HW3

Davide Mascolo - Andrea Conti - Mario Napoli

21 gennaio 2022

Point 2

Graph Linkages Test

```
set.seed(12345)
p <- 50; n <- 1000

## Test edge
D    <- matrix(0, p, p)

## H0: F = {(8, 4)}, and A[F] = 0
D[8, 4] <- 1

## Generate a random lower triangular adjacency matrix
A       <- matrix(0, p, p)
A[, 3]  <- sign(runif(p, min = -1, max = 1))
A[3, 3] <- 0

## Data matrix
X <- matrix(rnorm(n*p), n, p) %*% t(solve(diag(p) - A))

## Sigma Function
sigma_hat <- function(A, X){
  ## Input:
  ##    - A: adjacency matrix
  ##    - X: data matrix
  ## Output: sigma estimate

  n <- nrow(X)
  p <- ncol(X)
  sigma <- 0

  ## The next lines compute the formula in the paper
  for (j in 1:p){
    res2 <- 0
    for (i in 1:n){
      res <- 0
      for (k in 1:p){
        if (k != j){
          res <- res + (X[i,k] * A[j,k])
        }
      }
      res2 <- res2 + ((X[i,j] - res)^2)
    }
    sigma <- sigma + res2
  }
  sigma <- sigma * (n*p)^-1
  return (sigma)
}

## Log-Likelihood function
log_lk <- function(A, X, sigma){
  ## Input:
  ##    - A: adjacency matrix
  ##    - X: data matrix
  ##    - sigma: sigma estimate from the previous function
  ## Output: Log-likelihood function

  n <- nrow(X)
  p <- ncol(X)
  sig <- 0
  ## The next lines compute the formula in the paper
  for (j in 1:p){
    res2 <- 0
    for (i in 1:n){
      res <- 0
      for (k in 1:p){
        if (k != j){
          res <- res + (X[i,k] * A[j,k])
        }
      }
```

```
      res2 <- res2 + ((X[i,j] - res)^2)
    }
    sig <- sig + (1/(2*sigma)) * res2 +
      (n/2) * log(sigma)
  }
  return (-sig)
}

## Split Data
split_x <- function(D, A, X, t, mu = 1, f = NULL){
  ## Input:
  ##    - D: is the matrix representation of the set F
  ##    - A: adjacency matrix
  ##    - X: data matrix
  ##    -mu: sparsity parameter
  ##    -t : type of test (Linkages or Pathway)
  ## Output: Reject H0 or Not Reject H0 according to Wn
  if(is.null(X)){
    X_train <- NULL
    X_test <- NULL
  }else{
    idx_train <- sample(1:nrow(X) , size = round(nrow(X)/2),
                        replace = FALSE)
    X_train <- X[idx_train, ]
    X_test  <- X[-idx_train, ]
  }
  if (t == 'link'){
    Un   <- graph_linkages(D, A, X_train, X_test, mu)
    Un_s <- graph_linkages(D, A, X_test, X_train, mu)
  }
  else{
    Un   <- dir_pathway(D, A, f, X_train, X_test, mu)
    Un_s <- dir_pathway(D, A, f, X_test, X_train, mu)
  }
  Wn   <- (Un + Un_s)/2
  alpha <- .05
  t   <- -log(alpha)
  res <- ifelse(Wn > t, 'Reject H0', 'Not-Reject H0')
  return(res)
}

## Graph Linkages
graph_linkages <- function(D, A, X_train = NULL,
                           X_test = NULL, mu = 1){
  ## Input:
  ##    - D: is the matrix represantation of the set F
  ##    - A: adjacency matrix
  ##    - X_train, X_test: train and test set
  ##    -mu: sparsity parameter
  ## Output: Return Un

  ## If it is false, we are working with real data; else      ## we have to work with random data.
  if (is.null(X_train)){
    X <- matrix(rnorm(n*p), n, p) %*%
      t(solve(diag(p) - A))
    idx_train <- sample(1:nrow(X),
                        size = round(nrow(X)/2),
                        replace = FALSE)
    X_train <- X[idx_train, ]
    X_test  <- X[-idx_train, ]
  }

  ## Estimate Graphs
  out_train  <- MLEdag(X = X_train, D = D, tau = 0.3,
                       mu = mu, rho = 1.2,
                       trace_obj = FALSE)
  out_test   <- MLEdag(X = X_test, D = D, tau = 0.3,
                       mu = mu, rho = 1.2,
                       trace_obj = FALSE)

  ## Estimate sigma_0 and sigma_1
  sigma2_0 <- sigma_hat(out_train$A.H0, X_train)
  sigma2_1 <- sigma_hat(out_test$A.H1, X_test)

  ## Compute Likelihoods under the two hypotheses
  lk_H0 <- log_lk(out_train$A.H0, X_train, sigma2_0)
  lk_H1 <- log_lk(out_test$A.H1, X_train, sigma2_1)

  ## Constrained Likelihood Ratio Statistics
```

```
  Un      <- lk_H1 - lk_H0
  return(Un)
}
```

