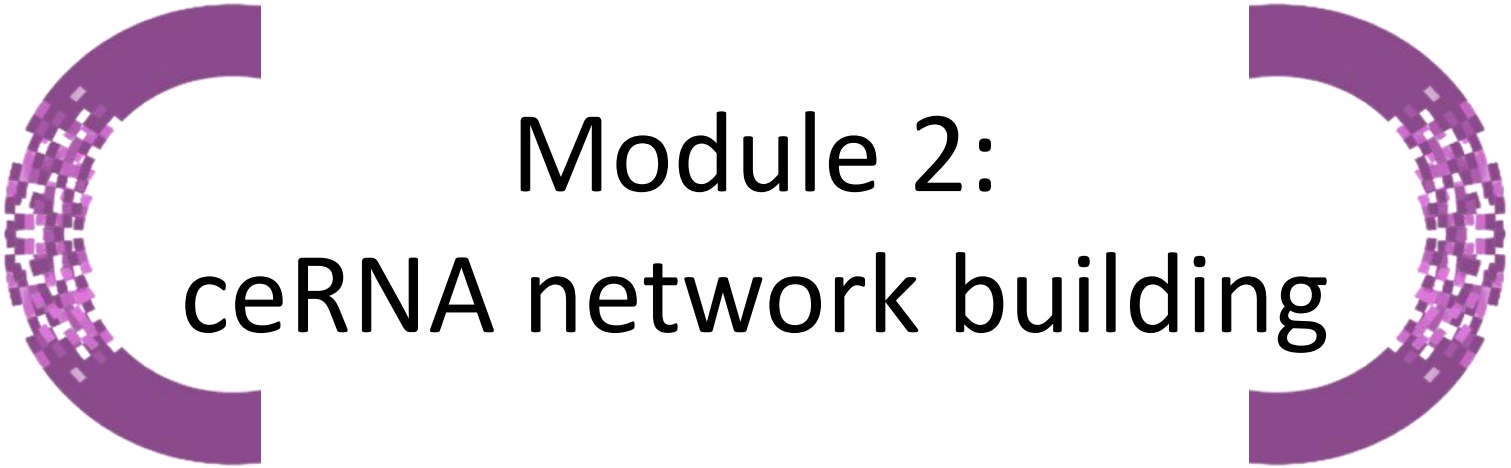
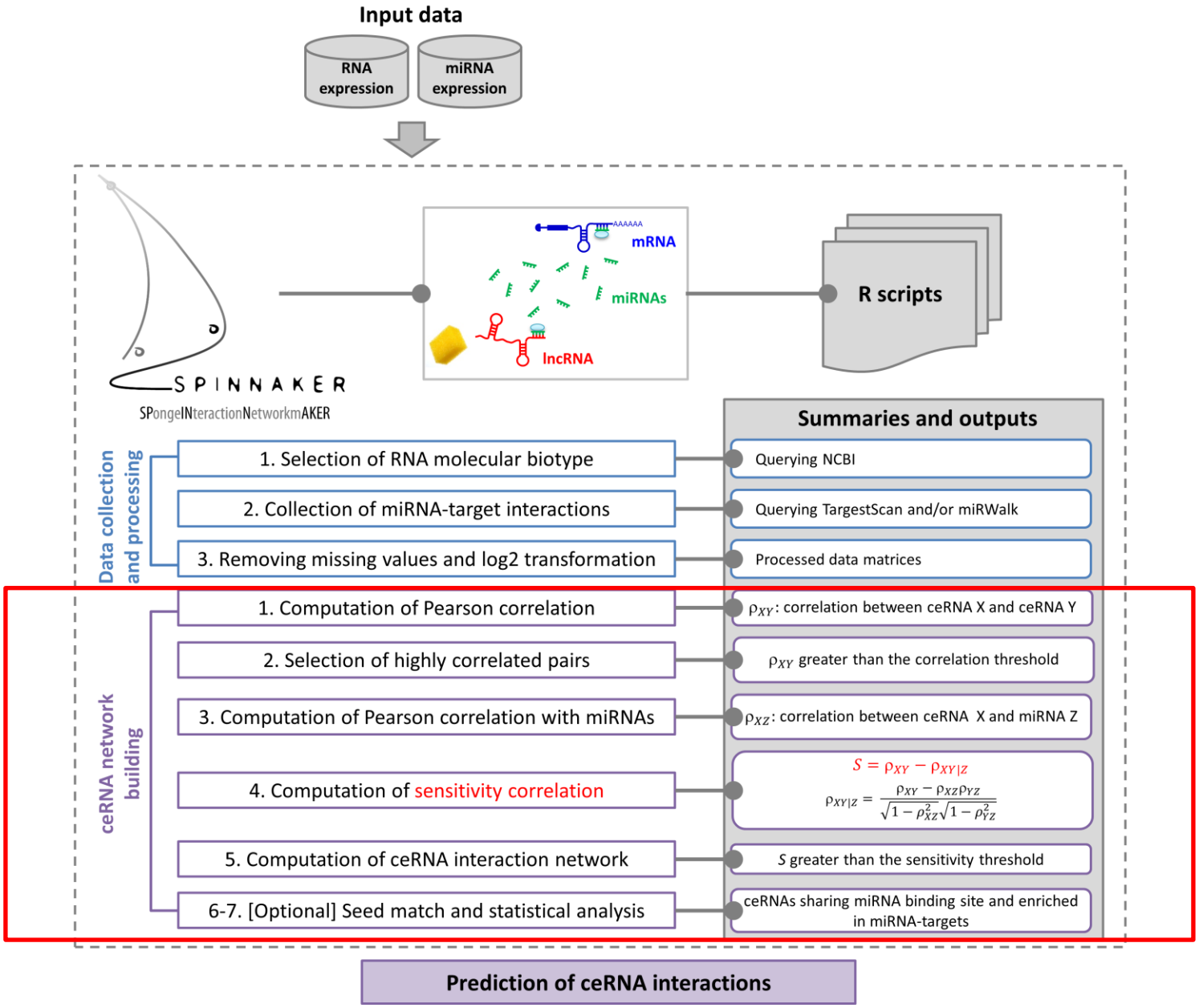


