

MGC on PFactor

Vidur Kailash

2/28/2018

Created 2 random 5x5 matrices to test MGC to make sure it was working properly.

```
require(mgc)

## Loading required package: mgc
## Loading required package: SDMTools

X <- matrix(abs(runif(25)), ncol=5)
X <- upper.tri(X)*X; X <- X + t(X)
X

##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.00000000 0.8820536 0.8715908 0.05857695 0.6161585
## [2,] 0.88205358 0.0000000 0.6449677 0.34278825 0.4320513
## [3,] 0.87159082 0.6449677 0.0000000 0.63017633 0.0838948
## [4,] 0.05857695 0.3427883 0.6301763 0.00000000 0.9205646
## [5,] 0.61615846 0.4320513 0.0838948 0.92056459 0.0000000

Y <- matrix(abs(runif(25)), ncol=5)
Y <- upper.tri(Y)*Y; Y <- Y + t(Y)
Y

##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.00000000 0.09362027 0.9292537 0.92396069 0.07139621
## [2,] 0.09362027 0.00000000 0.4948872 0.51985011 0.14678048
## [3,] 0.92925368 0.49488721 0.0000000 0.95751962 0.78992003
## [4,] 0.92396069 0.51985011 0.9575196 0.00000000 0.01586974
## [5,] 0.07139621 0.14678048 0.7899200 0.01586974 0.00000000

res <- mgc.test(X,Y)
print(res$statMGC)

## [1] -0.7263488
```

Since MGC requires 2 nxn distance matrices to inputted, I did the following caluculations on the pfactor data and the connectome data to get the input data in usaable form.

I found 5 patients with realtively high p factor vlaues. I found the pairwise distacnes of each and created a 5x5 matrix with the results and called that matrix 'x'.

I then got the corresponding connectomes from ndmg output (using the connectomes from the desikan poarcellation). I converted the connectomes into numpy matrices (shown in

jupyter notebook) and found the frobenius norm of the distance matrices. This became the second 5x5 distance matrix.