Directed Pathway Test

```
## Test edge
D    <- matrix(0, p, p)

## Set of edges
f <- list(c(8, 4), c(4, 7), c(7, 5), c(5, 10), c(10, 12))

## H0: F = f, and A[jk, jk+1] = 0 for some (jk, jk+1) in F
D[8, 4]   <- 1
D[4, 7]   <- 1
D[7, 5]   <- 1
D[5, 10]  <- 1
D[10, 12] <- 1

## Generate a random lower triangular adjacency matrix
A       <- matrix(0, p, p)
A[, 3]  <- sign(runif(p, min = -1, max = 1))
A[3, 3] <- 0

## Directed Pathway
dir_pathway <- function(D, A, f, X_train = NULL,
                        X_test = NULL, mu = 1){
  ## Input:
  ##    - D: is the matrix represantation of the set F        ##    - A: adjacency matrix
  ##    - f: is the list of edges to be tested
  ##    - X_train, X_test: train and test set
  ##    -mu: sparsity parameter
  ## Output: Return Un

  ## If it is false, we are working with real data; else      ## we have to work with random data.
  if (is.null(X_train)){
    X <- matrix(rnorm(n*p), n, p) %*%
      t(solve(diag(p) - A))
    idx_train <- sample(1:nrow(X) ,
                        size = round(nrow(X)/2),
                    replace = FALSE)
    X_train <- X[idx_train, ]
    X_test  <- X[-idx_train, ]

  }

  ## Estimate Graph
  out_test   <- MLEdag(X = X_test, D = D, tau = 0.3,
                       mu = mu, rho = 1.2,
                       trace_obj = FALSE)

  ## Estimate sigma_1 and compute the log-likelihood under    ## the alternative hypothesis
  sigma2_1 <- sigma_hat(out_test$A.H1, X_train)
  lk_H1 <- log_lk(out_test$A.H1, X_train, sigma2_1)

  ## Estimate sigma_0 and compute the log-likelihood under    ## the null hypothesis
  res <- rep(NA, length(f))
  for (i in 1:length(f)){
    ## We pick the i-th edge and we assign it to 0 in          ## order to test it
    D[f[i][[1]][1], f[i][[1]][2]] <- 0
    ## Estimate Graph
    out_train  <- MLEdag(X = X_train, D = D,
                         tau = 0.3, mu = 1, rho = 1.2,
                         trace_obj = FALSE)

    sigma2_0 <- sigma_hat(out_train$A.H0, X_train)
    res[i]   <- log_lk(out_train$A.H0, X_train, sigma2_0)
    ## We reassign it to 1 as it was originally
    D[f[i][[1]][1], f[i][[1]][2]] <- 1
  }

  ## Constrained Likelihood Ratio Statistics
  Un      <- lk_H1 - max(res)
  return(Un)
}
```

