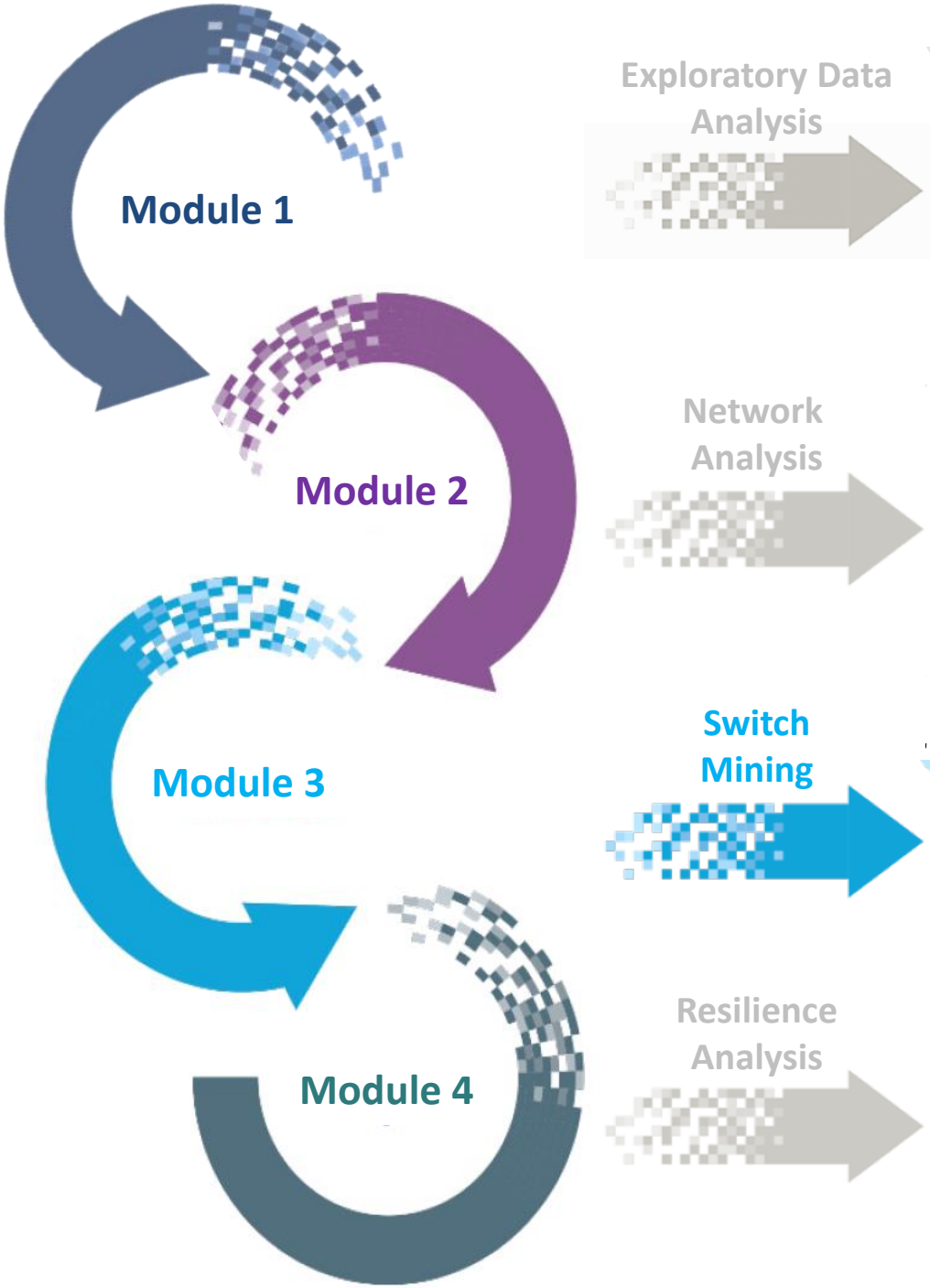
Module 3:
Switch Mining



Exploratory Data Analysis

Importing data

Pre-processing phase

Filtering phase

Network Analysis

Building the correlation network

Checking network integrity

Finding communities in the network

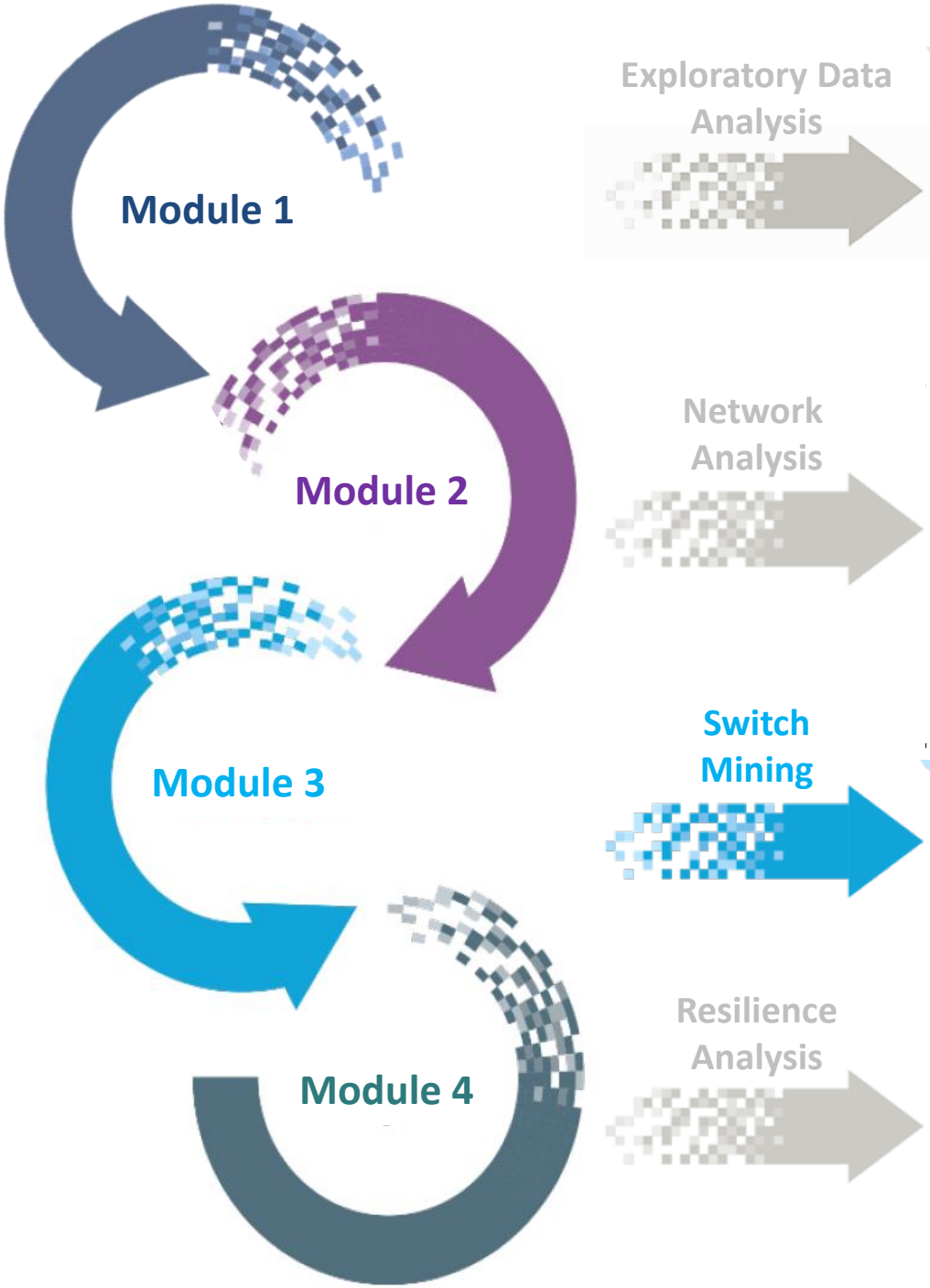
Switch Mining

Building the heat cartography map

Identification of switch genes

Resilience Analysis

Evaluation of network resilience



Exploratory Data Analysis

Importing data

Pre-processing phase

Filtering phase

Network Analysis

Building the correlation network

Checking network integrity

Finding communities in the network

Switch Mining

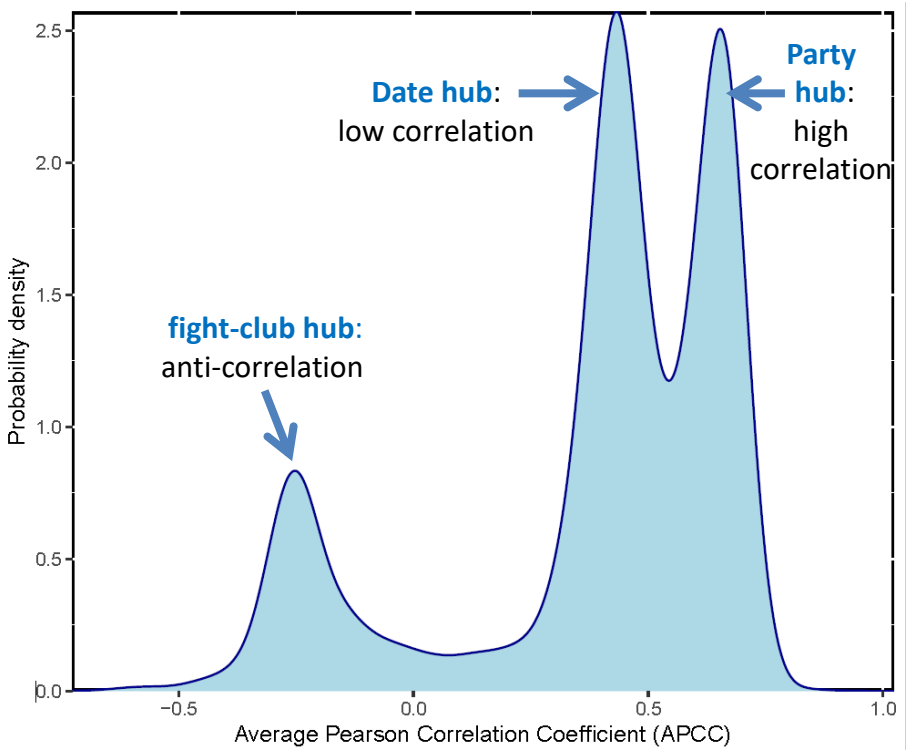
Building the heat cartography map

Identification of switch genes

Resilience Analysis

Evaluation of network resilience

APCC distribution

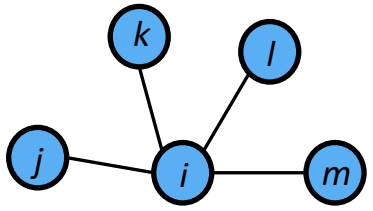


- x-axis refers to APCC values computed between the expression profiles of each hub of correlation network and its nearest neighbors
- y-axis refers to the probability density

Average Pearson Correlation Coefficient (APCC)

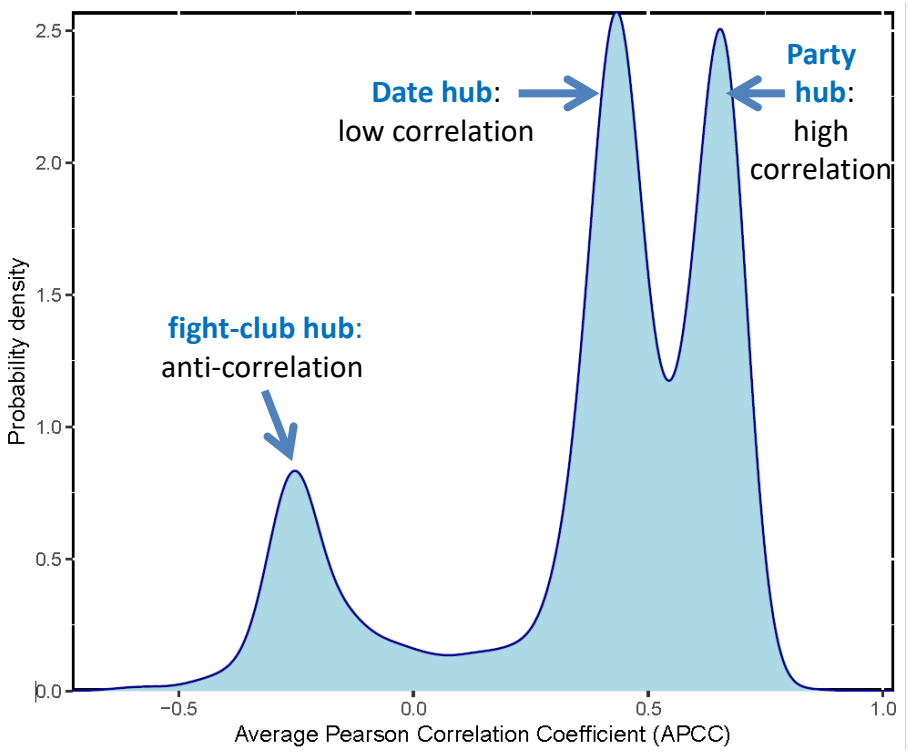
- An APCC value is assigned to each node of correlation network
- APCC measures how the nodes are co-expressed with their nearest neighbors