I then ran MGC on the 2 nxn matrices and got a statMGC value of -0.2202395.

```
require(mgc)
require(ggplot2)

## Loading required package: ggplot2

require(latex2exp)

## Loading required package: latex2exp

x = matrix( c(0, 0.0134725, 0.33970251, 0.647503, 0.0656137, 0.0134725, 0, 0.
353175, 0.6609755, 0.0790862, 0.33970251, 0.353175, 0, 0.3078005, 0.27408879,
0.647503, 0.6609755, 0.3078005, 0, 0.58188927, 0.0656137, 0.0790862, 0.274088
79, 0.58188927, 0), nrow=5, ncol=5)
x

##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.0000000 0.0134725 0.3397025 0.6475030 0.0656137
## [2,] 0.0134725 0.0000000 0.3531750 0.6609755 0.0790862
## [3,] 0.3397025 0.3531750 0.0000000 0.3078005 0.2740888
## [4,] 0.6475030 0.6609755 0.3078005 0.0000000 0.5818893
## [5,] 0.0656137 0.0790862 0.2740888 0.5818893 0.0000000

y = matrix( c(0, 21.0134223, 16.49268088, 23.9195127, 19.01896152, 21.0134223
, 0, 11.0393952, 12.62081885, 10.5266986, 16.49268088, 11.0393952, 0, 14.5288
314, 10.7402878, 23.9195127, 12.62081885, 14.5288314, 0, 12.42300484, 19.0189
6152, 10.5266986, 10.7402878, 12.42300484, 0), nrow=5, ncol=5)
y

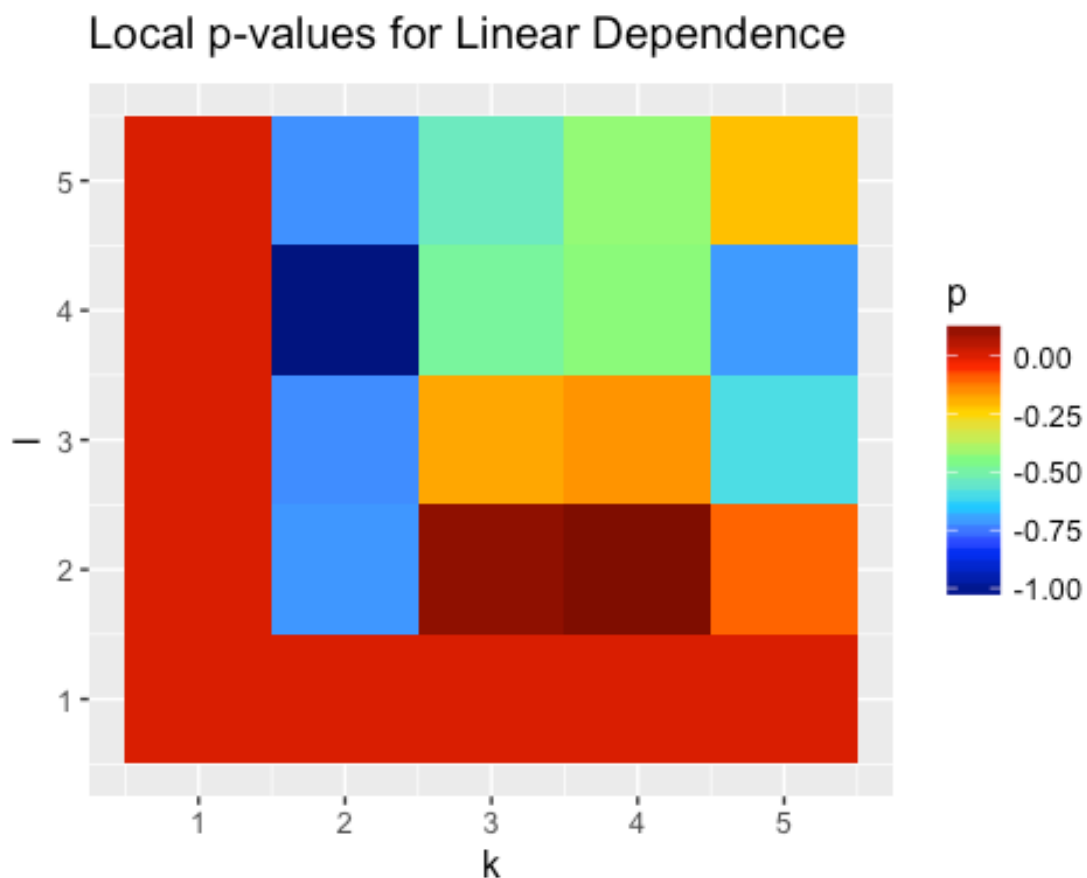
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.00000 21.01342 16.49268 23.91951 19.01896
## [2,] 21.01342 0.00000 11.03940 12.62082 10.52670
## [3,] 16.49268 11.03940 0.00000 14.52883 10.74029
## [4,] 23.91951 12.62082 14.52883 0.00000 12.42300
## [5,] 19.01896 10.52670 10.74029 12.42300 0.00000

mgc.test(x, y)

## $pMGC
## [1] 0.751
##
## $statMGC
## [1] -0.2202395
##
## $pLocalCorr
##           [,1] [,2] [,3] [,4] [,5]
## [1,]      1 1.000 1.000 1.000 1.000
## [2,]      1 1.000 0.985 1.000 0.961
```

```
## [3,] 1 0.663 0.899 1.000 0.846
## [4,] 1 0.582 0.779 0.991 0.779
## [5,] 1 0.520 0.980 1.000 0.751
##
## $localCorr
##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0 -0.7115987 -0.7233131 -1.0242553 -0.7189781
## [3,] 0 0.1118481 -0.1849661 -0.4852445 -0.5364851
## [4,] 0 0.1380807 -0.1593607 -0.4320417 -0.4226489
## [5,] 0 -0.1023249 -0.5943204 -0.7069746 -0.2202395
##
## $optimalScale
## [1] 25

res <- mgc.test(x,y)
mgc.plot.plot_matrix(res$localCorr, xlabel=TeX("k"), ylabel=TeX("l"), title =
TeX("Local $p$-values for Linear Dependence"), legend.name=TeX("p"))
```



```
print(res$optimalScale)
## [1] 25
print(res$statMGC)
```

```
## [1] -0.2202395
```