Point 3

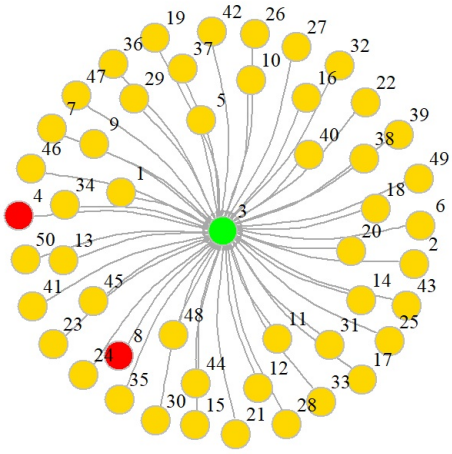
Simulation study to check size and power for linkage test

Test Size: $P(Reject\ H_0|H_0\ true) = \alpha$ $P(Reject\ H_0\,|\,H_0\ true) = \alpha$

```
## Simulation under H0 ---> A[8, 4] = 0

M <- 1000
w_res <- c(NA, M)
for (i in 1:M){
  w_res[i] <- split_x(D, A, NULL, 'link')
}
size <- sum(w_res == 'Reject H0')/M
cat(size)
```

```
0.042
```

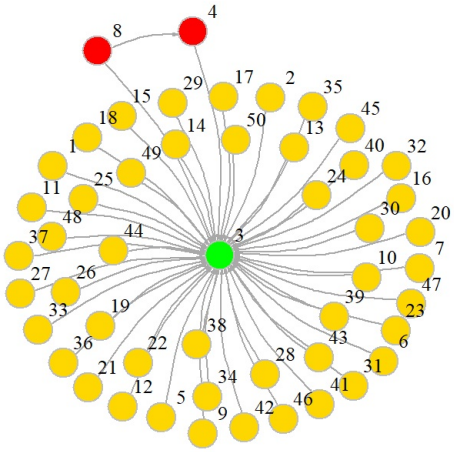


Test Power: $P(Reject\ H_0|H_0\ false) = 1 - \beta$ $P(Reject\ H_0\,|\,H_0 false) = 1 - \beta$

```
##                 H0 ---> A[8, 4] = 0
## Simulation H0 false ---> A[8, 4] = 1

A[8, 4] <- 1
w_res <- c(NA, M)
for (i in 1:M){
  w_res[i] <- split_x(D, A, NULL, 'link')
}
power <- sum(w_res == 'Reject H0')/M
cat(power)
```

```
0.995
```



Point 4

Linkage-Type Hypotheses

1. **Is that edge/relationship really reversed?** Let *F* be an index set where an index $(j,k) \in F$ $(j,k) \in F$ represents a reversed connection. We are interested in testing: $H_0 : A[j,k] = 1 \wedge A[k,j] = 0$ $H_0 : A[j,k] = 1 \wedge A[k,j] = 0$ vs H_1 $: H_1 :$ not $H_0 H_0$
2. **That missing edge is really missing?** Let *F* be an index set where an index $(j,k) \in F$ $(j,k) \in F$ represents a missed connection. We are interested in testing: $H_0 : A[j,k] = 0 H_0 : A[j,k] = 0$ vs $H_1 : A[j,k] = 1 H_1 : A[j,k] = 1$
3. **That present edge is really there?** Let *F* be an index set where an index $(j,k) \in F$ $(j,k) \in F$ represents a non missed connection. We are interested in testing: $H_0 : A[j,k] = 1 H_0 : A[j,k] = 1$ vs $H_1 : A[j,k] = 0 H_1 : A[j,k] = 0$

We have chosen these hypotheses because if our result will be discordant with the paper outcome, we can think that the latter may be an anomaly; instead if our result will be coherent with the paper outcome, it would be an interesting starting point for future works. For example: 1. With the first hypothesis, we want to know if the edge between *Plc* and *PIP3* is actually reversed; 2. With the second one, we want to know if the edge between *PIP3* and *Akt* is really missing; 3. With the third one, we want to know if the edge between *Erk* and *Akt* is really there because if it is so, this could be a starting point "to promote" this edge from reported to expected.

