Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

The brainKCCA Package: Building Region-level Functional Connectivity Network

Xubo Yue
University of Michigan
Ann Arbor

Chia-Wei Hsu
University of Michigan
Ann Arbor

Jian Kang
University of Michigan
Ann Arbor

Abstract

In this article, we introduce an R package, **brainKCCA**, for a region-level functional connectivity network analysis using the voxel-level resting state functional magnetic resonance imaging (fMRI) data. This package adopts a multi-attribute canonical graph approach (Kang, Bowman, Mayberg, and Liu 2016) to measuring the strength of functional connectivity for the whole brain network and assessing the statistical significance via permutation tests. The core algorithm is implemented with the parallel computing option, which is scalable to the analysis of massive neuroimaging data. This package also provides a useful brain network visualization tool which can flexibly show all the network connections among predefined regions in a three dimensional brain from any viewpoint. We provide hands-on tutorials on how to analyze the resting state fMRI data using this package and provide an example to the analysis of functional connectivity networks in the Autism Brain Imaging Data Exchange (ABIDE) study.

Keywords: **brainKCCA**, autism spectrum disorder, kernel canonical correlation analysis, permutation test, connectivity network, R.

1. Introduction

Brain functional connectivity is defined as the connectivity among regions that share functional properties. It has been shown that disruptions in brain connectivity can cause cognitive deficits or neurological disorders (Mayberg 2003). To capture the functional connectivity between regions, functional magnetic resonance imaging (fMRI) has been widely used in the field of neuroscience. fMRI records signals of brain activities by detecting changes associated with blood flow (Rinck 2014). There are approximately 300,000 brain voxels and more than 40 billion voxel pairs in a standard 2mm brain template, imposing a computational challenge to perform functional connectivity analysis at voxel level using fMRI data.

Numerous statistical approaches for functional connectivity analysis have been proposed. Independent component analysis (ICA) (Calhoun, Adali, Pekar, and Pearlson 2003) has been widely used to perform brain functional network analysis at voxel level. This method can identify a group of brain regions having strong functional connectivity, but does not focus on the functional connectivity between regions pairs. Multivariate pattern analysis-based classifier (Zeng, Shen, Liu, Wang, Li, Fang, Zhou, Li, and Hu 2012) has been utilized to discriminate the functional connectivity between different groups with applications to analysis of major depressive patients. Bayesian hierarchical predictive models (Dai, Guo, Initiative *et al.* 2017) have also been proposed for study functional connectivity at region level. This method allows the possibility of predicting disease progression or treatment-related changes in functional connectivity. However, the method is not scalable to voxel level analysis due to heavy computational cost of Bayesian posterior computation. Correlation analyses have also been widely adopted in functional connectivity studies (Power, Barnes, Snyder, Schlaggar, and Petersen 2012; Takagi, Sakai, Lisi, Yahata, Abe, Nishida, Nakamae, Morimoto, Kawato, Narumoto *et al.* 2017) at region level. The existing brain atlas, such as the 116 regions generated by the automated anatomical labeling (AAL) system, can provide a computationally feasible approach to the analysis of region-level functional connectivity. The voxel-level correlation analysis has not been frequently used as it has two major limitations: 1) the low signal to noise ratio in fMRI data, and 2) the challenges in performing multiple corrections to any statistical testing procedures with a huge number of voxel-pairs.

Canonical correlation analysis (CCA) can measure the linear association between two variables. It was first introduced by Hotelling (Hotelling 1936) and widely applied in the field of imaging analysis. However, CCA fails to extract some useful features due to its linearity. To increase the flexibility of the feature selection, Lai (Lai and Fyfe 2000) proposed the kernel canonical correlation analysis (KCCA) which can explore the nonlinear relationship between two variables. It transforms sample vectors into high dimensional feature space and maximize correlation coefficient by solving generalized eigenvalue problem. KCCA has been widely employed in machine learning area such as facial recognition (Zheng, Zhou, Zou, and Zhao 2006), and in brain imaging area such as fMRI data analysis (Hardoon, Mourao-Miranda, Brammer, and Shawe-Taylor 2007).

To the best of our knowledge, there has been relatively less user-friendly and flexible computer software providing brain network construction at region level using voxel level information. By implementing the multi-attribute graph model developed by Kang and others (Kang, Bowman, Mayberg, and Liu 2016), we designed an `R` package **brainKCCA** for assessing the functional brain connectivity for each region pairs and then constructing the whole-brain network. The package employs kernel canonical correlation analysis to pairwise assess the strength of connectivity between brain regions. The statistical significance of connectivity is established by permutation test with a control of the false discover rate. We resort to an `R` package **brainR** (Muschelli, Sweeney, and Crainiceanu 2014) for the functional connectivity and the whole-brain network visualization. This package also provides an option for parallel computing.

