

Examples of using GraphMM package

Tien Vo, Michael Newton

23 April, 2019

Loading library

```
library(GraphMM)
```

Example for data associated with line graph

Simulate data from multivariate normal distribution

```
set.seed(12345)
n.ver = 20
m1 = 100
m2 = 150
dat1.vec = MASS::mvrnorm(n=m1, mu=c(rep(0, n.ver/2), rep(0.4, n.ver/2)), Sigma = diag(0.5,
n.ver, n.ver))
dat2.vec = MASS::mvrnorm(n=m2, mu=c(rep(0, n.ver/2), rep(0.6, n.ver/2)), Sigma = diag(0.7,
n.ver, n.ver))

dataG1 = dat1.vec
dataG2 = dat2.vec
```

Use function GraphMMcompute() to perform multiple hypothesis testing for equal means. Results give posterior probability of null hypothesis for each graph's node. Since the method is computationally heavy, it is suggested to use 8 cores to run this example.

```
mccores = 8
res = GraphMMcompute(dataG1, dataG2, type = "line_graph",
                      mccores = mccores)
print(res)
```

```
## [1] 0.88245213 0.87963951 0.84862260 0.89712751 0.91200438 0.91406223
## [7] 0.87037182 0.87773520 0.80922378 0.83010432 0.01102405 0.12707691
## [13] 0.67841648 0.33912975 0.21610754 0.02434085 0.03420157 0.07570003
## [19] 0.58221181 0.82128455
```

Example for data associated with lattice graph

Simulate data from multivariate normal distribution

```
set.seed(4779541)
datadim = c(3,3)
n.ver = prod(datadim)
m1 = 200
m2 = 250
dat1.vec = MASS::mvrnorm(n=m1, mu=c(1, 1, 1, 1, 1, 0, 0, 2, 2),
                          Sigma = diag(0.7, n.ver, n.ver))
dat2.vec = MASS::mvrnorm(n=m2, mu=c(1.2, 1.2, 1.2, 1.2, 1.2, 0, 0, 2, 2),
                          Sigma = diag(1, n.ver, n.ver))

dataG1 = array(c(dat1.vec), c(m1, datadim))
dataG2 = array(c(dat2.vec), c(m2, datadim))
```

Use function GraphMMcompute() to perform multiple hypothesis testing for equal means. Results give posterior probability of null hypothesis for each graph's node. Since the method is computationally heavy, it is suggested to use 8 cores to run this example.

```
folder = "./Temp"
est_null = "ashr"
est_hyper = "mixed"
mccores = 8

res = GraphMMcompute(dataG1 = dataG1, dataG2 = dataG2,
```

```

        est_null = est_null, est_hyper = est_hyper,
        folder = folder, mcores = mcores)

print(res)

```

```

##           [,1]      [,2]      [,3]
## [1,] 0.016939305 0.02975482 0.9023456
## [2,] 0.002719217 0.01980029 0.9395013
## [3,] 0.037678091 0.93334640 0.9369102

```

Example for data associated with a general graph

Create graph structure

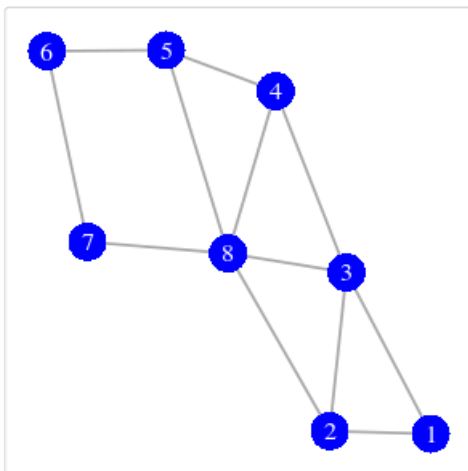
```

Nunits = 8
FullG = matrix(c(1, 2,
                 1, 3,
                 2, 3,
                 2, 8,
                 3, 4,
                 3, 8,
                 4, 5,
                 4, 8,
                 5, 6,
                 5, 8,
                 6, 7,
                 7, 8), 12, 2, byrow = T)

graph = list()
graph[[1]] = Nunits
graph[[2]] = FullG

```

Illustration of graph structure



Simulate data from multivariate normal distribution

```

set.seed(4779541)
n.ver = Nunits
m1 = 200
m2 = 250
dataG1 = MASS::mvrnorm(n=m1, mu=c(1, 1, 1, 0, 0, 0, 1, 1),
                        Sigma = diag(0.7, n.ver, n.ver))
dataG2 = MASS::mvrnorm(n=m2, mu=c(1.2, 1.2, 1.2, 0, 0.2,
                                   0, 1.2, 1.2),
                        Sigma = diag(1, n.ver, n.ver))

```

Use function `GraphMMcompute()` to perform multiple hypothesis testing for equal means. Results give posterior probability of null hypothesis for each graph's node.

```

folder = "./Temp"
est_null = "ashr"
est_hyper = "mixed"
mcores = 2

res = GraphMMcompute(dataG1 = dataG1, dataG2 = dataG2,
                      type = "general_graph", graph = graph,

```

```
        est_null = est_null,  
        est_hyper = est_hyper,  
        folder = folder, mcores = mcores)  
  
print(res)
```

```
## [1] 0.002101755 0.095441771 0.043090713 0.914424738 0.822639997 0.678375063  
## [7] 0.074257074 0.006994855
```