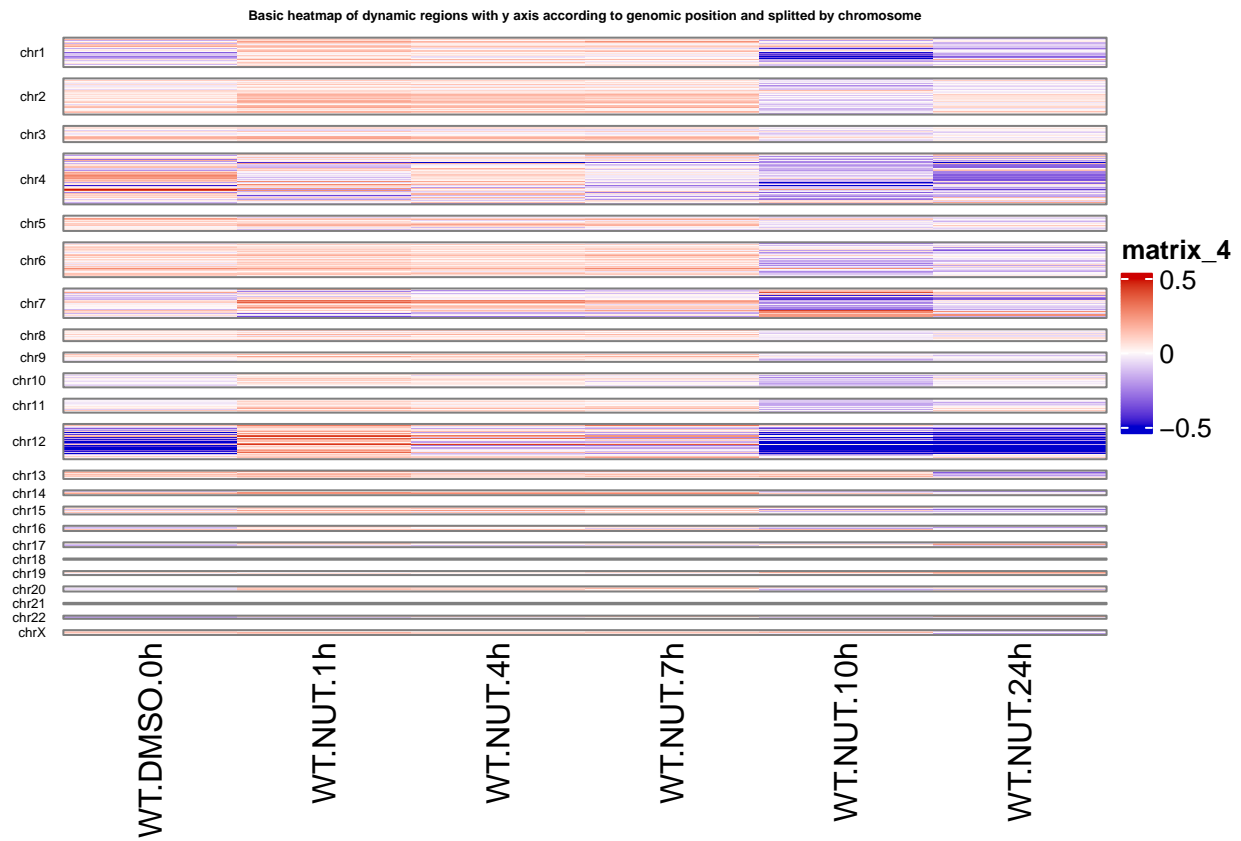
```
### Basic heatmap of only dynamic regions with y axis according to genomic position
```

```
Heatmap(as.matrix(Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
```



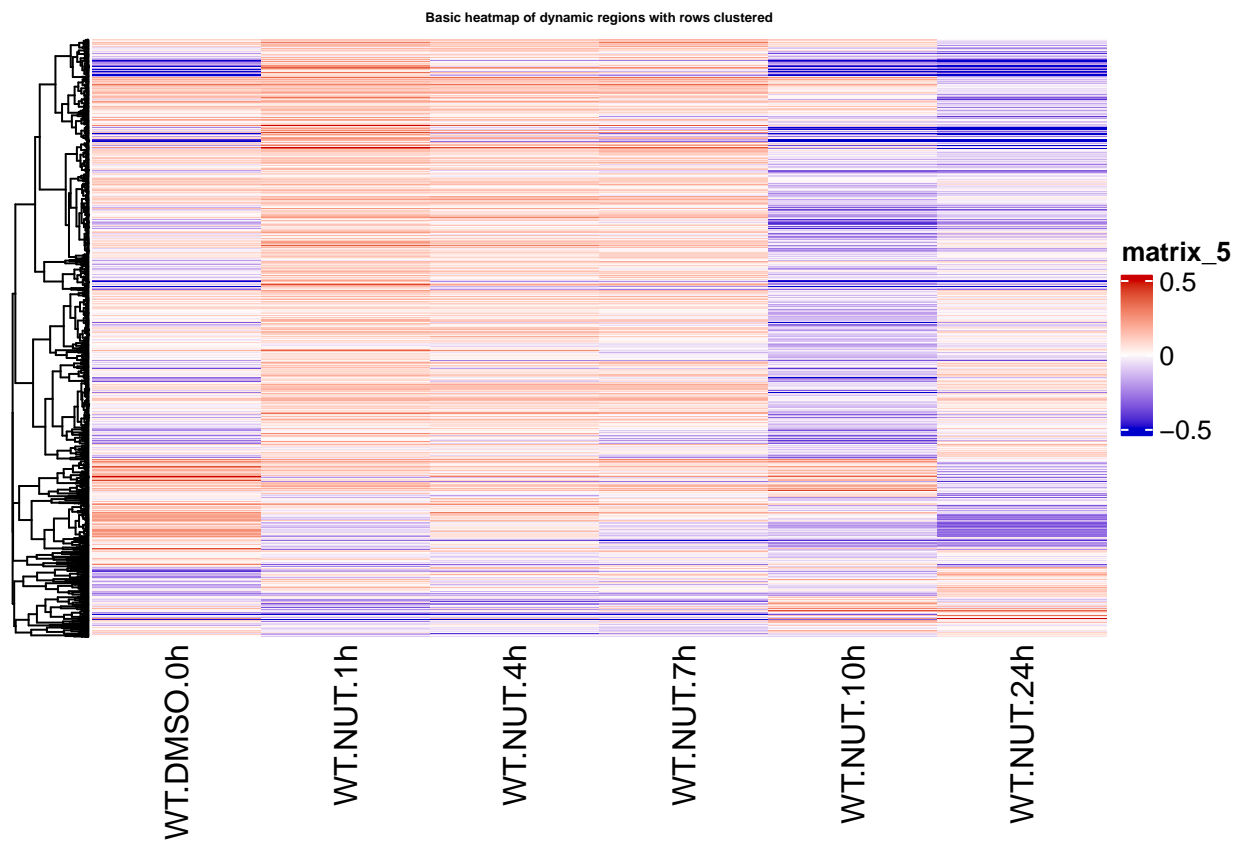
```
### Basic heatmap of only dynamic regions with rows splited by chromosome
```

```
Heatmap(as.matrix(Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
```