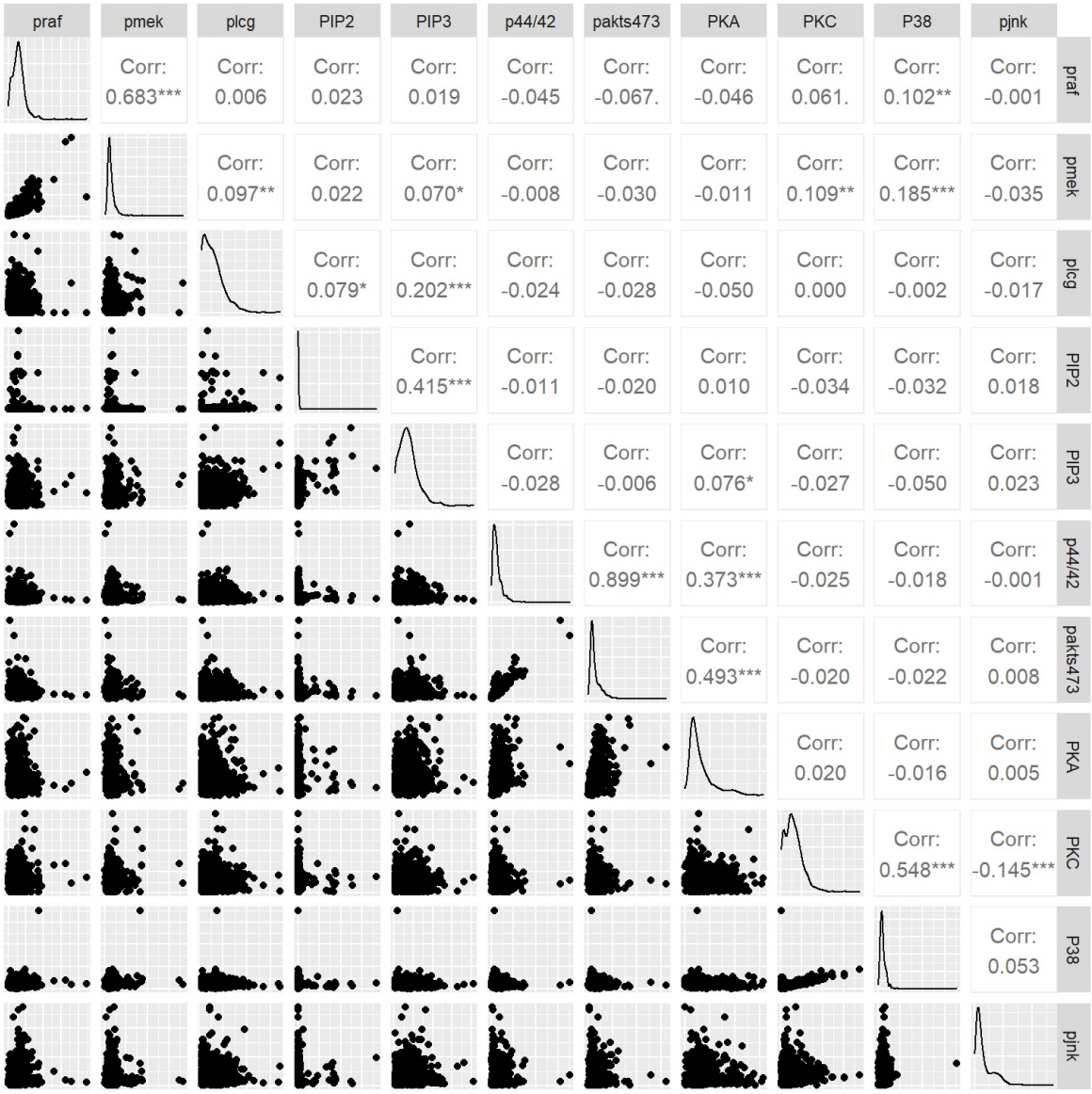
Pathway-Type Hypotheses

1. **A certain path that exists, is there really?** Let *F* be an index of size $|F| \, |F|$ where a common segment is shared by any two consecutive indices, like $F = \{(j_1, j_2), (j_2, j_3), \ldots, (j_{|F|-1}, j_{|F|})\}$ $F = \{(j_1,j_2), (j_2,j_3), \ldots, (j_{|F|-1},j_{|F|})\}$. We are interested in testing: $H_0 : A[j,k] = 1 \quad \forall (j,k) \in F$ $H_0 : A[j,k] = 1 \quad \forall (j,k) \in F$ vs $H_1 : A[j,k] = 0 H_1 : A[j,k] = 0$ for some $(j,k) \in F$ $(j,k) \in F$

The path we have decided to test is *PKC-PKA-Akt*, which appears more interesting than the other paths because its edges are not all expected, compared to