

# Appendix-1.R

SSosa

2023-12-18

```
# Function from :  
# https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.13400  
library(ANTs)  
library(ggplot2)  
library(ggpubr)  
make_network <- function(obs, focal.id) {  
  N <- ncol(obs)  
  network <- matrix(0, nrow = N, ncol = N)  
  for (i in 1:(N - 1)) {  
    for (j in (i + 1):N) {  
      xab <- sum(obs[which(focal.id %in% c(i, j)), c(i, j)] > 0)  
      fa <- sum(focal.id == i)  
      fb <- sum(focal.id == j)  
      ya <- fa - xab  
      yb <- fb - xab  
      if (yb < 0) {  
        yb = 0  
      }  
      if (ya < 0) {  
        ya = 0  
      }  
      sri <- ((xab) / (xab + ya + yb))  
  
      if (!is.nan(sri)) {  
        network[i, j] <- sri  
        network[j, i] <- sri  
      } else {  
        network[i, j] <- 0  
        network[j, i] <- 0  
      }  
    }  
  }  
  return(network)  
}  
  
make_network.corrected <- function(obs, focal.id) {  
  N <- ncol(obs)  
  network <- matrix(0, nrow = N, ncol = N) #SRI  
  
  # Calcule SRI  
  for (i in 1:(N - 1)) {  
    for (j in (i + 1):N) {
```

```

xab <- sum(obs[which(focal.id %in% c(i, j)), c(i, j)] > 0)
fa <- sum(focal.id == i)
fb <- sum(focal.id == j)
ya <- fa - xab
yb <- fb - xab
if (yb < 0) {
  yb = 0
}
if (ya < 0) {
  ya = 0
}
sri <- ((xab) / (xab + ya + yb))
gi = fa + fb
if (!is.nan(sri)) {
  network[i, j] <- sri
  network[j, i] <- sri
} else{
  network[i, j] <- 0
  network[j, i] <- 0
}
}
}

# Calcule Bias d'observations
obs.per.ind.Bias = rep(0, nrow(network))
for (x in 1:ncol(network)) {
  obs.per.ind.Bias[x] = length(which(focal.id %in% x))
}
mean.obs = mean(obs.per.ind.Bias, na.rm = T)
dif.mean = mean.obs - obs.per.ind.Bias
dif.mean.obs = abs(dif.mean)
#dif.mean = abs(dif.mean.obs)

# Calcule Nombre d'interactions
int = colSums(obs)

# définition de dif.mean
#dif.mean = dif.mean/dif.mean.obs
dif.mean = dif.mean.obs # similar false negatives rates

# CReation matrice en fonction de la définition de dif.mean
fa = matrix(0, ncol = nrow(network), nrow = nrow(network))
for (a in 1:nrow(network)) {
  fa[a, ] = dif.mean[a]
}

fb = t(fa)

sampling.effort = (fa + fb)
diag(sampling.effort) = 0

```

```

# Formule de correction
GI = (network / (sd(int) / sampling.effort))
return(GI)
}

error.rates <- function(R, p.side = "two.side") {
  if (p.side == "two.side") {
    d1 = data.frame(
      "Biases" = rep(TRUE, 2),
      "Error Type" = rep("False negatives rates", 2),
      "GI" = rep(FALSE, 2),
      "two.side" = rep(TRUE, 2),
      "approches" = c("Parametric", "Nertwork permutations"),
      "strength" = c(
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Strength.parametric", ]$p_value_two_side > 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == T &
          R$test %in% "Strength.parametric", ]),
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Strength.network", ]$p_value_two_side > 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == T &
          R$test %in% "Strength.network", ]),
      ),
      "eigenvector" = c(
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Eigen.parametric", ]$p_value_two_side > 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == T &
          R$test %in% "Eigen.parametric", ]),
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Eigen.network", ]$p_value_two_side > 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == T &
          R$test %in% "Eigen.network", ]),
      ),
      "Alters" = c(
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Alters.parametric", ]$p_value_two_side > 0.05, na.rm = T) *
        100 / nrow(R[R$FemPhenotypeBias == T &
          R$test %in% "Alters.parametric", ]),
        sum(R[R$FemPhenotypeBias == T &
          R$test %in% "Alters.network", ]$p_value_two_side > 0.05, na.rm = T) * 100 /
        nrow(R[R$FemPhenotypeBias == T & R$test %in% "Alters.network", ]),
      )
    )
  }

  d2 = data.frame(
    "Biases" = rep(TRUE, 2),
    "Error Type" = rep("False negatives rates", 2),
    "GI" = rep(TRUE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Nertwork permutations"),
    "strength" = c(