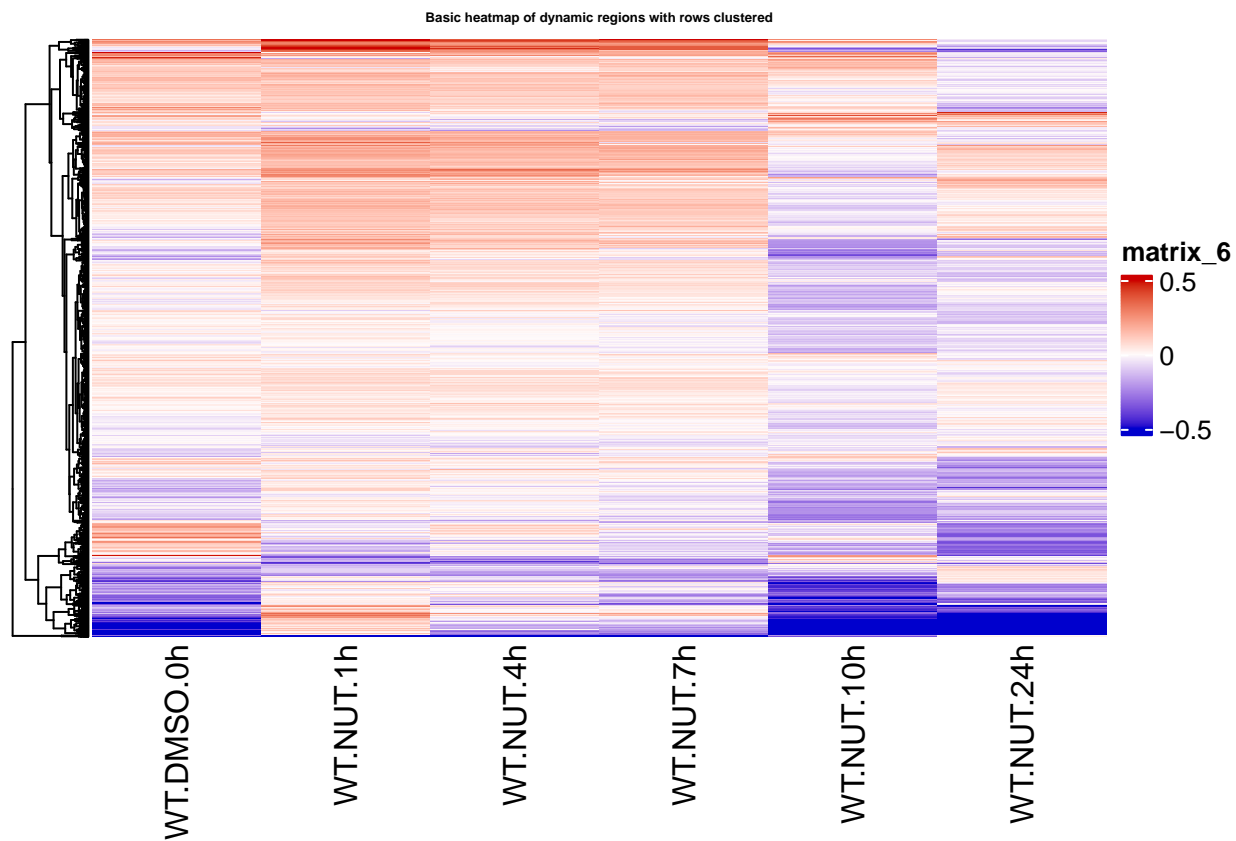
Basic heatmap of dynamic regions with rows clustered

```
Heatmap(as.matrix(Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
```

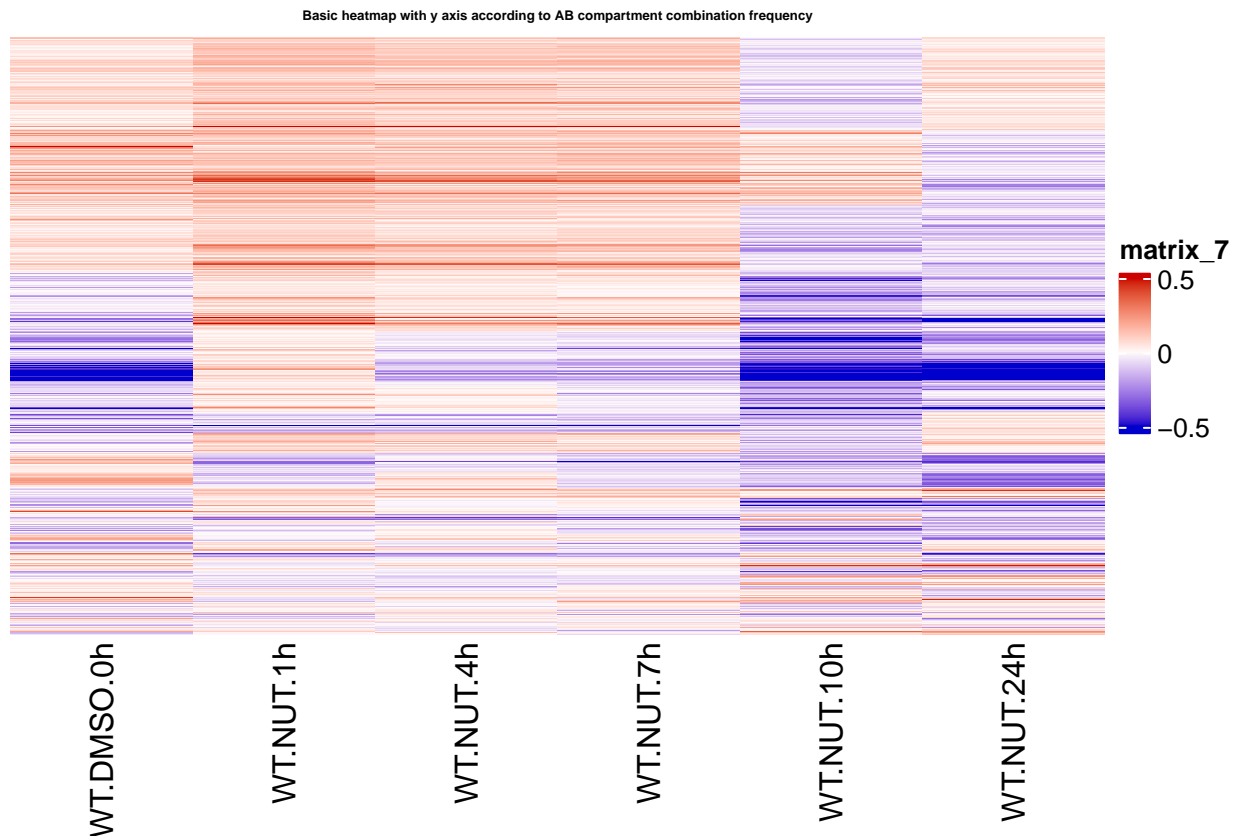


```
### Basic heatmap of dynamic regions with rows clustered
```

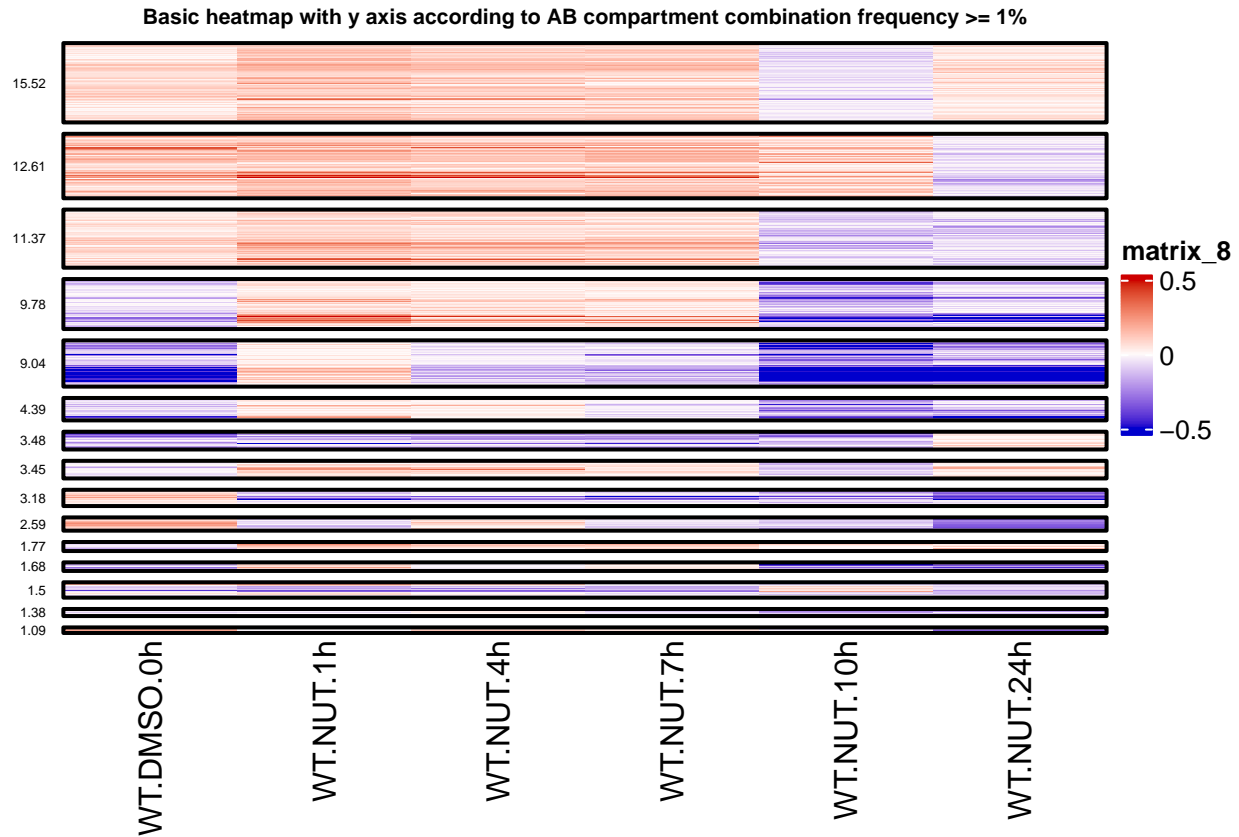
```
Heatmap(as.matrix(Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
```



```