others; so, it might be more interesting to know what happens in a path whose some edges are reported and not expected.

Point 5

Let's start visualizing the main descriptive characteristics of the data.



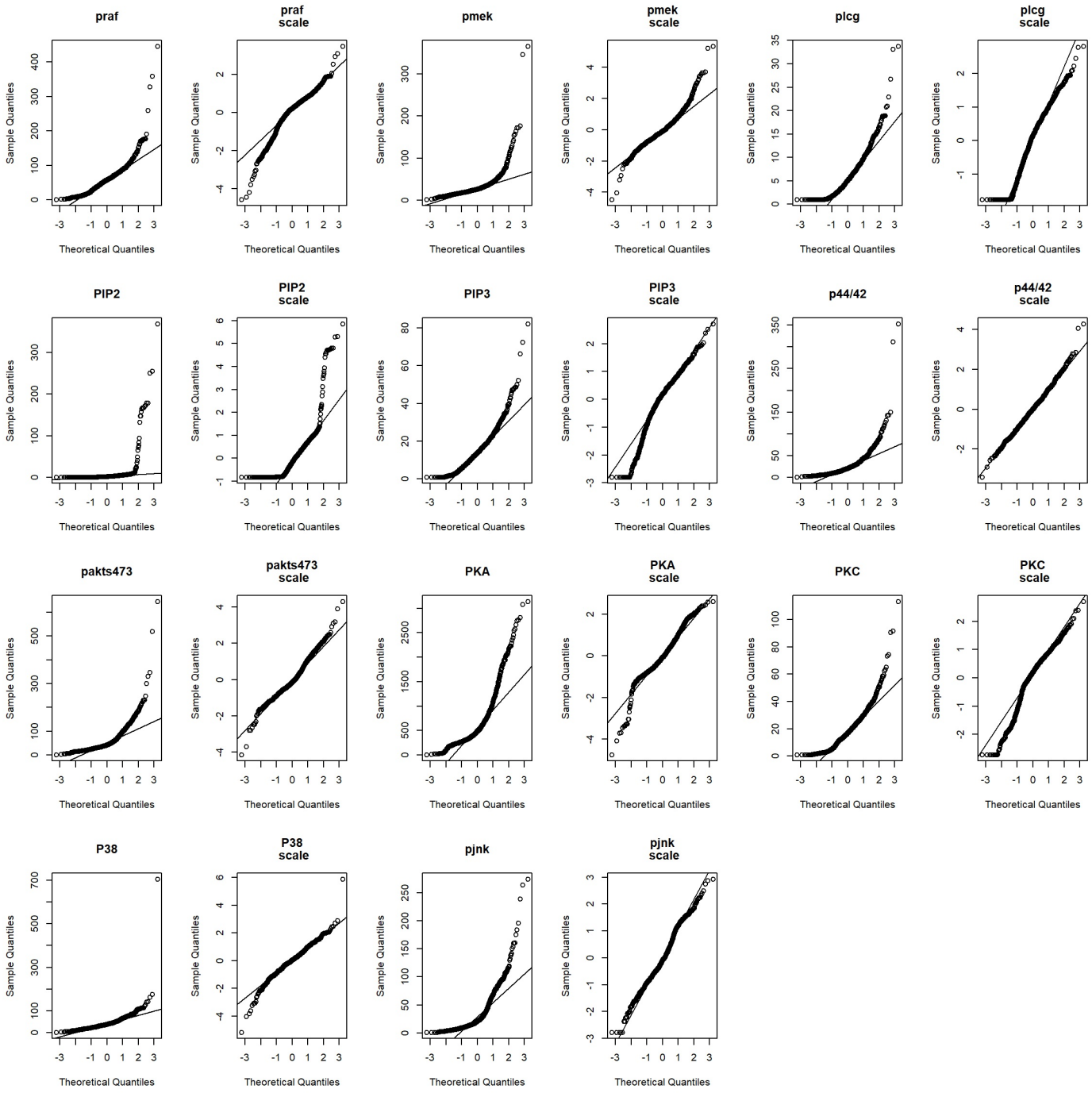

```
## Normalize data
normalize_data <- function(dat){
  ## Input: original data loaded from excel file
  ## Output: standardized data

  b <- boxcox(lm(as.matrix(dat) ~ 1), plotit = F)
  lambda <- b$x[which.max(b$y)]

  ## Normalize Data
  dat_b <- (dat ^ lambda - 1) / lambda

  ## Scale
  dat_scale <- scale(dat_b)
  dat_scale <- as.data.frame(dat_scale)
  dat_scale           <- as.matrix(dat_scale)
  colnames(dat_scale) <- NULL
  return(dat_scale)
}
```