Module 2:
ceRNA network building


Input data
RNA expression
miRNA expression
SPINNAKER
SPongeINteractionNetworkmAKER
mRNA
miRNAs
lncRNA
R scripts
Summaries and outputs
Data collection and processing
1. Selection of RNA molecular biotype
Querying NCBI
2. Collection of miRNA-target interactions
Querying TargestScan and/or miRWalk
3. Removing missing values and log2 transformation
Processed data matrices
ceRNA network building
1. Computation of Pearson correlation
ρ_{XY}: correlation between ceRNA X and ceRNA Y
2. Selection of highly correlated pairs
ρ_{XY} greater than the correlation threshold
3. Computation of Pearson correlation with miRNAs
ρ_{XZ}: correlation between ceRNA X and miRNA Z
4. Computation of sensitivity correlation
$S = \rho_{XY} - \rho_{XY|Z}$
$\rho_{XY|Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{YZ}}{\sqrt{1-\rho_{XZ}^2}\sqrt{1-\rho_{YZ}^2}}$
5. Computation of ceRNA interaction network
S greater than the sensitivity threshold
6-7. [Optional] Seed match and statistical analysis
ceRNAs sharing miRNA binding site and enriched in miRNA-targets
Prediction of ceRNA interactions

ceRNA network building

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

...

- The goal of this module is to build the **ceRNA interactions network**
- This module is composed of **seven steps**:
  i. Computation of Pearson correlation
  ii. Selection of highly correlated pairs
  iii. Computation of Pearson correlation with miRNAs
  iv. Computation of sensitivity correlation
  v. Computation of ceRNA interaction network
  vi. [Optional] Search seed-match for all triplets
  vii. [Optional] Compute statistical analysis

ceRNA network building

...

```
  ######################################
  # STEP 4

  print("STEP 4: compute sensitivity correlation")

  triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
  ######################################
  # STEP 5

  print("STEP 5: compute ceRNA interaction network")

  threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

  ceRNA <- selectTriplets(triplets,threshold_sensitivity)

  res <- list(ceRNA = ceRNA,
              threshold_corr = threshold_corr,
              threshold_sensitivity = threshold_sensitivity)
  ######################################

  if(searchSeedMatch == "YES"){

    ######################################
    # STEP 6

    print("STEP 6: search seed-match for all triplets")

    triplets <- searchSeedMatch(triplets,miRNAtarget)
    ######################################
    # STEP 7

    print("STEP 7: compute statistical analysis")

    ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

    res_stat <- list(ceRNA_pval = ceRNA_pval)
    ######################################

    res <- c(res,res_stat)

  }

  return(res)
}
```