### Basic heatmap of dynamic regions with y axis according to AB compartment combination frequency
dynamics_Freq_ordered <- Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
Heatmap(as.matrix(dynamics_Freq_ordered[,c(1:length(samples)+3)]),column_title = "Basic heatmap with y axis according to AB compartment combination frequency")
```



```
### Basic heatmap of dynamic regions with y axis according to AB compartment combination frequency >= 1
dynamics_Freq_ordered <- Compartments[Compartments$state_compartments=="dynamic" & !grepl("NULL",Compartments$
dynamics_Freq_ordered$Percentage_Freq = dynamic_compartments_Freq$Pe[match(dynamics_Freq_ordered$combination,
Heatmap(as.matrix(dynamics_Freq_ordered[dynamics_Freq_ordered$Percentage_Freq >= 1,]),c(1:length(samples)),
cluster_rows = F, show_row_names = F,row_split=factor(dynamics_Freq_ordered[dynamics_Freq_ordered$Percentage_Freq >= 1,]),
row_title_rot = 0,row_title_gp = gpar(col = c("black"),fontsize=5),row_gap = unit(1.5, "mm")
,border = TRUE,border_gp = gpar(col = "black", lwd = 2),
cluster_columns=F,col = mycols_dynamic)
```

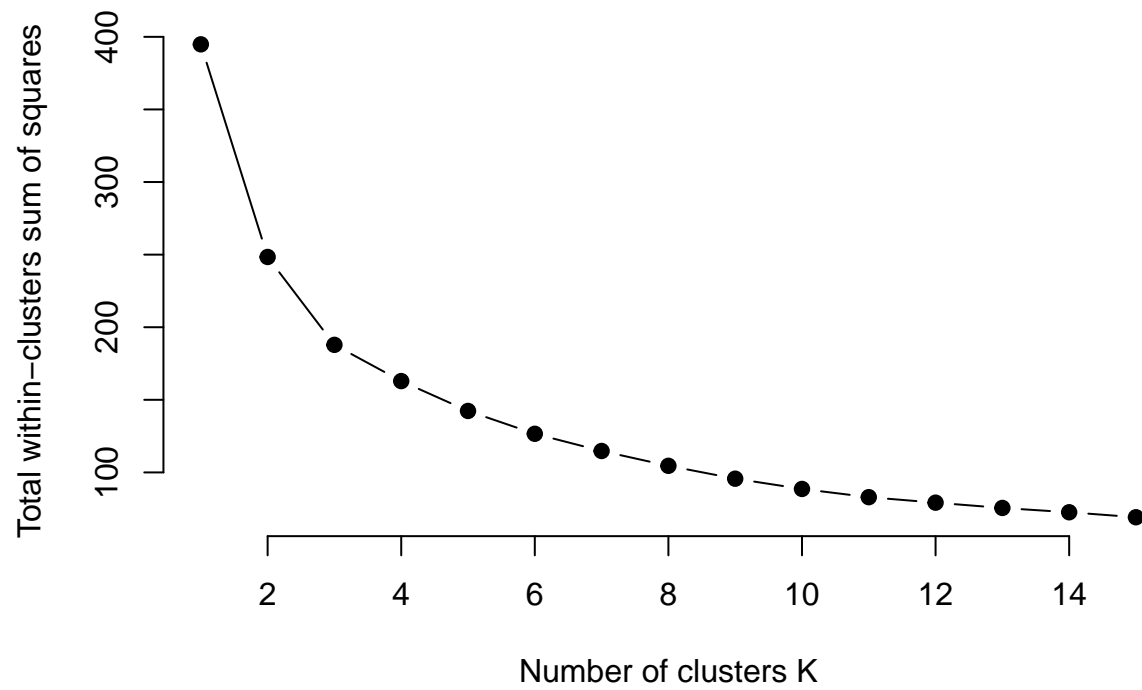



```
### Heatmap of dynamic regions switch frequency >=1 % clustered by kmeans ----

#Elbow Method for finding the optimal number of clusters
set.seed(103)