Our package mainly has the following advantages:

1. The core algorithm is implemented with the parallel computing option, which enables us to perform the analysis of massive neuroimaging data.

2. Our package provides a useful brain network visualization tool which can flexibly show

all the network connections among predefined regions in a three dimensional brain from any viewpoint.

3. Our package is flexible to accept various types of data. For example, the raw fMRI data, the processed neuroimaging data, and the adjacent matrix data.

4. Our package is user-friendly. We allow user to specify their own preferences in the function argument and save outputs in multiple ways.

The paper is organized as follows. In Section 2 we present the method of brain connectivity network analysis. In Section 3 we provide a hands-on tutorials on how to analyze the resting state fMRI data using this package. We illustrate the package performance on an analysis of resting state fMRI data in the ABIDE study in Section 4, and we conclude our paper in Section 5.

2. Method

Suppose we collect fMRI data at T time points over p regions. Let n_i be the number of voxels in region i. For $i = 1, \ldots, p$, $k = 1, \ldots, n_i$ and $t = 1, \ldots, T$, denote by x_{ikt} the observed fMRI signal at voxel k in region i at time t. Denote $\mathbf{x}_{it} = (x_{i1t}, x_{i2t}, \ldots, x_{in_it})'$ as a column vector of fMRI signals. We assume the temporal correlation of fMRI signals has been removed by a pre-whitening procedure using an autoregressive model. Consequently, $x_{i1}, \ldots, x_{iT}$ are independent and identically distributed ($i.i.d$). Denote $\mathbf{X}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iT})'$ as an centered and scaled fMRI signal matrix of $T \times n_i$ dimension. Let $\phi_n : \mathbb{R}^n \to \mathcal{F}_{\mathbb{R}^n}$ denote feature space mappings corresponding to the kernel function $\boldsymbol{\kappa}_n(\boldsymbol{a}, \boldsymbol{b}) =< \boldsymbol{\phi}_n(\boldsymbol{a}), \boldsymbol{\phi}_n(\boldsymbol{b}) >$. We represent the fMRI signal $\mathbf{x}_i$ in the $T \times n_i$ dimension feature space $\boldsymbol{\Phi}_{n_i} = [\boldsymbol{\phi}_{n_i}(\mathbf{x}_{i1}), \ldots, \boldsymbol{\phi}_{n_i}(\mathbf{x}_{iT})]'$. Denote by $\mathbf{K}_i = \boldsymbol{\Phi}_{n_i}\boldsymbol{\Phi}'_{n_i}$ a $T \times T$ inner product kernel matrix. The estimates of kernel canonical correlation coefficient (KCCC) between region $(i, j)(1 \leq i < j \leq p)$ can be expressed by

$$\begin{aligned} \hat{\rho}_c(i,j) = & \max_{\boldsymbol{\omega}_i \in \mathbb{R}^T, \boldsymbol{\omega}_j \in \mathbb{R}^T} \boldsymbol{\omega'}_i \mathbf{K'}_i \mathbf{K}_j \boldsymbol{\omega}_j \\ & \text{subject to } \boldsymbol{\omega}'_i \mathbf{K}'_i \mathbf{K}_i \boldsymbol{\omega}_i \leq 1 \text{ and } \boldsymbol{\omega}'_j \mathbf{K}'_j \mathbf{K}_j \boldsymbol{\omega}_j \leq 1, \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}_i \in \mathbb{R}^T$ represents a column vector of weights. We assume that $\hat{\rho}_c(i, j)$ does not change over time. (1) is equivalent to eigenvalue equations:

$$\begin{cases} (\mathbf{K}'_j \mathbf{K}_j)^{-1} \mathbf{K}'_j \mathbf{K}_i (\mathbf{K}'_i \mathbf{K}_i)^{-1} \mathbf{K}'_i \mathbf{K}_j \boldsymbol{w}_j = \lambda^2 \boldsymbol{\omega}_j \\ (\mathbf{K}'_i \mathbf{K}_i)^{-1} \mathbf{K}'_i \mathbf{K}_j (\mathbf{K}'_j \mathbf{K}_j)^{-1} \mathbf{K}'_j \mathbf{K}_i \boldsymbol{w}_i = \lambda^2 \boldsymbol{\omega}_i \end{cases} \tag{2}$$