- The goal of this module is to build the **ceRNA interactions network**
- This module is composed of **seven steps**:
  i. Computation of Pearson correlation
  ii. Selection of highly correlated pairs
  iii. Computation of Pearson correlation with miRNAs
  iv. Computation of sensitivity correlation
  v. Computation of ceRNA interaction network
  vi. [Optional] Search seed-match for all triplets
  vii. [Optional] Compute statistical analysis

Pearson correlation bewteen ceRNAs

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

Correlation between two variables

- The existence of a linear relationship between two normally distributed continuous variables (e.g., gene expression values of RNA X and RNA Y) can be expressed by the **Pearson correlation coefficient ρ**

$$\rho_{xy} = \frac{\sum_{i=1}^{col}\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{col}\left(x_i - \bar{x}\right)^2 \sum_{i=1}^{col}\left(y_i - \bar{y}\right)^2}}$$

- ρ coefficient varies between -1 and 1 and shows strength (value) and direction (sign) of correlation

Pearson correlation bewteen ceRNAs

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

computeCorrelation

- SPINNAKER computes the Pearson correlation between the expression profiles of the ceRNA pairs (ρ_{XY}).

```
computeCorrelation <- function(data1,data2){

  dim_min <- min(nrow(data1),nrow(data2),2000)

  size <- round_any(dim_min, 100, f = floor)

  rho <- bigcor(t(data1), t(data2), size = size, fun = "cor",
                verbose = F, use = "pairwise.complete.obs")

  rho <- rho[1:nrow(data1), 1:nrow(data2)]
  rownames(rho) <- rownames(data1)
  colnames(rho) <- rownames(data2)

  return(rho)
}
```

Select Pairs

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

selectPairs

- SPINNAKER selects the ceRNA pairs with ρ_{XY} greater than a defined threshold (by default equal to the 99th percentile)

```
selectPairs <- function(rho,thr){

  ind <- which(rho >= thr, arr.ind = T)

  if( nrow(ind) > 0 ){

    pairs <- data.frame(ceRNA1 = rownames(rho)[ind[,1]],
                        ceRNA2 = colnames(rho)[ind[,2]],
                        correlation = rho[ind])

  }else{

    stop("No pairs with the selected correlation threshold")

  }

  return(pairs)
}
```

Select Pairs

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

selectPairs

- SPINNAKER selects the ceRNA pairs with ρ_{XY} greater than a defined threshold (by default equal to the 99th percentile)

```
selectPairs <- function(rho,thr){

  ind <- which(rho >= thr, arr.ind = T)

  if( nrow(ind) > 0 ){

    pairs <- data.frame(ceRNA1 = rownames(rho)[ind[,1]],
                        ceRNA2 = colnames(rho)[ind[,2]],
                        correlation = rho[ind])

  }else{

    stop("No pairs with the selected correlation threshold")

  }
```

⚠ **Caveat:** if the chosen threshold does not allow to select any pairs an error message appears, SPINNAKER stops running, and you have to change the input parameter

Pearson correlation with miRNAs

```
ceRNANetworkBuilding <- function(){

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

  ceRNA1 <- input_parameter$ceRNA1
  ceRNA2 <- input_parameter$ceRNA2

  data_ceRNA1 <- data$data_ceRNA1
  data_ceRNA2 <- data$data_ceRNA2
  data_miRNA <- data$data_miRNA

  threshold_prc_corr <- input_parameter$threshold_prc_corr
  threshold_prc_sensitivity <- input_parameter$threshold_prc_sensitivity

  searchSeedMatch <- input_parameter$searchSeedMatch

  if(searchSeedMatch == "YES") {
    miRNAtarget <- data$miRNAtarget
  }

  filename_heatmap <- output_file$filename_heatmap
  ######################################
  # STEP 1

  print("STEP 1: compute Pearson correlation")

  rho_ceRNA1_ceRNA2 <- computeCorrelation(data_ceRNA1,data_ceRNA2)

  if(all(ceRNA1 == ceRNA2)){
    rho_ceRNA1_ceRNA2[upper.tri(rho_ceRNA1_ceRNA2,diag = T)] <- NA
  }
  ######################################
  # STEP 2

  print("STEP 2: select highly correlated pairs")

  threshold_corr <- as.numeric(quantile(c(rho_ceRNA1_ceRNA2),threshold_prc_corr,na.rm = T))

  pairs <- selectPairs(rho_ceRNA1_ceRNA2,threshold_corr)

  data_ceRNA1 <- data_ceRNA1[unique(pairs$ceRNA1),]
  data_ceRNA2 <- data_ceRNA2[unique(pairs$ceRNA2),]
  ######################################
  # STEP 3

  print("STEP 3: compute Pearson correlation with miRNAs")

  rho_ceRNA1_miRNA <- computeCorrelation(data_ceRNA1,data_miRNA)
  rho_ceRNA2_miRNA <- computeCorrelation(data_ceRNA2,data_miRNA)
  ######################################
```

