

使用NBS方法进行多重比较校正

Alex / 2021-12-12 / free_learner@163.com / learning-archive.org

更新于2023-09-18，主要是文字排版上的更新，内容基本保持不变。

一、背景和原理

对于功能连接（或结构连接）矩阵的多重比较校正是一个比较棘手的问题，比如，如果是10x10的功能连接矩阵，就需要进行45次假设检验，如何对这45次假设检验的结果（p值）进行校正呢？一个思路是，对45个p值进行FDR校正，这种方法假设45次假设检验是相互独立的，这种方法往往过于严格，因为实际上这45次假设检验之间并不独立，比如，两条边（edge，即功能连接）可能共享一个节点（node），那么这两条边（及其对应的假设检验）是不独立的。这和voxel-wise（即对每一个体素分别进行假设检验）的多重比较校正的情况是类似的，由于voxel之间具有空间上的连续性，因此voxel之间的假设检验不是相互独立的。为了解决voxel-level的校正过于严格的问题，研究者提出了cluster-level的校正方法。这里要讨论的NBS（network-based statistics）方法是将cluster-level的校正原理拓展到了对于连接矩阵的校正问题上。关于原理更深入的讨论参见：

Zalesky, A., Fornito, A., & Bullmore, E. (2010). Network-based statistic: Identifying differences in brain networks. *NeuroImage*, 53(4), 1197–1207.

NBS方法的具体步骤包括：

1. 对每条边进行统计分析，得到统计量（比如t值）；
2. 设置一个阈值，比如 $t > 3.1$ ，检测超过阈值的边构成的网络（也称为component），计算网络的大小（即网络包含的边的数目或者统计量的总和）；
3. 使用permutation的方法构造零分布（null distribution），在每一次permutation里，重复步骤1和2得到网络的大小。这个零分布的含义是，在随机情况下，可以得到的网络大小的分布；
4. 根据零分布，计算得到p值。注意这个p值是单边的（one-sided）。

二、在MATLAB环境下使用NBS

首先需要区分NBS方法和软件实现。这里称NBS方法的MATLAB实现为NBS包，以便于区分后面的R实现。

由于NBS包的使用手册非常详细，我这里只是总结一些要点。

1. 下载地址：<https://www.nitrc.org/projects/nbs>，目前最新版本仍然是1.2（2012年发布）；
2. 安装方法：将NBS包的文件夹添加到MATLAB搜索路径（Add with Subfolders）；
3. 在MATLAB命令行窗口输入NBS，弹出使用界面；
4. 使用界面包括三部分，在Statistical Model处设置统计模型，统计模型包含设计矩阵（design matrix）和对比向量（contrast vector），在Help处提供了常见的几种统计模型的设置方法。此外，还可以设置形成网络的阈值；在Data处选择输入数据，假设每个被试一个连接矩阵，将所有被试的数据放在一个文件夹下，选择其中一个被试的文件，会自动读取其他被试（注意被试文件的命名规则，被试文件的读取顺序依据字母—数字顺序）。此外，还可以设置每个节点的坐标和标签，方便后续可视化；在Advanced Settings处设置permutation的次数、显著性水平、网络大小的计算方法，也提供了FDR校正的选项。此外，还有一个Exchange Blocks的选项，这个选项在进行paired t-test或者repeated measures ANOVA的时候需要设置。在NBS包的文件夹下有一个名为SchizophreniaExample的文件夹提供了样例数据，在实际分析时可以参考；
5. 运行完成后，在MATLAB命令行窗口输入 `global nbs`，会得到一个包含所有结果的名为nbs的结构变量。在NBS包的使用手册里还提供了很多实现常用功能的代码，比如将显著结果保存到文件中；
6. 如果所有被试的边均为 0，那么这些边将不纳入分析。

三、在R环境下使用NBS

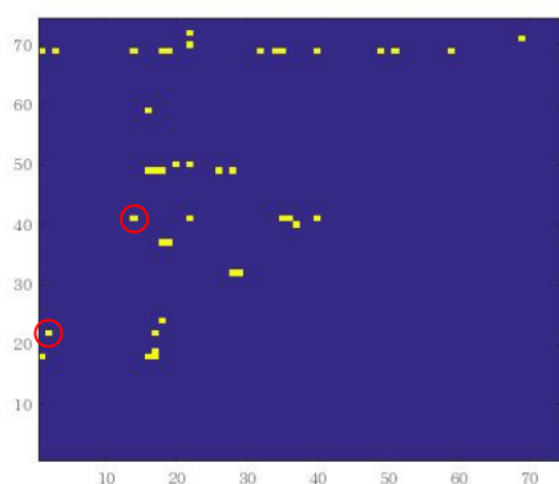
最近我看到一个在R环境下进行NBS分析的工具包（以下称为NBR包，下载地址：<https://cran.r-project.org/web/packages/NBR/index.html>），相比于MATLAB的实现（NBS包），NBR包还提供了线性混合模型（linear mixed model）的功能。这里使用NBS包里的SchizophreniaExample样例数据简单测试一下NBR包的使用方法，并和MATLAB下的NBS包的结果进行对比。

```

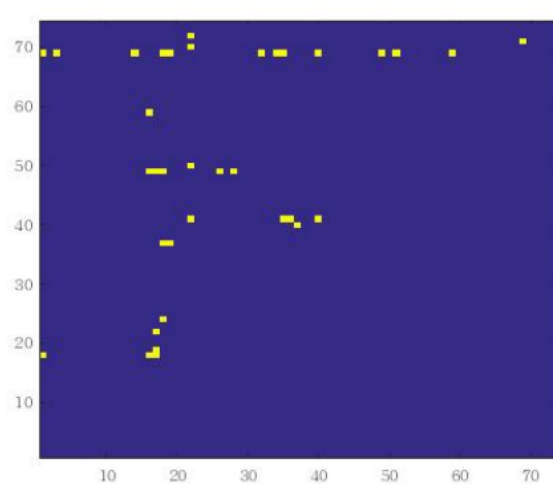
library(NBR)
setwd('/home/Alex/matlab/NBS1.2')
fnames <- list.files('./SchizophreniaExample/matrices', full.names = TRUE)
## read all subjects' FC matrices into a 2D matrix
## columns represent the upper triangular elements and rows represent subjects
FCmat <- NULL
for ( fname in fnames){
  subdat <- read.table(fname)
  subdat <- as.matrix(subdat)
  subdat <- subdat[upper.tri(subdat)]
  FCmat <- rbind(FCmat, subdat)
}
## call NBS using linear model
Grp <- as.factor(c(rep('G1_SZ',12), rep('G2_HC',15)))
## the dataframe must have two columns at least, which is potentially a bug
otherdat <- data.frame(Grp=Grp, CONST=1)
NROI <- 74
results <- nbr_lm(FCmat, NROI, otherdat, mod="~ Grp", diag = FALSE, nperm=1000,
thrP=NULL, thrT = 3.1, nudist = TRUE, cores=4)
## save the network for visualization
net_mat <- matrix(0, nrow=NROI,ncol=NROI)
net_idx <- result$components$GrpG2_HC[,4] == 1
net_mat[result$components$Grp[net_idx,2:3]] <- 1

```

使用NBS包的结果是只有一个网络，该网络包含40条边，p值为0.035；而使NBR包的结果是包含5个网络，第一个网络包含35条边，p值为0.046，具体边的差异如下图所示。因此，NBR包在网络检测上（即判断超过阈值的边是否构成一个网络）和NBS包的结果不同，具体原因不清楚。此外，NBR包的运行速度要慢很多。



NBS



NBR