```

```

sum(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.parametric.corrected", ]$p_value_two_side > 0.05) *
100 / nrow(R[R$FemPhenotypeBias == T &
              R$test %in% "Strength.parametric.corrected", ]),
sum(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.network.corrected", ]$p_value_two_side > 0.05) *
100 / nrow(R[R$FemPhenotypeBias == T &
              R$test %in% "Strength.network.corrected", ])
),
"eigenvector" = c(
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric.corrected", ]$p_value_two_side > 0.05) *
100 / nrow(R[R$FemPhenotypeBias == T &
              R$test %in% "Eigen.parametric.corrected", ]),
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network.corrected", ]$p_value_two_side > 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network.corrected", ])
),
"Alters" = c(
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric.corrected", ]$p_value_two_side > 0.05, na.rm = T) *
100 / nrow(R[R$FemPhenotypeBias == T &
              R$test %in% "Alters.parametric.corrected", ]),
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network.corrected", ]$p_value_two_side > 0.05, na.rm = T) *
100 / nrow(R[R$FemPhenotypeBias == T &
              R$test %in% "Alters.network.corrected", ])
)
)
)

d3 = data.frame(
  "Error Type" = rep("False positives rates", 2),
  "Biases" = rep(TRUE, 2),
  "GI" = rep(FALSE, 2),
  "two.side" = rep(TRUE, 2),
  "approches" = c("Parametric", "Nertwork permutations"),
  "strength" = c(
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Strength.parametric", ]$p_value_two_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Strength.parametric", ]),
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Strength.network", ]$p_value_two_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Strength.network", ])
  ),
  "eigenvector" = c(
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Eigen.parametric", ]$p_value_two_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Eigen.parametric", ]),
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Eigen.network", ]$p_value_two_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Eigen.network", ])
  )
)

```

```

    sum(R[R$FemPhenotypeBias == F &
        R$test %in% "Eigen.network", ]$p_value_two_side < 0.05) * 100 /
    nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Eigen.network", ])
),
"Alters" = c(
    sum(R[R$FemPhenotypeBias == F &
        R$test %in% "Alters.parametric", ]$p_value_two_side < 0.05, na.rm = T) *
    100 / nrow(R[R$FemPhenotypeBias == F &
        R$test %in% "Alters.parametric", ]),
    sum(R[R$FemPhenotypeBias == F &
        R$test %in% "Alters.network", ]$p_value_two_side < 0.05, na.rm = T) * 100 /
    nrow(R[R$FemPhenotypeBias == F & R$test %in% "Alters.network", ])
)
)

d4 = data.frame(
    "Error Type" = rep("False positives rates", 2),
    "Biases" = rep(TRUE, 2),
    "GI" = rep(TRUE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Nertwork permutations"),
    "strength" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.parametric.corrected", ]$p_value_two_side < 0.05) *
        100 / nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.network.corrected", ]$p_value_two_side < 0.05) *
        100 / nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.network.corrected", ])
    ),
    "eigenvector" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.parametric.corrected", ]$p_value_two_side < 0.05) *
        100 / nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.network.corrected", ]$p_value_two_side < 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.network.corrected", ])
    ),
    "Alters" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.parametric.corrected3", ]$p_value_two_side < 0.05, na.rm = T) *
        100 / nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.network.corrected", ]$p_value_two_side < 0.05, na.rm = T) *
        100 / nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.network.corrected", ])
    )
)