Example:



$$APCC_i = \frac{\rho(i,j)+\rho(i,k)+\rho(i,l)+\rho(i,m)}{4}$$

APCC distribution

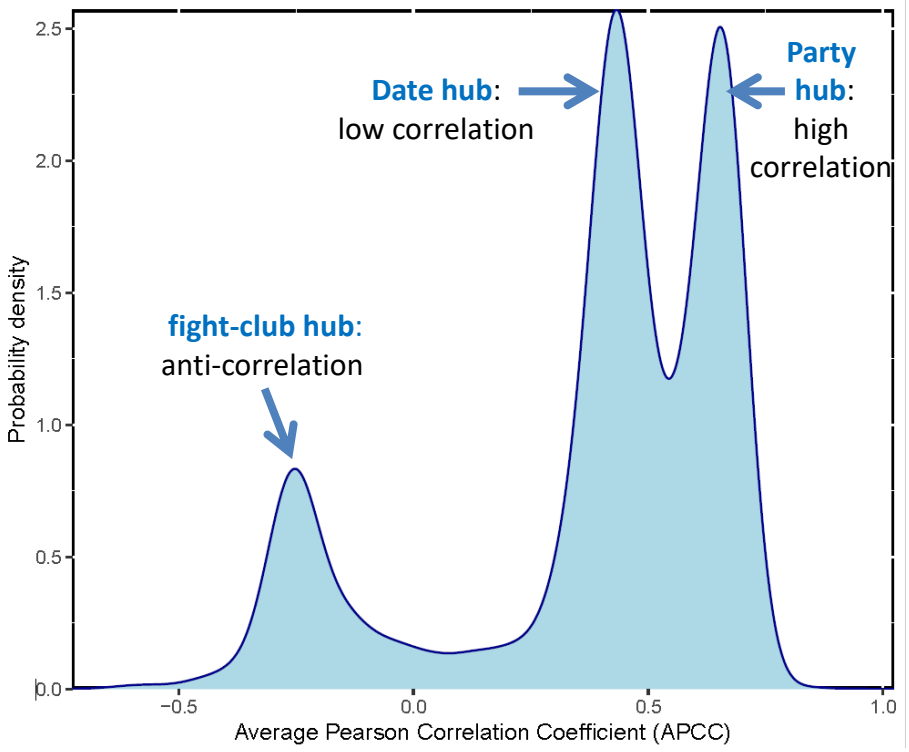


- x-axis refers to APCC values computed between the expression profiles of each hub of correlation network and its nearest neighbors
- y-axis refers to the probability density

Average Pearson Correlation Coefficient (APCC)

```
computeAPCC <- function(w_adj){
  w_adj[w_adj==0] <- NA
  APCC <- data.frame(APCC = rowMeans(w_adj,na.rm = T))
  return(APCC)
}
```

APCC distribution

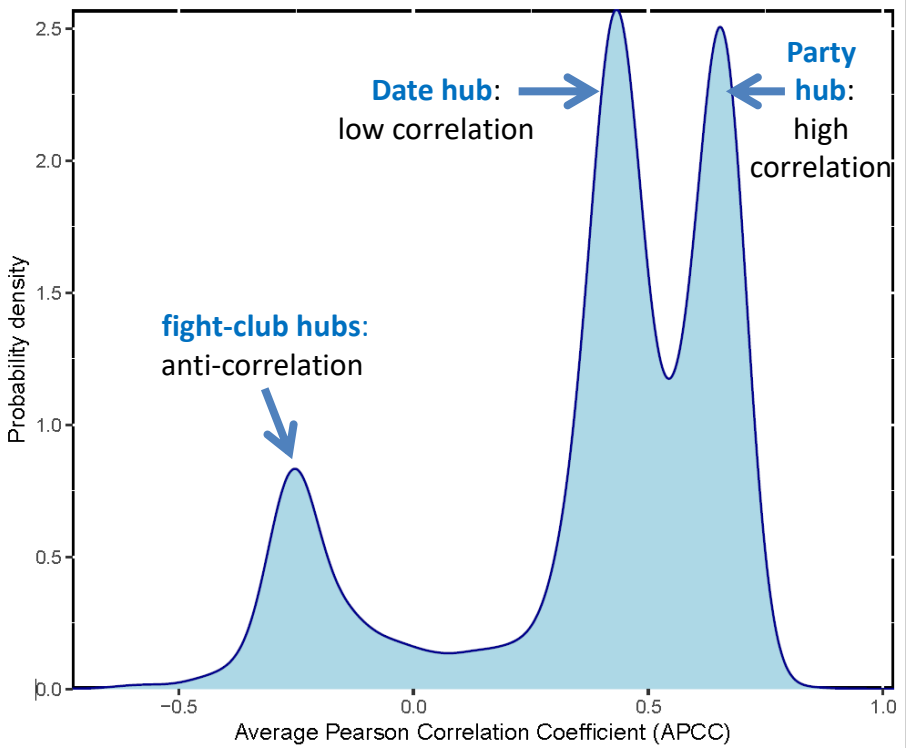


- x-axis refers to APCC values computed between the expression profiles of each hub of correlation network and its nearest neighbors
- y-axis refers to the probability density

APCC distribution

- The curve represents the estimated probability density using a smoothing algorithm with a Gaussian kernel of the APCC values' distribution
- It appears to be trimodal and the three peaks correspond to three different kind of hubs:
  1. **party hubs** which are highly correlated in expression with their interaction partners;
  2. **date hubs** which show moderate correlation in expression with their interaction partners;
  3. **fight-club hubs** which show an average negative correlation in expression with their interaction partners.