computeCorrelation

- SPINNAKER computes the Pearson correlation between the expression profiles of ceRNA X and miRNA Z (ρ_{XZ}) and the expression profiles of ceRNA Y and miRNA Z (ρ_{YZ})

```
computeCorrelation <- function(data1,data2){

  dim_min <- min(nrow(data1),nrow(data2),2000)

  size <- round_any(dim_min, 100, f = floor)

  rho <- bigcor(t(data1), t(data2), size = size, fun = "cor",
                verbose = F, use = "pairwise.complete.obs")

  rho <- rho[1:nrow(data1), 1:nrow(data2)]
  rownames(rho) <- rownames(data1)
  colnames(rho) <- rownames(data2)

  return(rho)
}
```

Sensitivity correlation

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

computeSensitivity

- SPINNAKER computes the **sensitivity correlation** as:

$$S = \rho_{XY} - \rho_{XY|Z}$$

where ρ_{XY} is the Pearson and $\rho_{XY|Z}$ the **partial correlation** between ceRNA X and ceRNA Y controlling for the miRNA Z defined as:

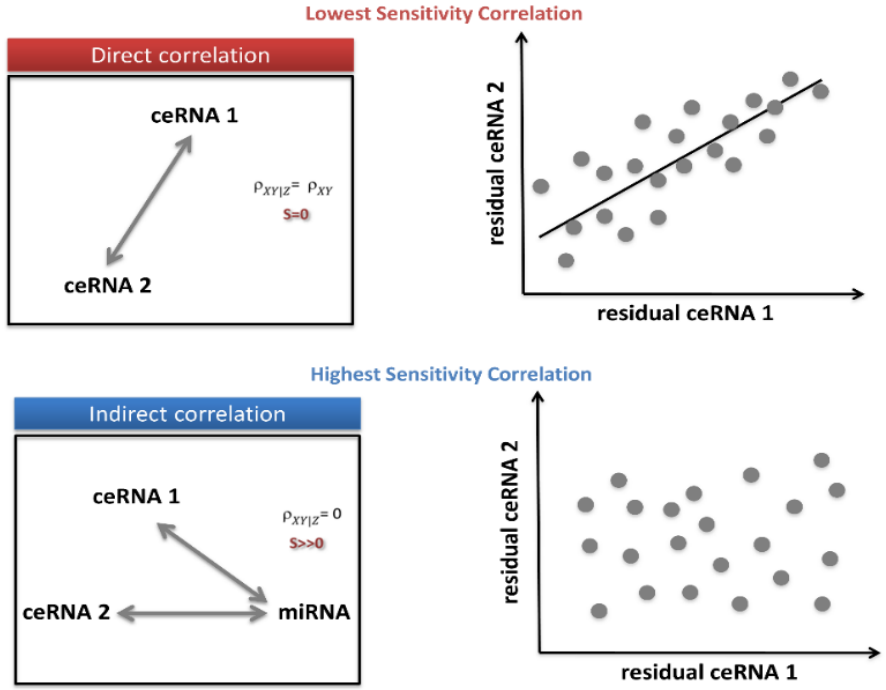
$$\rho_{XY|Z} = \frac{\rho_{XY} - \rho_{XZ}\rho_{YZ}}{\sqrt{1-\rho_{XZ}^2}\sqrt{1-\rho_{YZ}^2}}$$