$\mathbf{K}_i$ and $\mathbf{K}_j$ can be singular. To enhance numerical stability, a regularization term $\boldsymbol{\gamma}$ can be added to equation (2):

$$\begin{cases} (\mathbf{K}'_j \mathbf{K}_j + \boldsymbol{\gamma}\mathrm{I})^{-1} \mathbf{K}'_j \mathbf{K}_i (\mathbf{K}'_i \mathbf{K}_i + \boldsymbol{\gamma}\mathrm{I})^{-1} \mathbf{K}'_i \mathbf{K}_j \boldsymbol{w}_j = \lambda^2 \boldsymbol{\omega}_j \\ (\mathbf{K}'_i \mathbf{K}_i + \boldsymbol{\gamma}\mathrm{I})^{-1} \mathbf{K}'_i \mathbf{K}_j (\mathbf{K}'_j \mathbf{K}_j + \boldsymbol{\gamma}\mathrm{I})^{-1} \mathbf{K}'_j \mathbf{K}_i \boldsymbol{w}_i = \lambda^2 \boldsymbol{\omega}_i \end{cases} \tag{3}$$

(3) is equivalent to maximizing

$$\frac{\boldsymbol{\omega'}_i \mathbf{K'}_i \mathbf{K}_j \boldsymbol{\omega}_j}{\sqrt{\boldsymbol{\omega}'_i \mathbf{K}'_i \mathbf{K}_i \boldsymbol{\omega}_i + \boldsymbol{\gamma}\boldsymbol{\omega'}_i \boldsymbol{\omega}_i} \sqrt{\boldsymbol{\omega}'_j \mathbf{K}'_j \mathbf{K}_j \boldsymbol{\omega}_j + \boldsymbol{\gamma}\boldsymbol{\omega'}_j \boldsymbol{\omega}_j}}. \tag{4}$$

Please refer to the multi-attribute canonical graph model (Kang, Bowman, Mayberg, and Liu 2016) for more details.

The hypothesis $H_0 : \rho_c(i,j) = 0$, versus $H_1 : \rho_c(i,j) \neq 0$, for each region pair (i,j) is assessed by permutation test. We permute the data over $\{1, 2, \ldots, T\}$ scans, yielding a total of L permutations indexed by $l = 1, \ldots, L$. Let $\hat{\rho}_c(i,j)$ denote the permuted kernel canonical correlation coefficient (KCCC), the p-value is obtained by

$$\hat{p}(i,j) = \frac{1}{L} \sum_{l=1}^{L} \mathrm{I}[|\hat{\rho}_c^{(l)}(i,j)| > |\hat{\rho}_c(i,j)|], \tag{5}$$

where $\mathrm{I}(\cdot)$ is an indicator function. All $\hat{p}(i,j)$'s are corrected by controlling the false discovery rate in multiple testing. The significance level is set to 0.05.

3. Software Demonstration

Two core functions are provided for estimating connectivity and brain network visualization, where `permkCCA_multipleRegion` is designed for the purpose of calculating strength of connectivity among multiple brain regions; and `multipleRegion_plot` is developed to visualize the brain network. Other ancillary functions are also introduced in this section. The example fMRI data can be downloaded from https://github.com/xuboyue/brainKCCA_data.

3.1. Package Installation

Package is available in the The Comprehensive R Archive Network (CRAN). The dependencies are mainly **utils**, **elasticnet**, **rgl**, **CCA**, **kernlab**, **brainR**, **oro.nifti**, **misc3d**, **knitr** and **parallel**.

3.2. Data Format

The resting-state functional magnetic resonance imaging (Rs-fMRI) data of patients/controls were preprocessed based on the Configurable Pipeline for the analysis of Connectomes (C-PAC). The default images were registered in the standard 2mm MNI space with resolution $91 \times 109 \times 91$. The flexibility of core functions in **brainKCCA** allows other possible setting of processed Rs-fMRI data such as the standard 3mm MNI space. Figure 1 shows the scan of the NIfTI region file registered in 2mm MNI space.

Figure 2 demonstrates the layout of NIfTI region file registered in 2mm MNI space. As aforementioned, the dimensionality of this file is $91 \times 109 \times 91$.

The format of the region list (Figure 3) is defined as follows: the first three columns are the (x, y, z) coordinates of brain region. The fourth column is the brain region code. The fifth column is the region index. The sixth column is the region name. Users are allowed to create their own region list file: they only need to specify the last three columns of region list. The default region list and NIfTI region file (AAL 116 regions in the 2mm MNI space) have been provided in the R package.