APCC distribution



- x-axis refers to APCC values computed between the expression profiles of each hub of correlation network and its nearest neighbors
- y-axis refers to the probability density

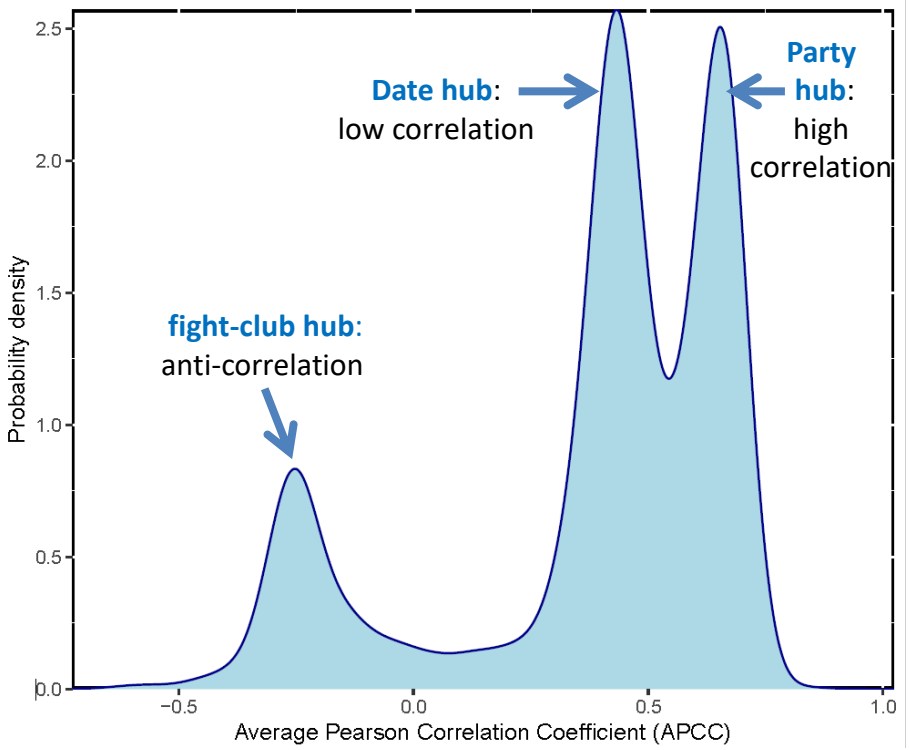
APCC distribution

```
getHubClassification <- function(APCC,deg){

  condition1 <- (APCC$APCC > 0) & (APCC$APCC < 0.5) & (deg$deg >= 5)
  condition2 <- (APCC$APCC >= 0.5) & (deg$deg >= 5)
  condition3 <- (APCC$APCC < 0) & (deg$deg >= 5)
  condition4 <- (deg$deg < 5)

  hub_class <- data.frame(hub = ifelse(condition1,"DATE",
                                       ifelse(condition2,"PARTY",
                                              ifelse(condition3,"FIGHT CLUB",
                                                     ifelse(condition4,"no hub","not
available")))))
  rownames(hub_class) <- rownames(deg)

  return(hub_class)

}
```

⚠ **Caveat:** hub is defined as a node with a degree greater than 5 [Han et al., Nature 2004]

⚠ **Caveat:** In order to optimally separate date and party hubs, a date/party threshold is arbitrarily set to 0.5 for the APCC [Han et al., Nature 2004].

APCC distribution



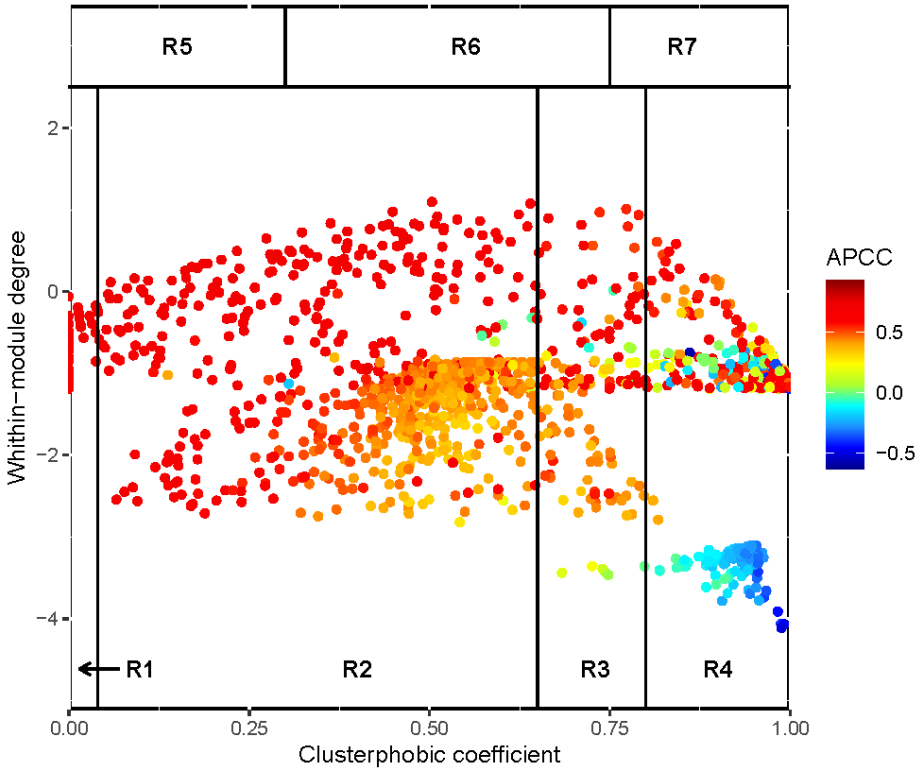
- x-axis refers to APCC values computed between the expression profiles of each hub of correlation network and its nearest neighbors
- y-axis refers to the probability density

APCC distribution

```
getAPCCdistribution <- function(attribute,output_file){

  m <- min(attribute$APCC) - 0.1
  M <- max(attribute$APCC) + 0.1

  ind <- which(attribute$Hub_classification == "no hub")
  if(length(ind) > 0) attribute <- attribute[-ind,]

  p <- ggplot(attribute, aes(x=APCC)) + geom_density(color="darkblue",
fill="lightblue") +
    scale_x_continuous(expand = c(0, 0), limits = c(m,M)) + scale_y_co
ntinuous(expand = c(0, 0)) +
    theme(panel.background = element_rect(fill = "white", colour =
"black", size = 1)) +
    labs(x = "Average Pearson Correlation Coefficient (APCC)", y =
"Probability density")

  print(p)

  savePDF(p,output_file)

}
```

Computes and draws kernel density estimate

Heat cartography map

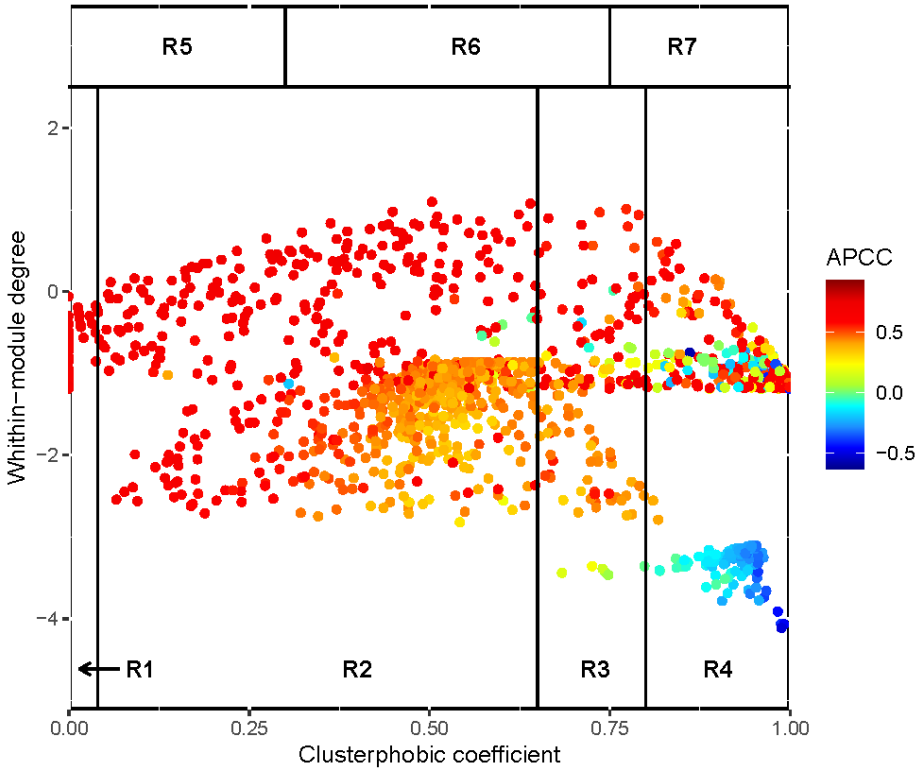


- x-axis refers to K_π (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

Heat cartography map

- The cartography is defined by two parameters: z_g (**within-module degree**) and K_π (**clusterphobic coefficient**) that divided the plane into seven regions, each defining a specific node role (R1-R7)
- The **heat cartography map** is computed by coloring each node of the correlation network according to its **APCC value**