```

```

)
}
if (p.side == "righthand") {
  d1 = data.frame(
    "Biases" = rep(TRUE, 2),
    "Error Type" = rep("False negatives rates", 2),
    "GI" = rep(FALSE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Network permutations"),
    "strength" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.parametric", ]$p_value_righthand > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network", ]$p_value_righthand > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network", ]))
    ),
    "eigenvector" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric", ]$p_value_righthand > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network", ]$p_value_righthand > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network", ]))
    ),
    "Alters" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric", ]$p_value_righthand > 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network", ]$p_value_righthand > 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network", ]))
    )
  )
}

d2 = data.frame(
  "Biases" = rep(TRUE, 2),
  "Error Type" = rep("False negatives rates", 2),
  "GI" = rep(TRUE, 2),
  "two.side" = rep(TRUE, 2),
  "approches" = c("Parametric", "Network permutations"),
  "strength" = c(
    sum(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.parametric.corrected", ]$p_value_righthand > 0.05) *
    100 / nrow(R[R$FemPhenotypeBias == T &

```

```

        R$test %in% "Strength.parametric.corrected", ]),
sum(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.network.corrected", ]$p_value_rigth_side > 0.05) *
100 / nrow(R[R$FemPhenotypeBias == T &
             R$test %in% "Strength.network.corrected", ])
),
"eigenvector" = c(
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric.corrected", ]$p_value_rigth_side > 0.05) *
100 / nrow(R[R$FemPhenotypeBias == T &
             R$test %in% "Eigen.parametric.corrected", ]),
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network.corrected", ]$p_value_rigth_side > 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == T &
      R$test %in% "Eigen.network.corrected", ])
),
"Alters" = c(
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric.corrected", ]$p_value_rigth_side > 0.05, na.rm = T) *
100 / nrow(R[R$FemPhenotypeBias == T &
             R$test %in% "Alters.parametric.corrected", ]),
  sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network.corrected", ]$p_value_rigth_side > 0.05, na.rm = T) *
100 / nrow(R[R$FemPhenotypeBias == T &
             R$test %in% "Alters.network.corrected", ])
)
)

d3 = data.frame(
  "Error Type" = rep("False positives rates", 2),
  "Biases" = rep(TRUE, 2),
  "GI" = rep(FALSE, 2),
  "two.side" = rep(TRUE, 2),
  "approches" = c("Parametric", "Nertwork permutations"),
  "strength" = c(
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Strength.parametric", ]$p_value_rigth_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.parametric", ]),
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Strength.network", ]$p_value_rigth_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.network", ])
  ),
  "eigenvector" = c(
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Eigen.parametric", ]$p_value_rigth_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Eigen.parametric", ]),
    sum(R[R$FemPhenotypeBias == F &
          R$test %in% "Eigen.network", ]$p_value_rigth_side < 0.05) * 100 /
nrow(R[R$FemPhenotypeBias == F &

```

```

        R$test %in% "Eigen.network", ])
    ),
    "Alters" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.parametric", ]$p_value_rigth_side < 0.05, na.rm = T) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Alters.parametric", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.network", ]$p_value_rigth_side < 0.05, na.rm = T) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Alters.network", ])
    )
)

d4 = data.frame(
    "Error Type" = rep("False positives rates", 2),
    "Biases" = rep(TRUE, 2),
    "GI" = rep(TRUE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Nertwork permutations"),
    "strength" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.parametric.corrected", ]$p_value_rigth_side < 0.05) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Strength.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.network.corrected", ]$p_value_rigth_side < 0.05) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Strength.network.corrected", ])
    ),
    "eigenvector" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.parametric.corrected", ]$p_value_rigth_side < 0.05) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Eigen.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.network.corrected", ]$p_value_rigth_side < 0.05) * 100 /
            nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Eigen.network.corrected", ])
    ),
    "Alters" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.parametric.corrected3", ]$p_value_rigth_side < 0.05, na.rm = T) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Alters.parametric.corrected", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Alters.network.corrected", ]$p_value_rigth_side < 0.05, na.rm = T) *
            100 / nrow(R[R$FemPhenotypeBias == F &
                R$test %in% "Alters.network.corrected", ])
    )
)