Sensitivity correlation

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

computeSensitivity



Sensitivity correlation

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

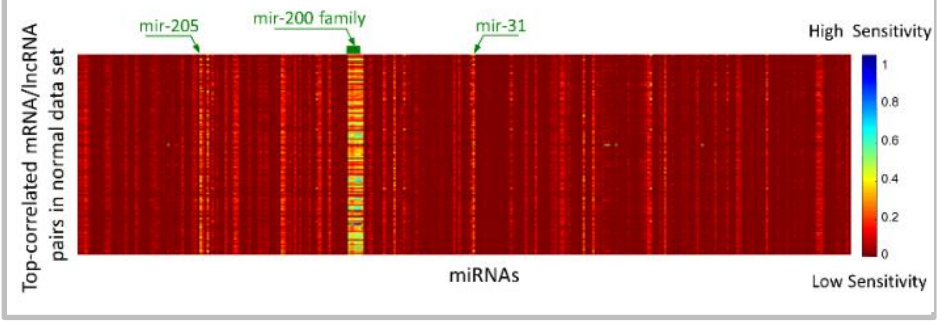
computeSensitivity

```
computeSensitivity <- function(pairs_xy,rho_xz,rho_yz,filename_heatmap){

  z <- colnames(rho_xz)

  list <- apply(pairs_xy,1,function(pair){

    x <- pair[1]
    y <- pair[2]
    rxy <- as.numeric(pair[3])

    rxz <- rho_xz[x,]
    ryz <- rho_yz[y,]

    pc <- ( rxy - (rxz * ryz) ) / ( sqrt(1 - rxz^2) * sqrt(1 - ryz^2) )

    s <- rxy - pc

    df <- data.frame(ceRNA1 = x, ceRNA2 = y, miRNA = z,
                     correlation = rxy, partial_correlation = pc,
                     sensitivity = s, row.names = NULL)

  })

  triplets <- rbindlist(list)

  makeHeatmap(z,list,filename_heatmap)

  return(triplets)
}
```

Sensitivity correlation

makeHeatmap

```
makeHeatmap <- function(mir,pairs,filename_heatmap){

  size <- 5000

  n_pairs <- length(pairs)

  n_group <- floor(n_pairs/size)

  for(i in 1:n_group){

    start <- 1 + size * (i-1)
    end <- size * i

    mat <- sapply(pairs[start:end], function(x){ x$sensitivity })

    mat[mat<0] <- 0
    mat <- t(mat)
    colnames(mat) <- mir
    mat <-mat[,sort(colnames(mat))]

    filename_heatmap_i <- paste0(filename_heatmap,"_",i,".pdf")

    plotHeatmap(mat,filename_heatmap_i)

  }
}
```



computeSensitivity

```
computeSensitivity <- function(pairs_xy,rho_xz,rho_yz,filename_heatmap){

  z <- colnames(rho_xz)

  list <- apply(pairs_xy,1,function(pair){

    x <- pair[1]
    y <- pair[2]
    rxy <- as.numeric(pair[3])

    rxz <- rho_xz[x,]
    ryz <- rho_yz[y,]

    pc <- ( rxy - (rxz * ryz) ) / ( sqrt(1 - rxz^2) * sqrt(1 - ryz^2) )

    s <- rxy - pc

    df <- data.frame(ceRNA1 = x, ceRNA2 = y, miRNA = z,
                     correlation = rxy, partial_correlation = pc,
                     sensitivity = s, row.names = NULL)

  })

  triplets <- rbindlist(list)

  makeHeatmap(z,list,filename_heatmap)

  return(triplets)
}
```

- SPINNAKER computes the **heatmap** representing the sensitivity correlation S, computed for the top-correlated ceRNA pairs

Select Triplets

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

selectTriplets

- SPINNAKER selects XYZ triplets with S greater than a defined threshold (by default equal to the 99th percentile)

```
selectTriplets <- function(df,thr){

  s <- df$sensitivity

  ind <- which(s >= thr)

  if( length(ind) > 0 ){

    ceRNA <- df[ind,]

  }else{

    stop("No triplets with the selected sensitivity threshold")

  }

  return(ceRNA)
}
```

⚠ **Caveat:** if the chosen threshold does not allow to select any triplets an error message appears, SPINNAKER stops running, and you have to change the input parameter