Let's take a look at the differences between originals and standardized variables.



Implementation of the tests for our hypotheses

Linkage-Type Hypotheses

1.Is that edge/relationship really reversed?

```
## Reject H0     --> literature is right
## Not Reject H0 --> paper is right
## Plc ---> PIP3 (3 ---> 5)

test_gl1 <- function(dat_scale, mu, alpha = 0.05){
  ## Input:
  ##  - dat_scale: standardized data
  ##  - mu: sparsity parameter
  ##  - alpha: level of significance
  ## Output: Reject H0 or not Reject H0

  p    <- ncol(dat_scale)
  D    <- matrix(0, p, p)

  ## H0: F = {(3, 5), (5, 3)}, and A[F] = 0
  D[3, 5] <- 1
  D[5, 3] <- 1

  A       <- matrix(0, p, p)
  A[3, 5] <- 1
  A[5, 3] <- 0

  ## Test
  res_graph <- rep(NA, length(mu))
  res_MLE   <- rep(NA, length(mu))

  for (i in 1:length(mu)){
    res_graph[i] <- split_x(D, A, dat_scale,
                            t = 'link', mu[i])
    res_MLE[i]   <- ifelse(MLEdag(X = dat_scale, D = D,
                                  tau = 0.3, mu = mu[i],
                                  rho = 1.2,
                                  trace_obj = FALSE)$pval
                   <= alpha, 'Reject H0','Not-Reject H0')
  }
  res <- data.frame(Graph_Linkages = res_graph,
                    MLEdag = res_MLE,
                    row.names = mu)
  return(res)
}
```

2.That missing edge is really missing?

```
## Reject H0     --> literature is right
## Not Reject H0 --> paper is right
## PIP3 ---> Akt (5 ---> 7)

test_gl2 <- function(dat_scale, mu, alpha = 0.05){
  ## Input:
  ##  - dat_scale: standardized data
  ##  - mu: sparsity parameter
  ##  - alpha: level of significance
  ## Output: Reject H0 or not Reject H0

  ## H0: F = {(5, 7)}, and A[F] = 0
  p    <- ncol(dat_scale)
  D    <- matrix(0, p, p)
  D[5, 7] <- 1

  A       <- matrix(0, p, p)
  A[5, 7] <- 0

  ## Test
  res_graph <- rep(NA, length(mu))
  res_MLE   <- rep(NA, length(mu))

  for (i in 1:length(mu)){
    res_graph[i] <- split_x(D, A, dat_scale,
                            t = 'link', mu[i])
    res_MLE[i]   <- ifelse(MLEdag(X = dat_scale, D = D,
                                  tau = 0.3, mu = mu[i],
                                   rho = 1.2,
                                   trace_obj = FALSE)$pval
                   <= alpha, 'Reject H0','Not-Reject H0')
  }
  res <- data.frame(Graph_Linkages = res_graph,
                    MLEdag = res_MLE,
                    row.names = mu)
  return(res)
}
```