Heat cartography map



- x-axis refers to K_π (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

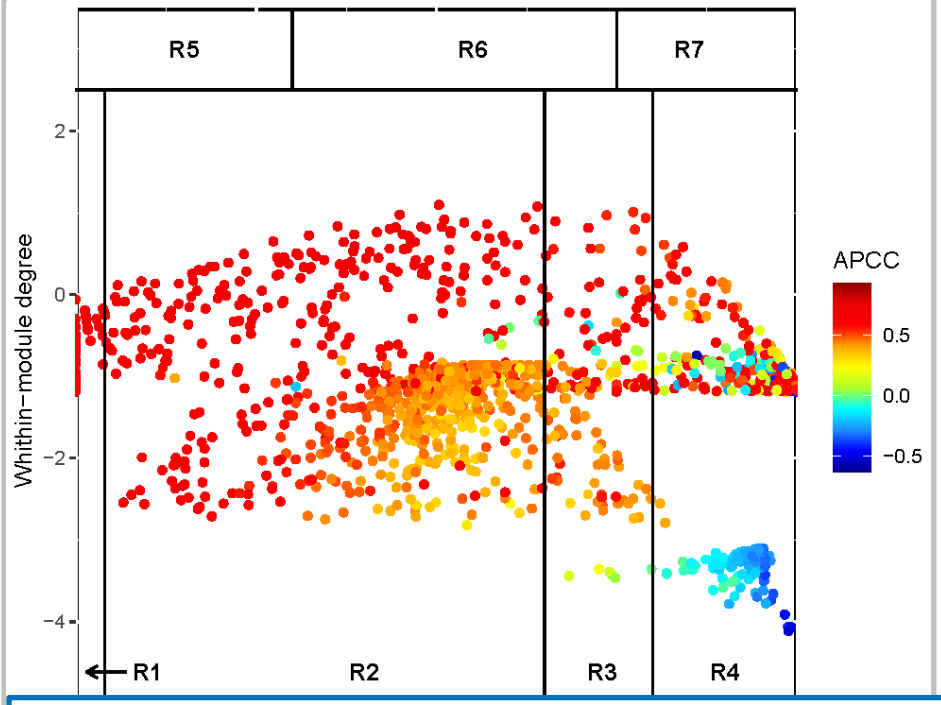
Clusterphobic coefficient (Kπ)

- K_π measures the "fear" of being confined in a cluster
- It evaluates the ratio of internal to external connections of a node and thus represents a measure of **global connectivity**
- A high value of K_π denotes nodes having much more external than internal links

$$K_\pi = 1 - \left(\frac{k_i^{in}}{k_i}\right)^2$$

- k_i^{in} is the number of links of node i to nodes in its module C_i
- k_i is the total degree of node i (i.e., the number of links connected to it)

Heat cartography map



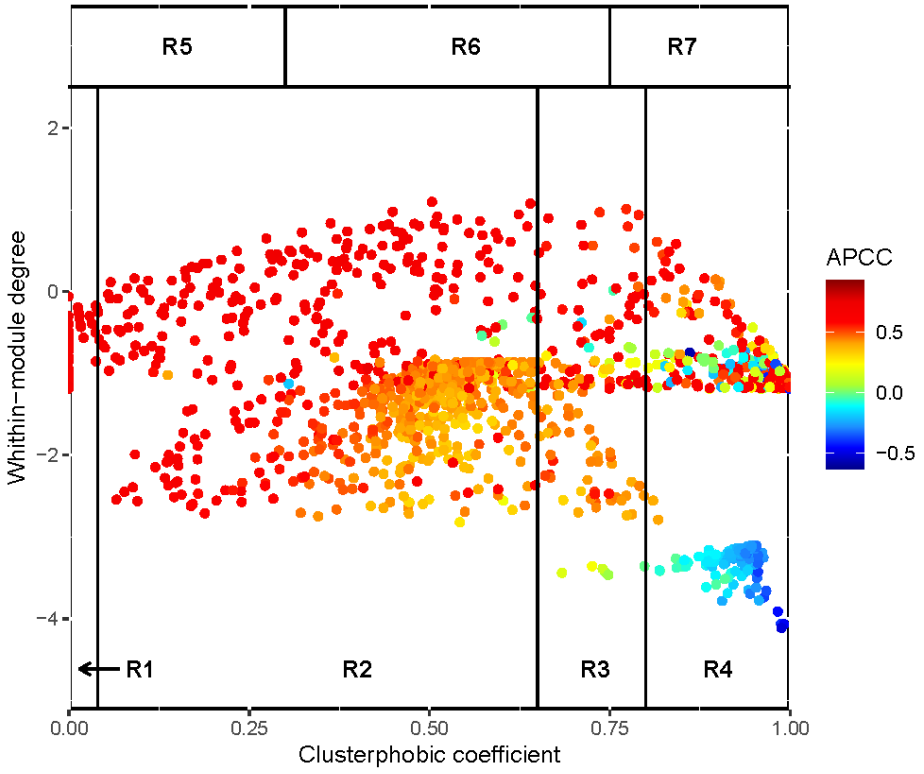
Clusterphobic coefficient (Kπ)

- K_π measures the "fear" of being confined in a cluster
- It evaluates the ratio of internal to external connections of a node and thus represents a measure of **global connectivity**
- A high value of K_π denotes nodes having much more external than internal links

$$K_\pi = 1 - \left(\frac{k_i^{in}}{k_i}\right)^2$$

Note that K_π = 0 when a node has only links within its module, i.e., it does not communicates with the other modules (k_i^{in}= k_i). On the contrary, K_π is close to 1 when the majority of its links are external to its own module.

Heat cartography map



- x-axis refers to K_π (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

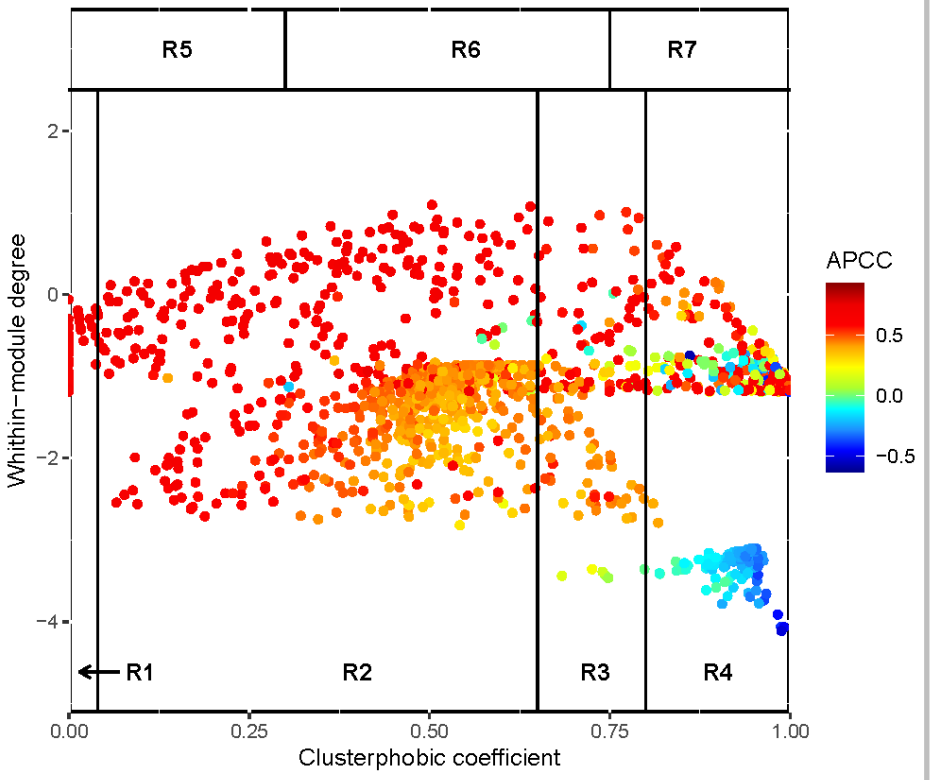
Within-module degree (z_g)