```



```

}
if (p.side == "left.side") {
  d1 = data.frame(
    "Biases" = rep(TRUE, 2),
    "Error Type" = rep("False negatives rates", 2),
    "GI" = rep(FALSE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Nertwork permutations"),
    "strength" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.parametric", ]$p_value_left_side > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network", ]$p_value_left_side > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network", ]))
    ),
    "eigenvector" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric", ]$p_value_left_side > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network", ]$p_value_left_side > 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network", ]))
    ),
    "Alters" = c(
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric", ]$p_value_left_side > 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric", ]),
      sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network", ]$p_value_left_side > 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network", ]))
    )
  )
}

d2 = data.frame(
  "Biases" = rep(TRUE, 2),
  "Error Type" = rep("False negatives rates", 2),
  "GI" = rep(TRUE, 2),
  "two.side" = rep(TRUE, 2),
  "approches" = c("Parametric", "Nertwork permutations"),
  "strength" = c(
    sum(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.parametric.corrected", ]$p_value_left_side > 0.05) *
    100 / nrow(R[R$FemPhenotypeBias == T &
      R$test %in% "Strength.parametric.corrected", ]),

```

```

    sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network.corrected", ]$p_value_left_side > 0.05) *
    100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Strength.network.corrected", ])
),
"eigenvector" = c(
    sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric.corrected", ]$p_value_left_side > 0.05) *
    100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.parametric.corrected", ]),
    sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network.corrected", ]$p_value_left_side > 0.05) * 100 /
    nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Eigen.network.corrected", ])
),
"Alters" = c(
    sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric.corrected", ]$p_value_left_side > 0.05, na.rm = T) *
    100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.parametric.corrected", ]),
    sum(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network.corrected", ]$p_value_left_side > 0.05, na.rm = T) *
    100 / nrow(R[R$FemPhenotypeBias == T &
        R$test %in% "Alters.network.corrected", ])
)
)

d3 = data.frame(
    "Error Type" = rep("False positives rates", 2),
    "Biases" = rep(TRUE, 2),
    "GI" = rep(FALSE, 2),
    "two.side" = rep(TRUE, 2),
    "approches" = c("Parametric", "Nertwork permutations"),
    "strength" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.parametric", ]$p_value_left_side < 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.parametric", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.network", ]$p_value_left_side < 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Strength.network", ])
    ),
    "eigenvector" = c(
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.parametric", ]$p_value_left_side < 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.parametric", ]),
        sum(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.network", ]$p_value_left_side < 0.05) * 100 /
        nrow(R[R$FemPhenotypeBias == F &
            R$test %in% "Eigen.network", ])
    )
)

```

```

),
"Alters" = c(
  sum(R[R$FemPhenotypeBias == F &
    R$test %in% "Alters.parametric", ]$p_value_left_side < 0.05, na.rm = T) *
    100 / nrow(R[R$FemPhenotypeBias == F &
    R$test %in% "Alters.parametric", ]),
  sum(R[R$FemPhenotypeBias == F &
    R$test %in% "Alters.network", ]$p_value_left_side < 0.05, na.rm = T) *
    100 / nrow(R[R$FemPhenotypeBias == F &
    R$test %in% "Alters.network", ]))
)
)

d4 = data.frame(
  "Error Type" = rep("False positives rates", 2),
  "Biases" = rep(TRUE, 2),
  "GI" = rep(TRUE, 2),
  "two.side" = rep(TRUE, 2),
  "approches" = c("Parametric", "Nertwork permutations"),
  "strength" = c(
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.parametric.corrected", ]$p_value_left_side < 0.05) *
      100 / nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.parametric.corrected", ]),
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.network.corrected", ]$p_value_left_side < 0.05) *
      100 / nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Strength.network.corrected", ]))
  ),
  "eigenvector" = c(
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Eigen.parametric.corrected", ]$p_value_left_side < 0.05) *
      100 / nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Eigen.parametric.corrected", ]),
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Eigen.network.corrected", ]$p_value_left_side < 0.05) * 100 /
      nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Eigen.network.corrected", ]))
  ),
  "Alters" = c(
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Alters.parametric.corrected3", ]$p_value_left_side < 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Alters.parametric.corrected", ]),
    sum(R[R$FemPhenotypeBias == F &
      R$test %in% "Alters.network.corrected", ]$p_value_left_side < 0.05, na.rm = T) *
      100 / nrow(R[R$FemPhenotypeBias == F &
      R$test %in% "Alters.network.corrected", ]))
  )
)
}