Select Triplets

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

selectTriplets

- The output of this step is the **ceRNA interactions network** where nodes represent ceRNAs with highly correlated expression profiles, while edges represent miRNAs mediating their interactions
- A link between two nodes (ceRNAs) occurs if they fulfilled the following conditions:
  1. matching **high** values of the **Pearson correlation** between their expression profiles
  2. matching **high** values of the **sensitivity correlation**

ceRNA interaction network

...

```
  ######################################
  # STEP 4

  print("STEP 4: compute sensitivity correlation")

  triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
  ######################################
  # STEP 5

  print("STEP 5: compute ceRNA interaction network")

  threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

  ceRNA <- selectTriplets(triplets,threshold_sensitivity)

  res <- list(ceRNA = ceRNA,
              threshold_corr = threshold_corr,
              threshold_sensitivity = threshold_sensitivity)
  ######################################

  if(searchSeedMatch == "YES"){

    ######################################
    # STEP 6

    print("STEP 6: search seed-match for all triplets")

    triplets <- searchSeedMatch(triplets,miRNAtarget)
    ######################################
    # STEP 7

    print("STEP 7: compute statistical analysis")

    ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

    res_stat <- list(ceRNA_pval = ceRNA_pval)
    ######################################

    res <- c(res,res_stat)

  }

  return(res)
}
```

ceRNA_network.txt

source target interaction

	A	B	C	D	E	F
1	ceRNA1	ceRNA2	miRNA	correlation	partial_correlation	sensitivity
2	ABCC9	AGAP11	hsa-miR-452-5p	0.7251236	0.46708507	0.258039
3	ABHD15	AGAP11	hsa-miR-224-5p	0.7250564	0.477516646	0.24754
4	ABHD15	AGAP11	hsa-miR-452-5p	0.7250564	0.448402624	0.276654
5	ACACB	AGAP11	hsa-miR-452-5p	0.7299151	0.460145715	0.269769
6	ACSL4	AGAP11	hsa-miR-452-5p	0.7003977	0.443672326	0.256725
7	ACSM5	AGAP11	hsa-miR-224-5p	0.7034438	0.457025091	0.246419
8	ACSM5	AGAP11	hsa-miR-452-5p	0.7034438	0.39869897	0.304745
9	ADIPOQ	AGAP11	hsa-miR-452-5p	0.7204383	0.438231989	0.282206
10	ADRB1	AGAP11	hsa-miR-224-5p	0.7099421	0.462706234	0.247236
11	ADRB1	AGAP11	hsa-miR-452-5p	0.7099421	0.426378342	0.283564
12	ANO3	AGAP11	hsa-miR-224-5p	0.6939186	0.425771987	0.268147
13	ANO3	AGAP11	hsa-miR-452-5p	0.6939186	0.401611936	0.292307
14	ANTXR2	AGAP11	hsa-miR-452-5p	0.748311	0.502458952	0.245852
15	AOC3	AGAP11	hsa-miR-224-5p	0.7265339	0.468498677	0.258035
16	AOC3	AGAP11	hsa-miR-452-5p	0.7265339	0.425578625	0.300955

Seed-match analysis

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

searchSeedMatch

- SPINNAKER searches for the **seed-match** of all the highly correlated pairs with the miRNA mediating their interactions, in order to restrict the above selected triplets to those including only ceRNAs that are targets of the shared miRNA
- This step and the following one are performed only if searchSeedMatch = “YES” in config.R

⚠ **Caveat:** this step could take some time

Seed-match analysis

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

searchSeedMatch

```
searchSeedMatch <- function(triplets,miRNAtarget){

  mir_common <- intersect(unique(triplets$miRNA),names(miRNAtarget))

  list <- lapply(mir_common,function(x){

    target <- miRNAtarget[[x]]

    ind <- which(triplets$miRNA == x)
    pairs <- triplets[ind,c("ceRNA1","ceRNA2")]

    seed_match <- apply(pairs,1,function(p){

      condition <- length(intersect(p,target)) == 2

      ifelse(condition,"yes","no")

    })

    df <- data.frame(pairs,miRNA=x,seed_match)

  })

  df_tmp <- rbindlist(list)

  triplets <- merge(triplets, df_tmp, by = c("ceRNA1","ceRNA2","miRNA"), all.x = T)

  return(triplets)
}
```

Seed-match analysis

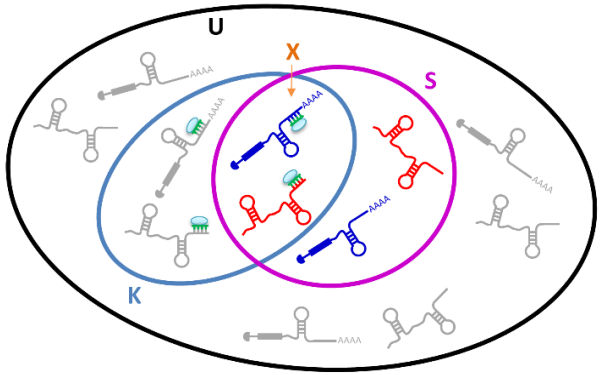
...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

