Forms of Redundant Networks

Abraham Iñiguez

2025-06-05

Network Test & Setup:

Load Packages

```
library(qgraph)
library(igraph)
source("00_project_functions.R")
```

True Network Test

Form 1: HCSF

```
# Functions needed
# corr_gen() ~ Obtain true network corr matrix
# cor2pcor() ~ Move from corr matrix to pcor matrix
# hcsf() ~ Obtain redundant corr matrix that is highly correlated & same function


metrics <- array(NA, dim = c(1000, 11, 3),
                 dimnames = list(NULL, c("p1_true", "p2_true", "p3_true",
                 "target", "p1_redun", "p2_redun", "p3_redun", "target_redun",
                 "clone","peripheral_change", "target_change"),
                                 c("strength", "closeness",
                                   "betweenness")))

for (i in 1:1000){
  true_mat <- corr_gen(nv = 4, EF = .5, edge.probability = .9)
  redun_mat <- hcsf(matrix = true_mat)
  redun_subset <- redun_mat[1:4, 1:4]
  redun_pcor_subset <- cor2pcor(redun_subset)
  redun_pcor <- cor2pcor(redun_mat)

  g_subset <- graph_from_adjacency_matrix(redun_pcor_subset,
                                          mode = "undirected",
                                          weighted = TRUE)
  g_full <- graph_from_adjacency_matrix(redun_pcor,
                                        mode = "undirected",
```

```
                                        weighted = TRUE)
 ## Strength ##
  strength_subset <- strength(g_subset)
  metrics[i, 1:4, "strength"] <- strength_subset
  strength_full <- strength(g_full)
  metrics[i, 5:9, "strength"] <- strength_full
  peripheral_change_s <- mean(strength_full[1:3] - strength_subset[1:3])
  target_change_s <- strength_full[4] - strength_subset[4]
  metrics[i, 10:11,"strength"] <- c(peripheral_change_s, target_change_s)

  ## Closeness ##
  closeness_subset <- closeness(g_subset, weights = 1/abs(E(g_subset)$weight))
  metrics[i, 1:4, "closeness"] <- closeness_subset
  closeness_full <- closeness(g_full, weights = 1/abs(E(g_full)$weight))
  metrics[i, 5:9, "closeness"] <- closeness_full
  peripheral_change_c <- mean(closeness_full[1:3] - closeness_subset[1:3])
  target_change_c <- closeness_full[4] - closeness_subset[4]
  metrics[i, 10:11,"closeness"] <- c(peripheral_change_c, target_change_c)

  ## Betweenness ##
  betweenness_subset <- betweenness(g_subset,
                                    weights = 1/abs(E(g_subset)$weight))
  metrics[i, 1:4, "betweenness"] <- betweenness_subset
  betweenness_full <- betweenness(g_full, weights = 1/abs(E(g_full)$weight))
  metrics[i, 5:9, "betweenness"] <- betweenness_full
  peripheral_change_b <- mean(betweenness_full[1:3] - betweenness_subset[1:3])
  target_change_b <- betweenness_full[4] - betweenness_subset[4]
  metrics[i,10:11,"betweenness"] <- c(peripheral_change_b, target_change_b)
}

head(metrics)
```

```
## , , strength
##
##        p1_true   p2_true   p3_true    target  p1_redun  p2_redun  p3_redun
## [1,] 0.4914635 0.7508084 0.8726357 0.4762247 0.5814956 0.6658460 0.7702376
## [2,] 0.4102845 0.8099328 0.4224413 0.8475045 0.3089079 0.7607501 0.3297134
## [3,] 0.9371334 0.6658636 0.4550099 0.4506893 0.8296702 0.5896502 0.5413340
## [4,] 0.7310798 0.6858690 0.6993944 0.6208082 0.6730372 0.6140682 0.6355719
## [5,] 0.6049693 0.6324735 0.7644737 0.6574459 0.5409826 0.5581302 0.6976465
## [6,] 0.9329169 0.4227576 0.7880320 0.3942471 0.8377538 0.5082476 0.7060112
##      target_redun     clone peripheral_change target_change
## [1,]    0.9067333 0.9067333       -0.03244281     0.4305085
## [2,]    1.0583183 1.0583183       -0.08109569     0.2108139
## [3,]    0.9012887 0.9012887       -0.03245083     0.4505994
## [4,]    0.9813376 0.9813376       -0.06455527     0.3605293
## [5,]    0.9942149 0.9942149       -0.06838571     0.3367690
## [6,]    0.8872372 0.8872372       -0.03056465     0.4929901
##
## , , closeness
##
##         p1_true    p2_true    p3_true     target   p1_redun   p2_redun
## [1,] 0.12736819 0.09850815 0.10626603 0.12515673 0.05105136 0.04781339
```

```
## [2,] 0.10569096 0.08566200 0.10728554 0.08814013 0.04981545 0.03824886
## [3,] 0.10207669 0.08541072 0.11689603 0.11595946 0.05381209 0.04089823
## [4,] 0.07802403 0.07546280 0.07522037 0.06827123 0.02758250 0.03149573
## [5,] 0.06385508 0.06651230 0.08353377 0.07265990 0.02731976 0.03043945
## [6,] 0.09251147 0.11073552 0.08346597 0.10859053 0.04590055 0.04368383
##        p3_redun target_redun      clone peripheral_change target_change
## [1,] 0.05516477   0.04698016 0.04698016       -0.05937095   -0.07817657
## [2,] 0.04761445   0.03935365 0.03935365       -0.05431991   -0.04878648
## [3,] 0.04817399   0.04372572 0.04372572       -0.05383304   -0.07223374
## [4,] 0.02910482   0.02516429 0.02516429       -0.04684138   -0.04310694
## [5,] 0.03103625   0.02684197 0.02684197       -0.04170190   -0.04581793
## [6,] 0.04030564   0.03983201 0.03983201       -0.05227431   -0.06875852
##
## , , betweenness
##
##      p1_true p2_true p3_true target p1_redun p2_redun p3_redun target_redun
## [1,]       1       0       0      0        1        0        0            0
## [2,]       0       0       1      0        2        0        0            0
## [3,]       0       0       1      0        2        0        1            0
## [4,]       0       0       0      0        0        0        0            0
## [5,]       0       0       1      0        0        0        1            0
## [6,]       0       1       0      0        0        1        0            0
##      clone peripheral_change target_change
## [1,]     0         0.0000000             0
## [2,]     0         0.3333333             0
## [3,]     0         0.6666667             0
## [4,]     0         0.0000000             0
## [5,]     0         0.0000000             0
## [6,]     0         0.0000000             0
```

Form 2: HCDF

```
# Check weights argument input for betweenness and closeness
```