3.That missing edge is really missing?

```
## Reject H0     --> literature is right
## Not Reject H0 --> paper is right
## Erk ---> Akt (6 ---> 7)

test_gl3 <- function(dat_scale, mu, alpha = 0.05){
  ## Input:
  ##  - dat_scale: standardized data
  ##  - mu: sparsity parameter
  ##  - alpha: level of significance
  ## Output: Reject H0 or not Reject H0

  ## H0: F = {(6, 7)}, and A[F] = 0
  p    <- ncol(dat_scale)
  D    <- matrix(0, p, p)
  D[6, 7] <- 1

  A       <- matrix(0, p, p)
  A[6, 7] <- 1

  ## Test
  res_graph <- rep(NA, length(mu))
  res_MLE   <- rep(NA, length(mu))

  for (i in 1:length(mu)){
    res_graph[i] <- split_x(D, A, dat_scale,
                            t = 'link', mu[i])
    res_MLE[i]   <- ifelse(MLEdag(X = dat_scale, D = D,
                                  tau = 0.3, mu = mu[i],
                                   rho = 1.2,
                                   trace_obj = FALSE)$pval
                   <= alpha, 'Reject H0','Not-Reject H0')
  }
  res <- data.frame(Graph_Linkages = res_graph,
                    MLEdag = res_MLE,
                    row.names = mu)
  return(res)
}
```

Pathway-Type Hypotheses

1.A certain path that exists, is there really?

```
## Reject H0     --> literature is right
## Not Reject H0 --> paper is right
## PKC ---> PKA ---> Akt(9 ---> 8 ---> 7)

test_dp <- function(dat_scale, mu, alpha = 0.05){
  ## Input:
  ##  - dat_scale: standardized data
  ##  - mu: sparsity parameter
  ##  - alpha: level of significance
  ## Output: Reject H0 or not Reject H0

  ## H0: F = {(9, 8), (8, 7)}, and A[F] = 0
  p    <- ncol(dat_scale)
  f <- list(c(9, 8), c(8, 7))
  D    <- matrix(0, p, p)
  D[9, 8] <- 1
  D[8, 7] <- 1

  A       <- matrix(0, p, p)
  A[9, 8] <- 1
  A[8, 7] <- 1

  ## Test
  res_dir   <- rep(NA, length(mu))
  res_MLE   <- rep(NA, length(mu))

  for (i in 1:length(mu)){
    res_dir[i] <- split_x(D, A, dat_scale,
                          t = 'path', mu[i], f)

    res_MLE[i]   <- ifelse(MLEdag(X = dat_scale, D = D,
                                  tau = 0.3, mu = mu[i],
                                   rho = 1.2,
                                   trace_obj = FALSE)$pval
                    <= alpha, 'Reject H0','Not-Reject H0')
  }
  res <- data.frame(Directed_Pathway = res_dir,
                    MLEdag = res_MLE,
                    row.names = mu)
  return(res)
}
```

Now we perform tests only on the first sheet of data

```
----- Graph Linkages 1 -----
     Graph_Linkages    MLEdag
1     Not-Reject H0 Reject H0
10    Not-Reject H0 Reject H0
100   Not-Reject H0 Reject H0
1000  Not-Reject H0 Reject H0
----- Graph Linkages 2 -----
     Graph_Linkages        MLEdag
1     Not-Reject H0 Not-Reject H0
10    Not-Reject H0 Not-Reject H0
100   Not-Reject H0 Not-Reject H0
1000  Not-Reject H0 Not-Reject H0
----- Graph Linkages 3 -----
     Graph_Linkages    MLEdag
1         Reject H0 Reject H0
10        Reject H0 Reject H0
100       Reject H0 Reject H0
1000      Reject H0 Reject H0
----- Directed Pathway -----
     Directed_Pathway    MLEdag
1           Reject H0 Reject H0
10          Reject H0 Reject H0
100         Reject H0 Reject H0
1000        Reject H0 Reject H0
```

We can see that MLEdag and Universal Hypothesis Test results are different in Graph Linkages 1; this could be due to the different assumptions behind the approaches and the small quantity of data (only one sheet).