- z_g measures how much a node is a hub in its community and thus represents a measure of **local connectivity**

$$z_g^i = \frac{k_i^{in} - \bar{k}_{C_i}}{\sigma_{C_i}}$$

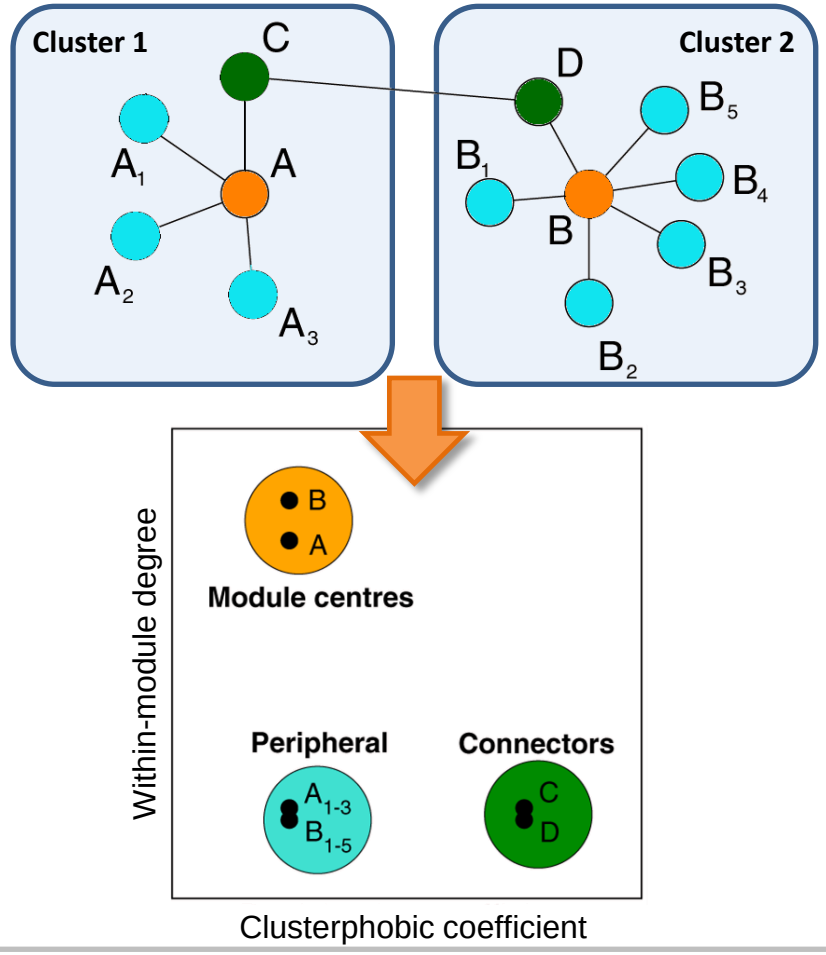
- k_i^{in} is the number of links of node *i* to nodes in its module C_i
- $\bar{k}_{C_i}$ and σ_{C_i} are the average and standard deviation of the total degree distribution of the nodes in the module C_i

Heat cartography map

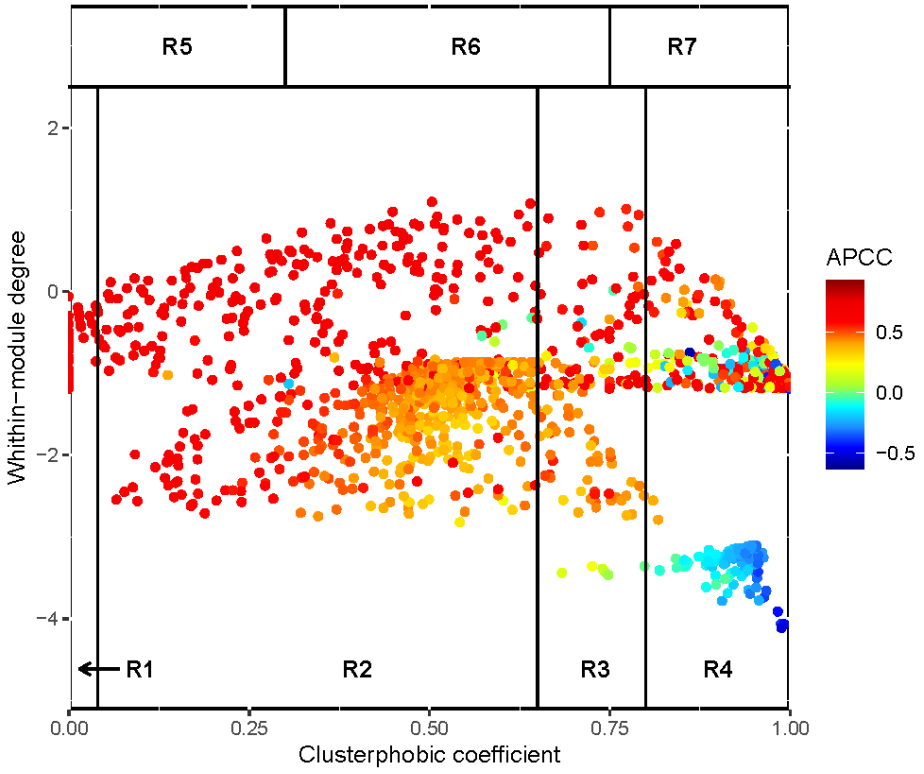


- x-axis refers to K_{π} (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

Roles of nodes in the correlation network



Heat cartography map



- x-axis refers to K_π (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

Roles of nodes in the correlation network

z_g and K_π divide the plane into R1-R7 regions, each defining a specific node role:

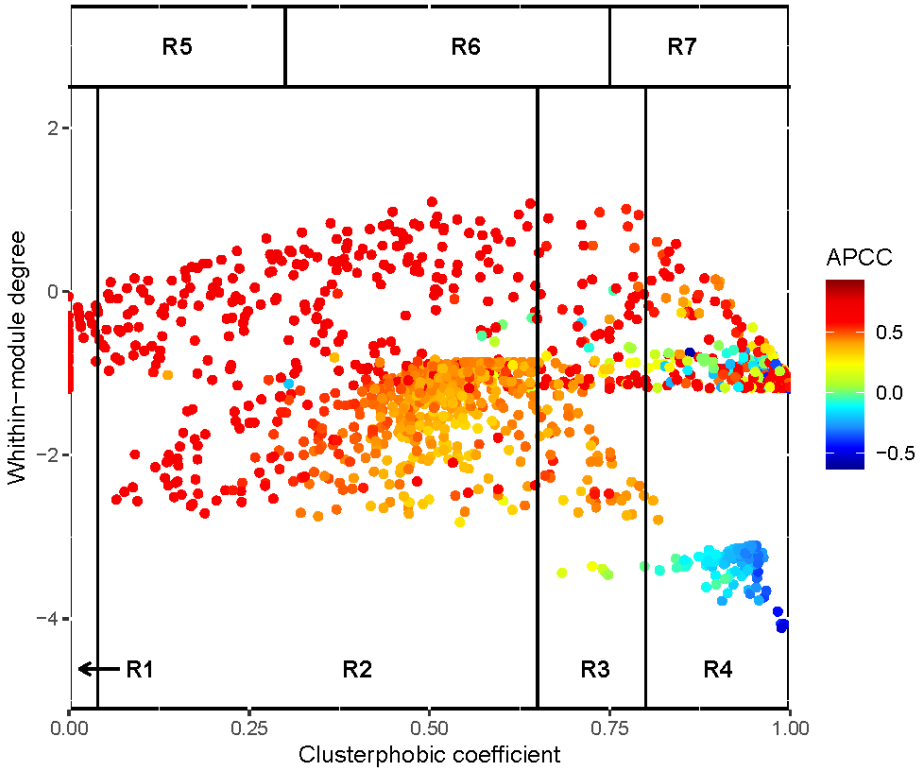
non local hub for $z_g < 2.5$

$K_\pi = 0$	Ultra-peripheral nodes (role R1)
$K_\pi \leq 0.625$	Peripheral nodes (role R2)
$0.62 < K_\pi \leq 0.8$	Non-hub connectors (role R3)
$K_\pi > 0.8$	Non-hub kinless nodes (role R4)

local hub for $z_g \geq 2.5$

$K_\pi = 0.3$	Provincial hubs (role R5)
$K_\pi \leq 0.75$	Connector hubs (role R6)
$K_\pi > 0.75$	Kinless hubs (role R7)