StatisticalAnalysis

- SPINNAKER performs a seed-match enrichment analysis by calculating the following p-value resulting from the hypergeometric test:

$$p = 1 - \sum_{i=0}^{X-1} \frac{\binom{K}{i}\binom{U-K}{S-i}}{\binom{U}{S}} = \sum_{i=X}^{S} \frac{\binom{K}{i}\binom{U-K}{S-i}}{\binom{U}{S}}$$



- **U (universe) :** number of the top correlated RNA pairs
- **K (property):** number of RNA pairs sharing the binding site for the miRNA under test
- **S (selection):** number of RNA pairs with high sensitivity correlation for the miRNA under test
- **X (intersection):** number of RNA pairs with high sensitivity correlation sharing the binding site for the miRNA under test

Seed-match analysis

...

```
########################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
########################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
########################################

if(searchSeedMatch == "YES"){

  ########################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  ########################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  ########################################

  res <- c(res,res_stat)

}

return(res)
}
```

StatisticalAnalysis

```
statisticalAnalysis <- function(pairs,triplets,ceRNA,miRNAtarget){

  ceRNA <- removeNoTarget(ceRNA,miRNAtarget)

  # universe: highly correlated pairs (rho > thr)
  u <- nrow(pairs)

  mir_common <- intersect(unique(ceRNA$miRNA),names(miRNAtarget))

  list <- lapply(mir_common, function(m){

    ind <- which(triplets$miRNA == m)
    triplets_mir <- triplets[ind,]

    found_yes <- which(triplets_mir$seed_match == "yes")

    df <- NULL

    if( length(found_yes) > 0 ){

      # property: highly correlated pairs with the binding site for that mir
      pairs_property <- triplets_mir[found_yes,c("ceRNA1","ceRNA2","miRNA")]
      k <- nrow(pairs_property)

      # selection: highly correlated pairs with sensitivity > thr for that mir
      pairs_selection <- ceRNA[ceRNA$miRNA == m,c("ceRNA1","ceRNA2","miRNA")]
      s <- nrow(pairs_selection)

      pairs_intersection <- merge(pairs_property, pairs_selection,
                                  by = c("ceRNA1","ceRNA2","miRNA"), all = F)
      x <- nrow(pairs_intersection)

      n <- u - k

      p_value <- sum(dhyper(x:s,k,n,s))

      if (nrow(pairs_intersection) > 0){
        df <- data.frame(pairs_intersection, p_value = p_value,
                         check.names = F, row.names = NULL)
      }else{
        df <- NULL
      }
    }

    return(df)
  })

  df <- rbindlist(list)

  if(nrow(df) > 0){

    ceRNA_pval <- merge(ceRNA, df, by = c("ceRNA1","ceRNA2","miRNA"), all = F)

  }else{

    ceRNA_pval <- NULL

  }

  return(ceRNA_pval)
}
```

Seed-match analysis

...

```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

StatisticalAnalysis

- The output of this step is the **ceRNA interaction network** with the additional information of p-values
- A link between two nodes (ceRNAs) occurs if they fulfilled the following conditions:
  1. matching high values of the Pearson correlation between their expression profiles
  2. matching high values of the sensitivity correlation
  3. sharing the binding sites for miRNAs

ceRNA interaction network with p-value

...

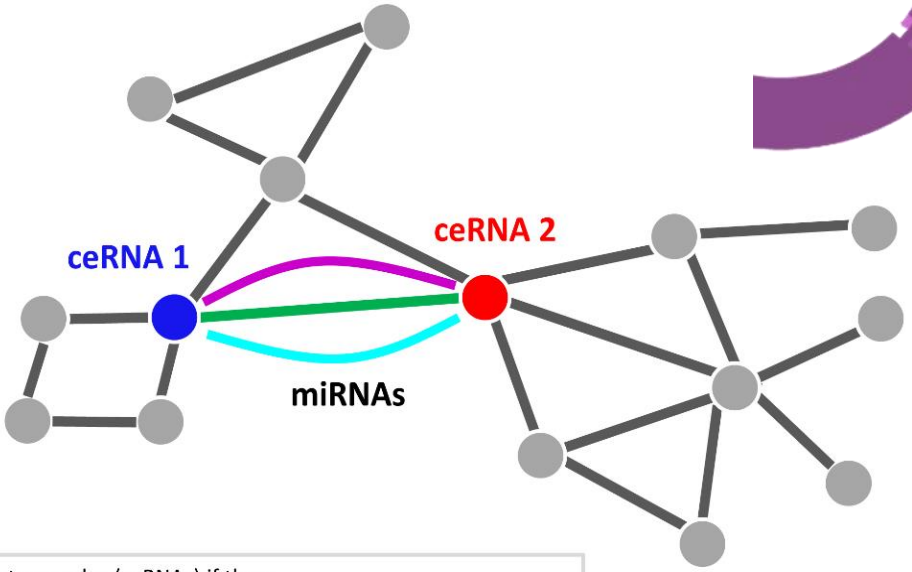
```
#####################################
# STEP 4