Point 6

Now, starting from the entire dataset, we repeat the previous tests with the following values:

1. $\mu\mu$ = (1, 10, 100, 1000)
2. $\alpha\alpha$ = 0.05

```
----- Graph Linkages 1 -----
     Graph_Linkages        MLEdag
1     Not-Reject H0 Not-Reject H0
10    Not-Reject H0 Not-Reject H0
100   Not-Reject H0 Not-Reject H0
1000      Reject H0 Not-Reject H0
----- Graph Linkages 2 -----
     Graph_Linkages    MLEdag
1     Not-Reject H0 Reject H0
10    Not-Reject H0 Reject H0
100       Reject H0 Reject H0
1000      Reject H0 Reject H0
----- Graph Linkages 3 -----
     Graph_Linkages    MLEdag
1         Reject H0 Reject H0
10        Reject H0 Reject H0
100       Reject H0 Reject H0
1000      Reject H0 Reject H0
----- Directed Pathway -----
     Directed_Pathway    MLEdag
1       Not-Reject H0 Reject H0
10          Reject H0 Reject H0
100         Reject H0 Reject H0
1000        Reject H0 Reject H0
```

We can see that in Graph Linkages, augmenting the quantity of data (all sheets) MLEDag converges to our results, with mu = 1000 we can also see that our test reject the null, this could be due to the fact that the bigger is mu the bigger is the penalty that pushes to zero the edges values in the adjacency matrix. In Graph Linkages 2 we can see that MLEDag changed its response from the previous point, probably even in this case this change is due to the augmenting of the size of the data; even for our test we have a similar behavior to the one in GL 1 related to the increase of mu. In the direct pathway test there has been a change in the output of Universal Test Hypothesis for mu = 1, this could be due to the small mu value.

Adjust for multiplicity

Suppose the split LRT failed to reject the null. Then we are allowed to collect more data and update the test statistic, and check if the updated statistic crosses $1/\alpha\alpha$. If it does not, we can further collect more data and reupdate the statistic, and this process can be repeated indefinitely. Importantly we do not need any correction for repeated testing; this is primarily because the statistic is upper bounded by a nonnegative martingale. For more info about, click section 8 (https://arxiv.org/pdf/1912.11436.pdf):

Causal relations

In the biological sciences, and especially biomedical science, causality is typically reduced to those molecular and cellular mechanisms that can be isolated in the laboratory and thence manipulated experimentally. Experimental perturbations are commonly used to establish causal relationships between the molecular components of a pathway and their cellular functions; however, this approach suffers inherent limitations. Especially in pathways with a significant level of nonlinearity and redundancy among components, such perturbations induce compensatory responses that obscure the actual function of the targeted component in the unperturbed pathway. A complementary approach uses constitutive fluctuations in component activities to identify the hierarchy of information flow through pathways. A major goal of cell biology is to determine how a network of highly interconnected, context-dependent pathways connects the activity of specific molecules to cellular processes. The complexity of the pathway networks can make it difficult to determine the roles individual pathway components play: they may contribute to many different cell functions or they may have no obvious function at all. Causality in cellular systems is not as well defined. One particular difficulty is that we are interested in the causality of molecular events that are not necessarily connected by linear pathways but more complex topologies where cause-and-effect relations can be obscured by pathway features such as compensation and feedback. Moreover, quite often we are not even concerned with cause-and-effect relations between pathway components, but instead in the specific contribution a pathway component makes to the cellular outputs conferred by the pathway. For more info, click here (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4255263/).

We discussed about this homework with Cruoglio Antonella and Iovino Giuliana.