3.3. Data Processing

Function `nii2RData` can convert an fMRI NIfTI file (with extension nii or nii.gz) to an R format, where the input argument `imgPath` specifies the path to NIfTI data, `niiFile1`

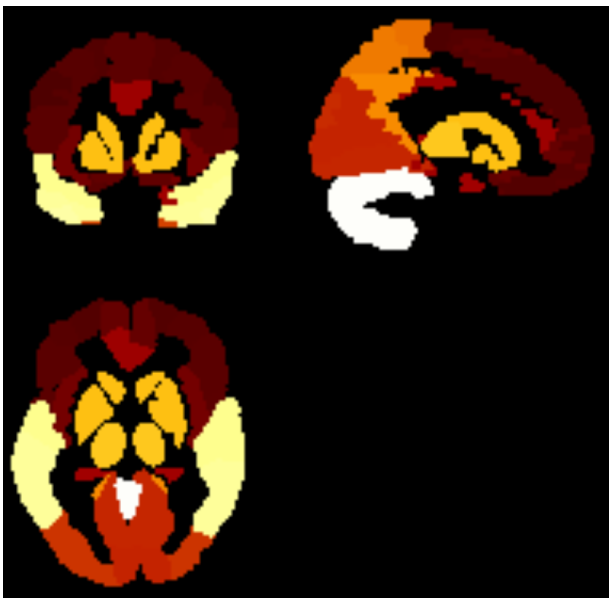

Figure 1: Image of 2mm MNI region file.

```
nif                          Large nifti (902629 elements, 6.9 Mb)
  ..@ .Data : num [1:91, 1:109, 1:91] 0 0 0 0 0 0 0 0 0 0 ...
  ..@ sizeof_hdr : int 348
  ..@ data_type : chr ""
  ..@ db_name : chr ""
  ..@ extents : int 0
  ..@ session_error : int 0
  ..@ regular : chr "r"
  ..@ dim_info : chr ""
  ..@ dim_ : int [1:8] 3 91 109 91 1 1 0 0
```

Figure 2: Format of the NIfTI region data.

	Var1	Var2	Var3	V1	V2	V3
2001	-38.930233	-6.9614294	49.6403857	1	2001	Precentral_L
2002	41.100858	-9.5498373	50.8098196	2	2002	Precentral_R
2101	-18.772992	33.4898583	40.9541539	3	2101	Frontal_Sup_L
2102	21.603057	29.9102564	42.5044379	4	2102	Frontal_Sup_R
2111	-16.824507	45.9190031	-14.7185877	5	2111	Frontal_Sup_Orb_L
2112	18.178536	46.6178536	-15.4944835	6	2112	Frontal_Sup_Orb_R
2201	-33.758997	31.4892042	34.1307835	7	2201	Frontal_Mid_L
2202	37.391850	31.7507837	32.8142633	8	2202	Frontal_Mid_R
2211	-30.885135	49.1418919	-10.9864865	9	2211	Frontal_Mid_Orb_L
2212	32.894581	51.2748768	-12.1418719	10	2212	Frontal_Mid_Orb_R
2301	-48.795761	11.4913295	17.8015414	11	2301	Frontal_Inf_Oper_L
2302	49.927091	13.6754825	20.1958542	12	2302	Frontal_Inf_Oper_R
2311	-45.890075	28.6666667	12.5828391	13	2311	Frontal_Inf_Tri_L
2312	50.061367	28.9019061	12.8172943	14	2312	Frontal_Inf_Tri_R
2321	-36.240237	29.4923077	-13.4591716	15	2321	Frontal_Inf_Orb_L
2322	40.910369	30.9947276	-13.2583480	16	2322	Frontal_Inf_Orb_R
2331	-47.406061	-9.7757576	12.6383838	17	2331	Rolandic_Oper_L
2332	52.381668	-7.5371901	13.3298272	18	2332	Rolandic_Oper_R
2401	-5.690731	3.5500699	60.0642757	19	2401	Supp_Motor_Area_L

Figure 3: Format of the region list.

specifies fMRI NIfTI data file. The output of `nii2RData` is an R list object consisting of two elements: one is an fMRI signal matrix where rows are fMRI scans and columns index voxels; the other is an R list object for the region list (the format is similar to Figure 3). We refer to the output of `nii2RData` as the fMRI data in R format. Function `nii2RData` is able to convert multiple fMRI NIfTI files, in which case the input argument `niiFile1` can take multiple NIfTI files and the output is still an R list object but consisting of multiple fMRI signal matrices along with the region list.