# Compute and plot wss for k = 2 to k = 15.
k.max <- 15
data <- dynamics_Freq_ordered[dynamics_Freq_ordered$Percentage_Freq >= 1,][,c(1:length(samples)+3)]
wss <- sapply(1:k.max,
              function(k){kmeans(data, k, nstart=50, iter.max = 15 )$tot.withinss})
```

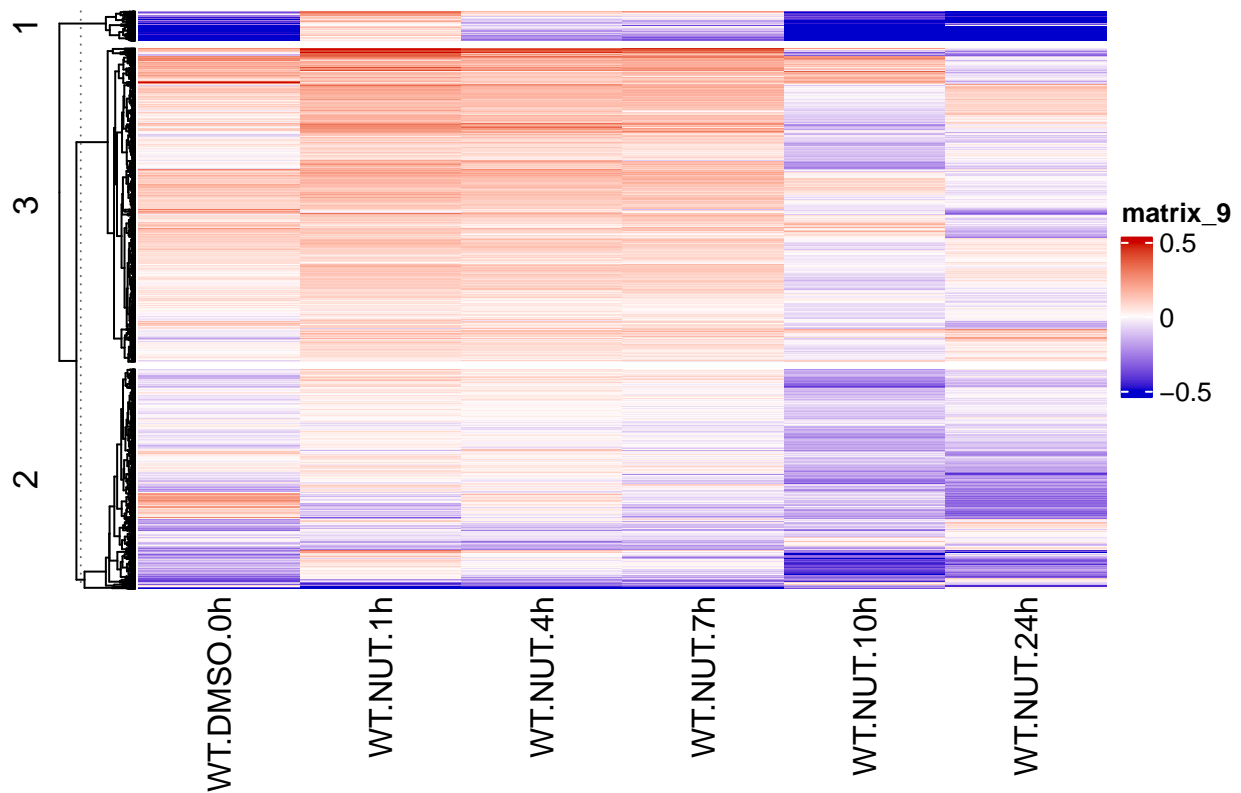
```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 143150)
```

```
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



```
Heatmap(as.matrix(data),column_title = "Heatmap of dynamic regions switch frequency >=1 % clustered by 1
```

Heatmap of dynamic regions switch frequency ≥ 1 % clustered by kmeans



#FALTA COLOREAR REGIONES AB compartments

Creating GRanges

`Compartments.gr <- makeGRangesFromDataFrame(Compartments, seqnames.field = "Chromosome", start.field = "start", end.field = "end")`

Saving the object in an RDS to avoid loading the information again if we want to use the compartments

`saveRDS(Compartments.gr, file = paste(wd, "Compartments.rds"))`

Saving all the workspace created in this session

`save.image(file = paste(wd, "HiC_Compartments.RData"))`