MGC on PFactor

Vidur Kailash

3/7/2018

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,] 0.0000000 0.3620506 0.9463819 0.7819331 0.4765464
## [2,] 0.3620506 0.0000000 0.2034751 0.8580444 0.2870943
## [3,] 0.9463819 0.2034751 0.0000000 0.9852107 0.3863476
## [4,] 0.7819331 0.8580444 0.9852107 0.0000000 0.9350292
## [5,] 0.4765464 0.2870943 0.3863476 0.9350292 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 0.8134588 0.6604503 0.09802727 0.9386084
## [2,] 0.81345882 0.0000000 0.3001436 0.11677002 0.2933737
## [3,] 0.66045028 0.3001436 0.0000000 0.71264395 0.2526982
## [4,] 0.09802727 0.1167700 0.7126440 0.00000000 0.4701080
## [5,] 0.93860840 0.2933737 0.2526982 0.47010803 0.0000000
```

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

```
## [1] 0.2182026
```

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 realtivelty 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 matirces and got a statMGC value of -0.2202395.

The MGC test statistic is the local generalized correlation with the best scale, that is, the scales (k, l) whose local generalized correlation is largest after smoothing (Mgc smooths to address noisy samples).

A local generalized correlation is the generalized correlation that only includes the k smallest distances for each xi , and the l smallest distances for each yi . Mgc computes these local generalized correlations for all possible scales k and l, incrementally increasing the number of neighbors k for each xi , and separately increasing the number of neighbors l for each yi. pMGC value is 0.731. Null: f(x, y) = fxfy (independence) Alt: f(x, y) =/= fxfy

pValue is not significant. Does not show enough evidence to reject the null.

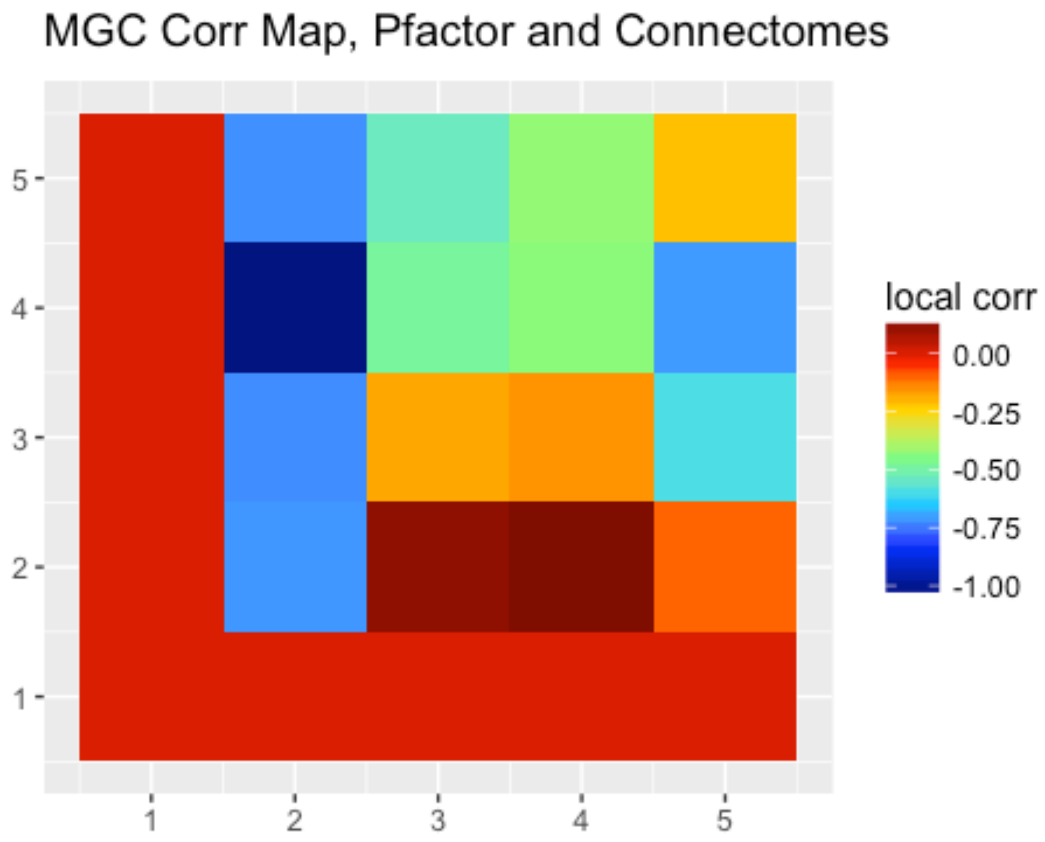
But since there are only 5 samples, the power is very low. Could change with increased number of samples.

```
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] 0.714
## 
## $statMGC
## [1] -0.2202395
## 
## $pLocalCorr
##      [,1]  [,2]  [,3] [,4]  [,5]
## [1,]    1 1.000 1.000 1.00 1.000
## [2,]    1 1.000 0.982 1.00 0.959
## [3,]    1 0.621 0.881 1.00 0.863
## [4,]    1 0.524 0.763 0.99 0.771
## [5,]    1 0.490 0.967 1.00 0.714
## 
## $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, title = TeX("MGC Corr Map, Pfactor and Connectomes"), legend.name=TeX("local corr"))
```



```
print(res$optimalScale)

## [1] 25

print(res$statMGC)

## [1] -0.2202395
```