Heat cartography map



- x-axis refers to K_{π} (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

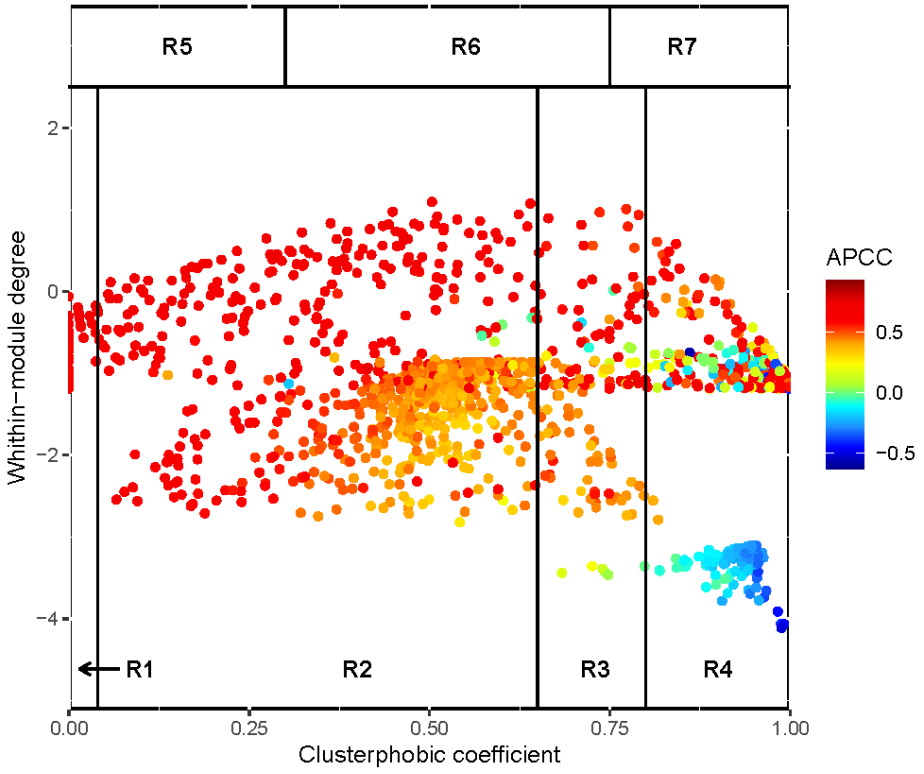
Heat cartography map

```
buildCartography <- function(df_stat,idx,res_deg,APCC,hub_class,network,output_file_attribu
te,
                             output_file_CartographyNetwork,output_file_CartographyNetwork_
R){

  deg <- res_deg$deg
  ideg <- res_deg$ideg

  cluster <- split(idx,idx$cluster)

  cluster_info <- lapply(names(cluster),function(k){

    y <- rownames(cluster[[k]])

    parameter <- computePz(y,deg,ideg)
    P <- parameter$P
    z <- parameter$z

    role <- getNodeRole(y,z,P)

    df <- data.frame(y,role$hub,role$region,role$type,deg[y,],ideg[y,],APCC[y,],hub_class[y
,],P,z,k)
    colnames(df) <- c("node", "Hub", "Region", "Type", "Total_Degree", "Internal_Degree",
                      "APCC", "Hub_classification", "P", "z", "Cluster_ID")

    return(df)
  })

  attribute <- data.frame(rbindlist(cluster_info))

  ind <- which( (attribute$`Total Degree` == 0) | (attribute$`Internal Degree` == 0)
)

  if(length(ind)>0){

    node_to_remove <- attribute$node[ind]

    found <- which( (network$source %in% node_to_remove) | (network$target %in%
node_to_remove) )
    network <- network[-found,]

    attribute <- attribute[-ind,]

  }
```

...

Scroll down

Heat cartography map



- x-axis refers to K_{π} (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

Heat cartography map

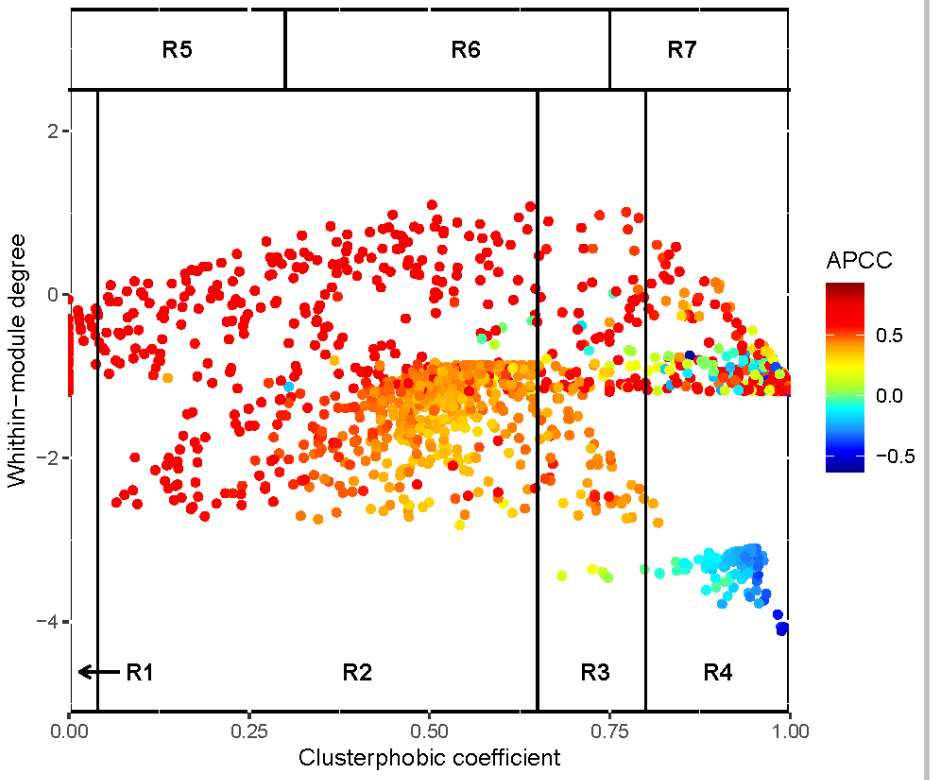
```
buildCartography <- function(df_stat,idx,res_deg,APCC,hub_class,network,output_file_attribu
te,
                             output_file_CartographyNetwork,output_file_CartographyNetwork_
R){

  deg <- res_deg$deg
  ideg <- res_deg$ideg

  cluster <- split(idx,idx$cluster)

  cluster_info <- lapply(names(cluster),function(k){

    y <- rownames(cluster[[k]])

    parameter <- computePz(y,deg,ideg)
    P <- parameter$P
    z <- parameter$z

    role <- getNodeRole(y,z,P)

    df <- data.frame(y,role$hub
,],P,z,k)
    colnames(df) <- c("node", "
                      "APCC", "

    return(df)

  })
```

Add statistical information from df_stat

...

```
  attribute <- merge(attribute, df_stat, by.x = "node", by.y = 0, all = F)

  write.table(attribute, output_file_attribute, row.names = F, col.names = T, sep =
"\t", quote = F)
  write.table(network, output_file_CartographyNetwork, row.names = F, col.names = T,
sep = "\t", quote = F)

  save(network, file = output_file_CartographyNetwork_R)

  res <- list(attribute = attribute, num_node = nrow(attribute), num_edge = nrow
(network))

  return(res)

}
```

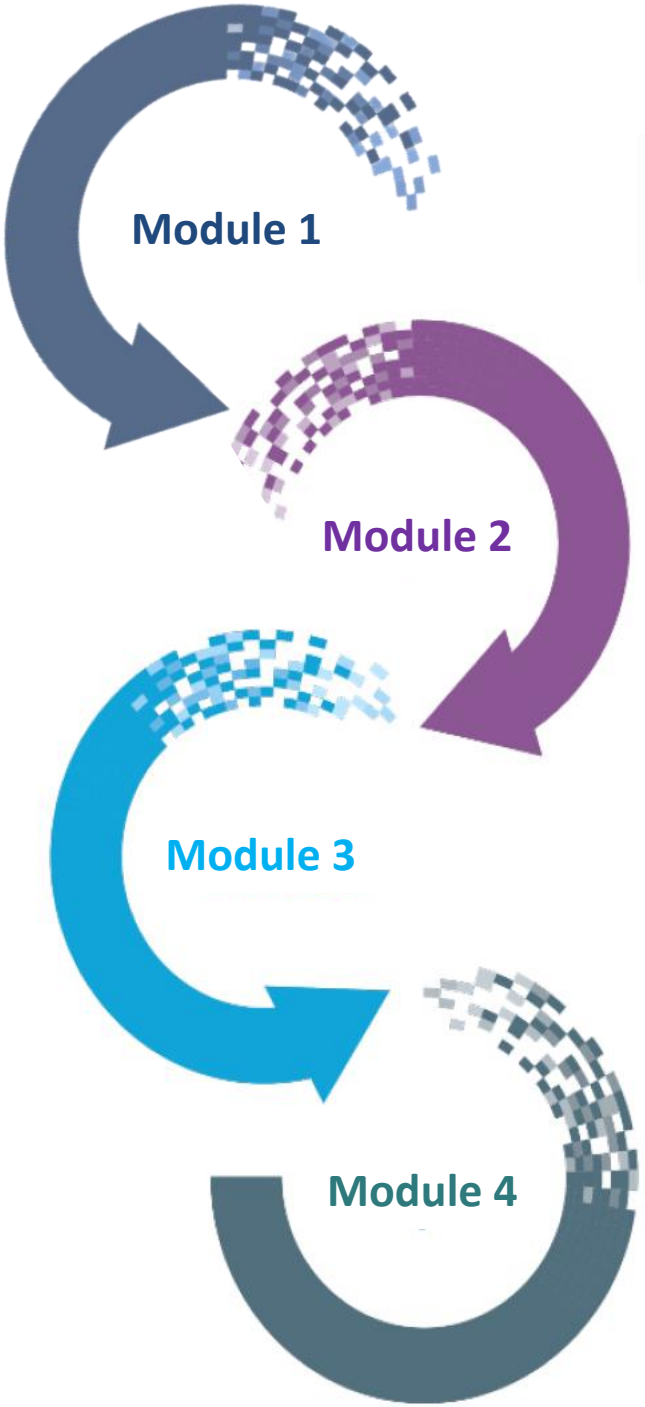
Heat cartography map



- x-axis refers to K_{π} (clusterphobic coefficient)
- y-axis refers to z_g (within-module degree)
- Color scale refers to APCC