```

```

RESULTS = rbind(d1, d2, d3, d4)
return(RESULTS)
}

#####
### function to generate pre-network permutations (swaps of individuals between focals)
rand_network <- function(obs.p, focal.id, n.perm, n_focals) {
  N <- ncol(obs.p)
  networks_rand <- array(0, c(n.perm, N, N))
  for (i in 1:n.perm) {
    # first randomly select two focal observations
    repeat {
      o <- 1:n_focals
      a <- sample(o, 1)
      b <- sample(o[-a], 1)

      # check if these are different individuals and they have associates
      if ((focal.id[a] != focal.id[b]) &
          (sum(obs.p[a, ]) > 0) & (sum(obs.p[b, ]) > 0)) {
        # next select two associates to swap
        d <- sample(which(obs.p[a, ] > 0), 1)
        e <- sample(which(obs.p[b, ] > 0), 1)

        # check they do not occur in the other focal
        if ((obs.p[a, e] == 0) & obs.p[b, d] == 0) {
          # now check we have 4 distinct individuals, otherwise repeat this process
          if (!(d %in% c(focal.id[a], focal.id[b], e)) &
              !(e %in% c(focal.id[a], focal.id[b], d))) {
            break
          }
        }
      }
    }
  }

  # swap individuals
  obs.p[a, d] <- 0
  obs.p[b, d] <- 1
  obs.p[b, e] <- 0
  obs.p[a, e] <- 1
  # caculate network
  networks_rand[i, , ] <- make_network(obs.p, focal.id)
}
return(networks_rand)
}

# Data stream with GI
rand_network2 <- function(obs.p, focal.id, n.perm, n_focals) {
  N <- ncol(obs.p)
  networks_rand <- array(0, c(n.perm, N, N))
  for (i in 1:n.perm) {
    # first randomly select two focal observations
    repeat {
      o <- 1:n_focals

```

```

a <- sample(o, 1)
b <- sample(o[-a], 1)

# check if these are different individuals and they have associates
if ((focal.id[a] != focal.id[b]) &
    (sum(obs.p[a, ]) > 0) & (sum(obs.p[b, ]) > 0)) {
  # next select two associates to swap
  d <- sample(which(obs.p[a, ] > 0), 1)
  e <- sample(which(obs.p[b, ] > 0), 1)

  # check they do not occur in the other focal
  if ((obs.p[a, e] == 0) & obs.p[b, d] == 0) {
    # now check we have 4 distinct individuals, otherwise repeat this process
    if (!(d %in% c(focal.id[a], focal.id[b], e)) &
        !(e %in% c(focal.id[a], focal.id[b], d))) {
      break
    }
  }
}

# swap individuals
obs.p[a, d] <- 0
obs.p[b, d] <- 1
obs.p[b, e] <- 0
obs.p[a, e] <- 1
# calculate network
networks_rand[i, , ] <- make_network.corrected(obs.p, focal.id)
}
return(networks_rand)
}

### Function to allocate number of observations to groups

rand_vect <- function(N, M, sd = 1, pos.only = TRUE) {
  vec <- rnorm(N, M / N, sd)
  if (abs(sum(vec)) < 0.01)
    vec <- vec + 1
  vec <- round(vec / sum(vec) * M)
  deviation <- M - sum(vec)
  for (. in seq_len(abs(deviation))) {
    vec[i] <- vec[i <- sample(N, 1)] + sign(deviation)
  }
  if (pos.only)
    while (any(vec < 0)) {
      negs <- vec < 0
      pos <- vec > 0
      vec[negs][i] <- vec[negs][i <- sample(sum(negs), 1)] + 1
      vec[pos][i] <- vec[pos][i <- sample(sum(pos), 1)] - 1
    }
  vec
}