print("STEP 4: compute sensitivity correlation")

triplets <- computeSensitivity(pairs,rho_ceRNA1_miRNA,rho_ceRNA2_miRNA,filename_heatmap)
#####################################
# STEP 5

print("STEP 5: compute ceRNA interaction network")

threshold_sensitivity <- as.numeric(quantile(triplets$sensitivity,threshold_prc_sensitivity))

ceRNA <- selectTriplets(triplets,threshold_sensitivity)

res <- list(ceRNA = ceRNA,
            threshold_corr = threshold_corr,
            threshold_sensitivity = threshold_sensitivity)
#####################################

if(searchSeedMatch == "YES"){

  #####################################
  # STEP 6

  print("STEP 6: search seed-match for all triplets")

  triplets <- searchSeedMatch(triplets,miRNAtarget)
  #####################################
  # STEP 7

  print("STEP 7: compute statistical analysis")

  ceRNA_pval <- statisticalAnalysis(pairs,triplets,ceRNA,miRNAtarget)

  res_stat <- list(ceRNA_pval = ceRNA_pval)
  #####################################

  res <- c(res,res_stat)

}

return(res)
}
```

ceRNA_network_pval.txt

source **target** **interaction** **p-value**

	A	B	C	D	E	F	G
1	ceRNA1	ceRNA2	miRNA	correlatio	partial_co	sensitivity	p_value
2	ABAT	EGOT	hsa-miR-200b-3p	0.723733	0.378208	0.345525418	9.55E-89
3	ABAT	EGOT	hsa-miR-200c-3p	0.723733	0.3842	0.339532842	4.59E-81
4	ABCA1	EMX2OS	hsa-miR-200b-3p	0.750324	0.493991	0.256333696	9.55E-89
5	ABCC9	H19	hsa-miR-200b-3p	0.698159	0.395393	0.302765546	9.55E-89
6	ABCC9	H19	hsa-miR-200c-3p	0.698159	0.431533	0.266625932	4.59E-81
7	ABI2	EGOT	hsa-miR-200b-3p	0.81511	0.401184	0.4139269	9.55E-89
8	ABI2	EGOT	hsa-miR-200c-3p	0.81511	0.415847	0.399263465	4.59E-81
9	ABI2	EGOT	hsa-miR-205-5p	0.81511	0.530219	0.284891767	4.24E-50
10	ABI2	FAM66C	hsa-miR-200b-3p	0.819644	0.544121	0.275523135	9.55E-89
11	ABI2	FAM66C	hsa-miR-200c-3p	0.819644	0.546277	0.273367104	4.59E-81
12	ABI2	PART1	hsa-miR-200b-3p	0.774807	0.127118	0.647689599	9.55E-89
13	ABI2	PART1	hsa-miR-200c-3p	0.774807	0.188392	0.586415039	4.59E-81
14	ABI2	PART1	hsa-miR-205-5p	0.774807	0.491132	0.283675212	4.24E-50
15	ABI2	PART1	hsa-miR-429	0.774807	0.437099	0.337707791	4.59E-17
16	ABI2	PVT1	hsa-miR-200b-3p	0.811366	0.271848	0.539518214	9.55E-89

At the end of module 2, you will obtain the ceRNA interactions network



A link occurs between two nodes (ceRNAs) if they:
1) show high values of the Pearson correlation between their expression profiles
2) show high values of the sensitivity correlation
3) [optional] share the binding site for the miRNA mediating their interaction