Heat cartography map

```
getHeatCartography <- function(attribute,output_file){

  df <- attribute[,c("P","z","APCC")]

  m <- min(df$z) - 1
  M <- 3.5
  d <- 0.5

  my_color <- colorRampPalette(colors = c("blue4","blue","dodgerblue1","deepskyblue","cyan"
,"greenyellow","yellow","orange","red","red2","red4"))(100)

  p <- ggplot(df, aes(x = P, y = z, color = APCC)) + geom_point() +
    scale_x_continuous(expand = c(0, 0)) + scale_y_continuous(expand = c(0, 0)) +
    theme(panel.background = element_rect(fill = "white", colour = "black", size = 1)) +
    labs(x = "Clusterphobic coefficient", y = "Whithin-module degree") +
    geom_line(data = data.frame(x = c(0, 1), y = 2.5), aes(x = x , y = y),
              linetype = "solid", color = "black")  +
    geom_line(data = data.frame(x = 0.04, y = c(m, 2.5)), aes(x = x, y = y),
              linetype = "solid", color = "black") +
    geom_line(data = data.frame(x = 0.65, y = c(m, 2.5)), aes(x = x, y = y),
              linetype = "solid", color = "black") +
    geom_line(data = data.frame(x = 0.8, y = c(m, 2.5)), aes(x = x, y = y),
              linetype = "solid", color = "black") +
    geom_line(data = data.frame(x = 0.3, y = c(2.5,M)), aes(x = x, y = y),
              linetype = "solid", color = "black") +
    geom_line(data = data.frame(x = 0.75, y = c(2.5,M)), aes(x = x, y = y),
              linetype = "solid", color = "black") +
    annotate(geom="text", x = 0.1, y = m+d, label = "R1",
           color = "black", fontface = "bold") +
    annotate(geom="text", x = 0.4, y = m+d, label = "R2",
             color = "black", fontface = "bold") +
    annotate(geom="text", x = 0.73, y = m+d, label = "R3",
             color = "black", fontface = "bold") +
    annotate(geom="text", x = 0.9, y = m+d, label = "R4",
             color = "black", fontface = "bold") +
    annotate(geom="text", x = 0.15, y = M-d, label = "R5",
             color = "black", fontface = "bold") +

    geom_segment(aes(x = 0.07, y = m+d, xend = 0.015, yend = m+d),
                 arrow = arrow(length = unit(0.2, "cm")), colour = "black") +
    scale_color_gradientn(colours = my_color, limits=c(-1,1))

  print(p)

  savePDF(p,output_file)

}
```



Exploratory Data Analysis

Importing data

Pre-processing phase

Filtering phase

Network Analysis

Building the correlation network

Checking network integrity

Finding communities in the network

Switch Mining

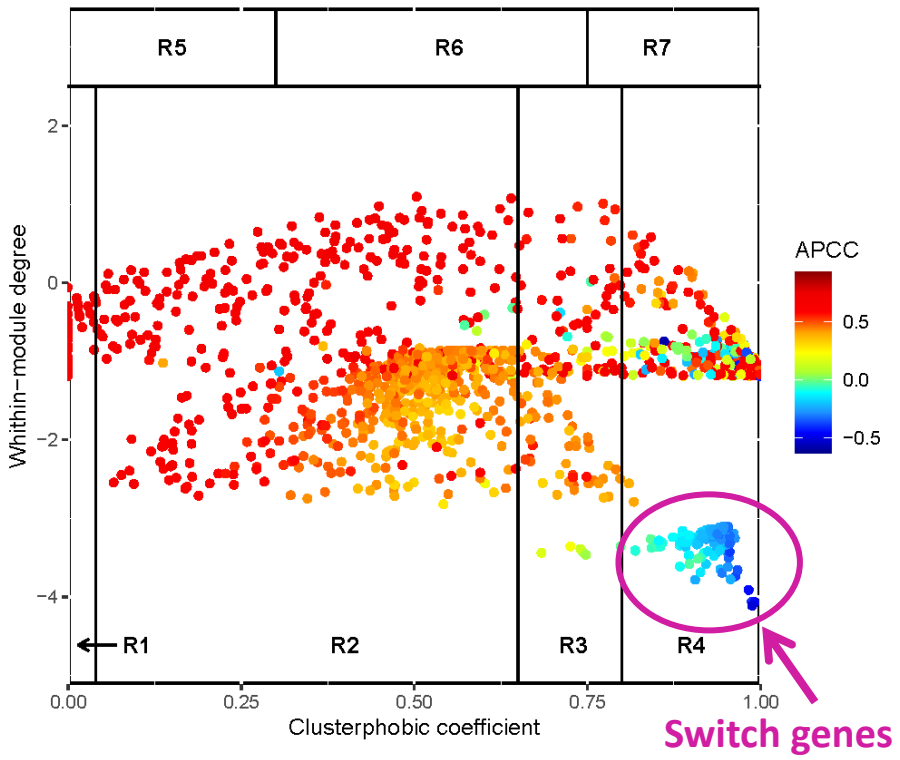
Building the heat cartography map

Identification of switch genes

Resilience Analysis

Evaluation of network resilience

Switch genes



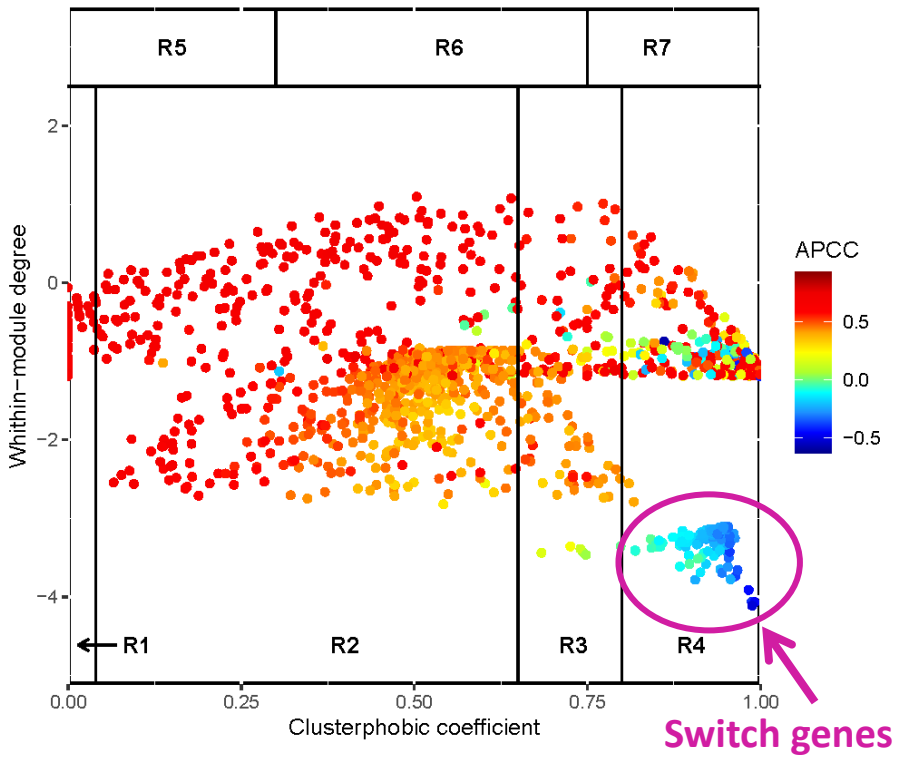
Switch genes

- Looking at the heat cartography map, we called **"switch genes"** the subset of the **fight-club hubs** falling in **R4** (i.e., mainly interacting outside their cluster)

- In particular, they satisfy the following topological and expression features:

  1. being not a hub in their own cluster (z_g<2.5)
  2. having many links outside their own cluster (K_π > 0.8)
  3. having a negative average weight of their incident links (APCC < 0)