```

### ### MAIN SIMULATION FUNCTION #####

@param GS numeric argument indicating group size @param ObsBias numeric argument indicating the percent of censoring bias [0.5-1.0] @param FemPhenotypeBias boolean argument indicating whether a phenotype bias is present among females @param nfocals numeric argument indicating number of focal samples @param N.perm numeric argument indicating number of permutations

```
Simulation <-  
  function(GS,  
           ObsBias,  
           FemSexRatio,  
           FemPhenotypeBias,  
           nfocals,  
           N.Perm,  
           print = FALSE)  
{  
  # Set parameters  
  N <- GS  
  n_focals <- nfocals  
  # Generate nodes  
  NumFem <- round(GS * FemSexRatio)  
  NumMal <- GS - NumFem  
  Sex <- c(rep("F", NumFem), rep("M", NumMal))  
  Sex <- sample(Sex, GS, replace = F)  
  ids <- data.frame(ID = 1:(N), SEX = Sex)  
  # Generate a distribution of group sizes  
  group_size <- sample(c(1:(N / 2)), n_focals, replace = TRUE)  
  # Create blank observation matrix  
  obs <- matrix(0, nrow = n_focals, ncol = N)  
  ## set number of observations of an individual in a group per individual  
  ids$OBS <- rand_vect(N, sum(group_size), pos.only = TRUE)  
  ## Variables to Allocate individuals to groups,  
  GroupID <- c(1:n_focals)  
  group_size.tmp <- group_size  
  # IF Fem phenotype is stronger than males, start with males so that they end up in smaller groups  
  if (FemPhenotypeBias == T)  
  {  
    which.males <- which(ids$SEX == "M")  
    which.females <- which(ids$SEX == "F")  
    for (i in which.males)  
    {  
      g <- sample(GroupID[which(group_size.tmp > 0)], ids$OBS[i])  
      group_size.tmp[g] <- group_size.tmp[g] - 1  
      obs[g, i] <- 1  
    }  
    for (i in which.females)  
    {  
      if ((sum(group_size.tmp > 0) < ids$OBS[i]))  
      {  
        Needed <- ids$OBS[i] - (sum(group_size.tmp > 0))  
        group.tmp <- group_size  
        group.tmp[group_size.tmp > 0] = 0  
        BiggestGroups <-
```

```

        sort(group.tmp,
              decreasing = T,
              index.return = T)$ix
        ExtraGroups <- BiggestGroups[1:Needed]
        g <- c(GroupID[which(group_size.tmp > 0)], ExtraGroups)
      } else
      {
        g <- sample(GroupID[which(group_size.tmp > 0)], ids$OBS[i])
      }
      group_size.tmp[g] <- group_size.tmp[g] - 1
      obs[g, i] <- 1
    }
  } else
  {
    # IF Fem phenotype is equal to males, allocate individuals to groups at random
    {
      Inds <- c(1:GS)
      for (. in 1:GS)
      {
        id <- Inds[1]
        if (length(Inds) > 1) {
          id <- sample(Inds, 1)
        }
        Inds <- Inds[-which(Inds == id)]
        if ((sum(group_size.tmp > 0) < ids$OBS[id]))
        {
          Needed <- ids$OBS[id] - (sum(group_size.tmp > 0))
          Fullgroups <- which(group_size.tmp == 0)
          ExtraGroups <- sample(Fullgroups, Needed, replace = F)
          g <- c(GroupID[which(group_size.tmp > 0)], ExtraGroups)
        } else
        {
          g <- sample(GroupID[which(group_size.tmp > 0)], ids$OBS[id])
        }
        group_size.tmp[g] <- group_size.tmp[g] - 1
        obs[g, id] <- 1
      }
    }
  }
  # Select a focal individual from each group
  focal.id <- apply(obs, 1, function(x) {
    sample(which(x == 1), 1)
  })

  # Now remove cases where individuals occur in a group for which they are focal
  obs[cbind(1:n_focals, focal.id)] <- 0

  ## NOW DO NETWORK ANALYSIS ON THESE DATA
  # Calculate network
  Net.Ori <- make_network(obs, focal.id)
  Net.Ori.corrected <- make_network.corrected(obs, focal.id)
  Net.Ori.corrected[is.infinite(Net.Ori.corrected)] = 0

  # Remove some observations according to the degree of observation bias ObsBias