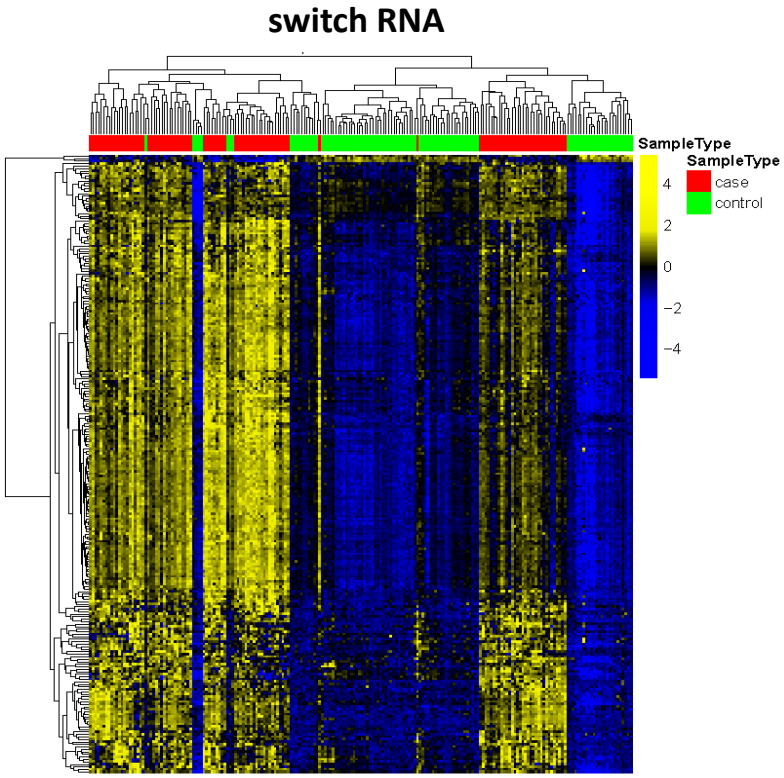
Switch genes



Switch genes

```
getSwitch <- function(attribute,output_file){

  ########################################
  # input parameters

  output_file_attribute_switch <- output_file$filename_attribute_switch
  output_file_switch <- output_file$filename_switch
  ########################################

  ind <- which(attribute$Region == "R4" &
                 attribute$Hub_classification == "FIGHT CLUB")

  attribute_switch <- attribute[ind,]

  switch <- attribute_switch$node

  write.table(attribute_switch, output_file_attribute_switch , row.names = F, col.names = T,
sep = "\t", quote = F)
  write.table(switch, output_file_switch, row.names = F, col.names = F, sep = "\t", quote = F
)

  return(attribute_switch)

}
```

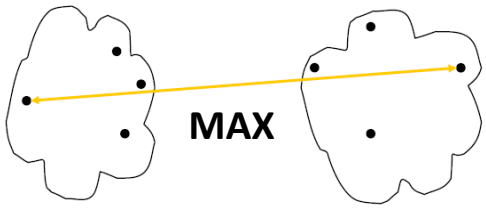
Heatmap of switch genes (RNAs)



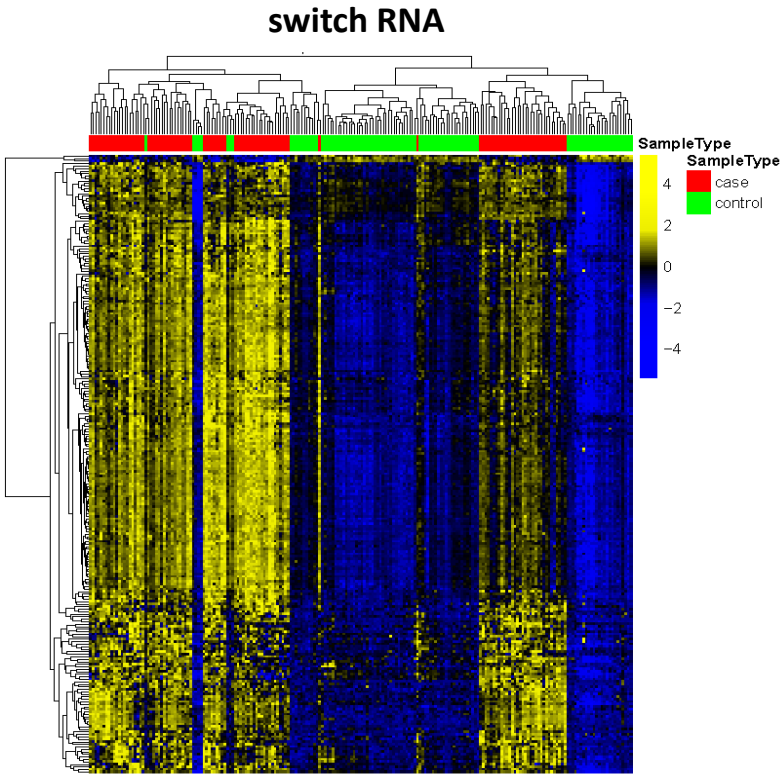
- Row refers to switch RNAs
- Columns refers to samples
- Colors represent different expression levels that increase from blue to yellow

Hierarchical clustering

- Hierarchical clustering of switch genes by using:
  - **Pearson correlation** as distance metric
  - linkage **complete** as clustering method (where distance is measured between the farthest pair of observations in two clusters)



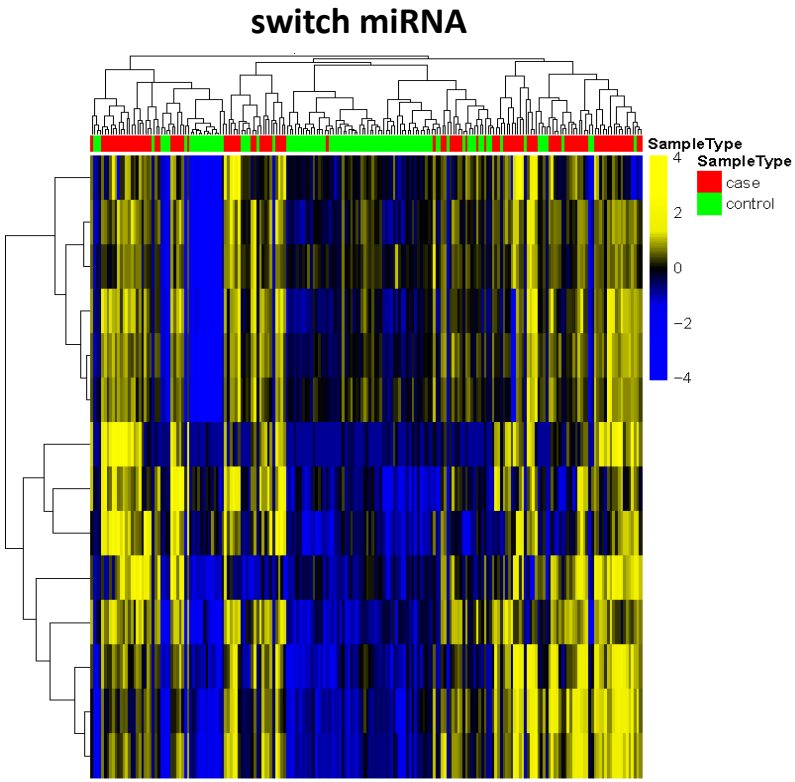
Heatmap of switch genes (RNAs)



- Row refers to switch RNAs
- Columns refers to samples
- Colors represent different expression levels that increase from blue to yellow

```
getHeatmap <- function(data.Filtered,output_file,title){

  ########################################
  # input parameters

  data <- data.Filtered$data
  control <- colnames(data.Filtered$data_control)
  case <- colnames(data.Filtered$data_case)
  ########################################
  samples <- ifelse( (colnames(data) %in% control), "control", "case"
)
  annotation <- data.frame(SampleType = samples)
  rownames(annotation) <- colnames(data)

  annotation_colors <- list(SampleType = c(case = "red", control =
"green"))

  colorbar <- colorRampPalette(colors = c("blue","blue1","blue2"
,"black","yellow2","yellow1","yellow"))(100)

  out <- pheatmap(data, scale = "row",
                  border_color = NA,
                  clustering_distance_rows = "correlation",
                  clustering_distance_cols = "correlation",
                  clustering_method = "complete",
                  cluster_cols = T,
                  cluster_rows = T,
                  annotation_col = annotation,
                  annotation_colors = annotation_colors,
                  color = colorbar,
                  show_rownames = F,
                  show_colnames = F,
                  main = title
                  #width = 10,
                  #height = 10,
                  #treeheight_row = 30,
                  #fontsize = 10,
                  #cellwidth = 0.3,
                  #cellheight = 0.3
  )

  saveHeatmapPDF(out,output_file)

}
```

Heatmap of switch genes (RNAs)



- Row refers to switch miRNAs
- Columns refers to samples
- Colors represent different expression levels that increase from blue to yellow

```
getHeatmap <- function(data.Filtered,output_file,title){

  ########################################
  # input parameters

  data <- data.Filtered$data
  control <- colnames(data.Filtered$data_control)
  case <- colnames(data.Filtered$data_case)
  ########################################
  samples <- ifelse( (colnames(data) %in% control), "control", "case"
)
  annotation <- data.frame(SampleType = samples)
  rownames(annotation) <- colnames(data)

  annotation_colors <- list(SampleType = c(case = "red", control =
"green"))

  colorbar <- colorRampPalette(colors = c("blue","blue1","blue2"
,"black","yellow2","yellow1","yellow"))(100)

  out <- pheatmap(data, scale = "row",
                  border_color = NA,
                  clustering_distance_rows = "correlation",
                  clustering_distance_cols = "correlation",
                  clustering_method = "complete",
                  cluster_cols = T,
                  cluster_rows = T,
                  annotation_col = annotation,
                  annotation_colors = annotation_colors,
                  color = colorbar,
                  show_rownames = F,
                  show_colnames = F,
                  main = title
                  #width = 10,
                  #height = 10,
                  #treeheight_row = 30,
                  #fontsize = 10,
                  #cellwidth = 0.3,
                  #cellheight = 0.3
  )

  saveHeatmapPDF(out,output_file)

}
```

At the end of Module 3, you will obtain the list of switch genes