```

```

# Generate probability of being observed (males=1,females=ObsBias)
ids$OBS_PROB <- ObsBias
ids$OBS_PROB[which(ids$SEX == "M")] <- 1

# Remove observations from GBI
obs.Bias <- obs
for (i in 1:N) {
  obs.Bias[which(obs.Bias[, i] > 0), i] <-
    sample(
      c(0, 1),
      sum(obs.Bias[, i]),
      replace = TRUE,
      prob = c(1 - ids$OBS_PROB[i], ids$OBS_PROB[i])
    )
}

# Calculate new network
Net.Biais <- make_network(obs.Bias, focal.id)
Net.Biais.corrected <- make_network.corrected(obs.Bias, focal.id)
Net.Biais.corrected[is.infinite(Net.Biais.corrected)] = 0
Net.Biais.corrected[is.nan(Net.Biais.corrected)] = 0

# Calculate Strength
ids$DEGREE <- rowSums(Net.Ori)
ids$DEGREE.Corrected <- rowSums(Net.Ori.corrected)

ids$DEGREE.Bias <- rowSums(Net.Biais)
ids$DEGREE.Bias.Corrected <- rowSums(Net.Biais.corrected)

# Sampling effort for each individuals-----
obs.per.ind.Bias = rep(0, nrow(ids))
for (x in 1:nrow(ids)) {
  obs.per.ind.Bias[x] = length(which(focal.id %in% x))
}
ids$obs.bias = obs.per.ind.Bias
if (print) {
  print(ggplot(ids, aes(
    x = DEGREE, y = DEGREE.Bias, color = SEX
  )) + geom_point())
}

#####
##### Modification 2 (extension): Compute degree and eigenvector
ids$alters <- met.degree(Net.Ori)
ids$alters.Bias <- met.degree(Net.Biais)
ids$alters.Bias.Corrected <- (ids$alters.Bias) / obs.per.ind.Bias
if (any(is.infinite(ids$alters.Bias.Corrected))) {
  ids$alters.Bias.Corrected[which(is.infinite(ids$alters.Bias.Corrected))] = NA
}

ids$eigen <- met.eigen(Net.Ori)
ids$eigen.Bias <- ((met.eigen(Net.Biais)))
ids$eigen.Bias.Corrected <- ((met.eigen(Net.Biais.corrected)))

```



```
#####
##### results visualization
#####
p1 = ggplot(ids, aes(x = SEX, y = DEGREE, group = SEX)) + geom_boxplot() +
  geom_point()
p2 = ggplot(ids, aes(x = SEX, y = DEGREE.Bias, group = SEX)) + geom_boxplot() +
  geom_point()
p3 = ggplot(ids, aes(x = SEX, y = DEGREE.Bias.Corrected, group = SEX)) + geom_boxplot() +
  geom_point()
p4 = ggplot(ids, aes(x = SEX, y = DEGREE.Bias.Corrected2, group = SEX)) + geom_boxplot() +
  geom_point()

p4 = ggplot(ids, aes(x = SEX, y = alters, group = SEX)) + geom_boxplot() +
  geom_point()
p5 = ggplot(ids, aes(x = SEX, y = alters.Bias, group = SEX)) + geom_boxplot() +
  geom_point()
p6 = ggplot(ids, aes(x = SEX, y = alters.Bias.Corrected, group = SEX)) + geom_boxplot() +
  geom_point()

p7 = ggplot(ids, aes(x = SEX, y = eigen, group = SEX)) + geom_boxplot() +
  geom_point()
p8 = ggplot(ids, aes(x = SEX, y = eigen.Bias, group = SEX)) + geom_boxplot() +
  geom_point()
p9 = ggplot(ids, aes(x = SEX, y = eigen.Bias.Corrected, group = SEX)) + geom_boxplot() +
  geom_point()

if (print) {
  print(ggarrange(
    p1,
    p2,
    p3,
    p4,
    p5,
    p6,
    p7,
    p8,
    p9,
    ncol = 3,
    nrow = 3
  ))
}

return(ids)
}

# Testing presence of exposure bias in simulation
N = 100
result = NULL
for (a in 1:N) {
  GS.sample = sample(30:100, 1)
  ObsBias.sample = sample(0.5:0.1) # exposure bias appear when censoring bias is simulated
  sim = Simulation(
```

```

    GS = GS.sample,
    ObsBias = ObsBias.sample,
    FemSexRatio = 0.5 ,
    FemPhenotypeBias = TRUE,
    nfocals = 1000,
    N.Perm = 1
  )
  ggplot(sim, aes(x = SEX, y = obs.bias, group = SEX)) + geom_boxplot() +
    geom_point()
  s = summary(lm(obs.bias ~ SEX, data = sim))$coefficients[2, ]
  result = rbind(result, data.frame(t(s), GS.sample, ObsBias.sample, 'sim' =
    a))
}
exposure.bias = sum(result$Pr...t.. < 0.05) / nrow(result) * 100
cat(exposure.bias, '%')

```

## 100 %