Users can call this function and save the fMRI data in R format for further analysis. Users can name their saved files by using argument `saveName`. The saved R data files are located in `datPath`.

```
R> testcase1 <- nii2RData(niiFile1 = "preproc_con21_rest_MNI_2mm",
+  resolution = "2mm", saveName = "patient1.RData", regionCode = "",
+  niiFile2 = "", imgPath = getwd(), datPath = getwd())

Reading nii files...this progress may take a long time...
  |======================================================================| 100%
 reading data  1
  |======================================================================| 100%
 saving data...

 Completed...

R> str(testcase1[[1]])
```

```
 num [1:149, 1:185405] 0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:185405] "V1" "0" "0" "0" ...
```

```
R> str(testcase1[[2]])
```

```
'data.frame':        116 obs. of  6 variables:
 $ Var1: num  -38.9 41.1 -18.8 21.6 -16.8 ...
 $ Var2: num  -6.96 -9.55 33.49 29.91 45.92 ...
 $ Var3: num  49.6 50.8 41 42.5 -14.7 ...
 $ V1  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ V2  : int  2001 2002 2101 2102 2111 2112 2201 2202 2211 2212 ...
 $ V3  : Factor w/ 116 levels "Amygdala_L","Amygdala_R",..: 83 84 47 52 50 51 43 46 44 45
```

3.4. Multiple Regions Connectivity Analysis for Single Subjects

Function `permkCCA_multipleRegion` is designed to calculate the strength of connectivity among multiple brain regions for each subject in the dataset. It can either accept the user-defined region list and the NIfTI region file to partition brain region or employ its default region data to perform partition. To call this function without user-defined region data, data files and regions of interest can be directly provided to this function:

```
R> result0 <- permkCCA_multipleRegion(imageDat = c("preproc_con21_rest_MNI_2mm",
+  "preproc_con23_rest_MNI_2mm"), region = c(1, 60, 70), resolution = "2mm",
+  saveName = "None", kernel = "rbfdot", regionCode = "", niiFile2 = "",
+  imgPath = getwd(), datPath = getwd(),
+  parallel = FALSE, loc = "local", perm = 50, saveData = "None")
```

```
 Checking data format...
Reading nii files...this progress may take a long time...
  |======================================================================| 100%
 reading data  1
  |======================================================================| 100%
 reading data  2
  |======================================================================| 100%
 Completed...

Calculating KCC...
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
  |======================================================================| 100%
```

When the function is running, it will generate progress bars with detailed explanations to notify users the current progress.

The argument `imageDat` can take two types of input: (1) the name of the NIfTI file without extension (.nii or .nii.gz). In this case, function `nii2RData` will be invoked to convert NIfTI file(s) to the fMRI data in R format; (2) the output generated by function `nii2RData`. One advantage of this setting allows users to transform NIfTI data files and store the output R data files so that they do not need to re-read data in the future. The original NIfTI files are supposed to be located in the path specified by `imgPath`.

The argument `region` can take vectors of region indices. The resolution ("2mm" or "3mm") can be specified by the argument `resolution`.

Users can write name (for example, "pro1.RData") in argument `saveName` to save processed data generated by function `nii2RData`. The argument `saveData` specifies the name of the R data to save the output of `permkCCA_multipleRegion` (for example, "result1.RData"). All results are saved in the path specified by `datPath`.

The argument `kernel` can instruct function to calculate correlation by different kernel. `"rbfdot"` represents the the radial basis kernel function "Gaussian". The option `perm` is the number of permutation in the permutation test.

If users do not want to use the default region list or NIfTI region file, they are supposed to specify their own region information in the `regionCode` and `niiFile2`. To call this function with user-defined NIfTI region file and region list:

```
R> result1 <- permkCCA_multipleRegion(imageDat = c("preproc_con21_rest_MNI_2mm",
+  "preproc_con23_rest_MNI_2mm"), region = c(1, 60, 70),
+  regionCode = "RegionList.txt", niiFile2 = "AAL_MNI_2mm.nii")
```

The customized region file can have different partitions and region name. Additionally, users are supposed to provide corresponding region partition data in the argument `niiFile2`. Four key components are in the list objects `result0` and `result1`: region index, p-value, region type ("two" or "multiple"), and region name.

The argument `parallel` determines whether the parallel computing will be executed; and option `loc = "local"` can be used for local computer. Users also can run the parallel computing on cluster, which will be discussed in Section 4.

3.5. Brain Network Analysis for Multiple Subjects

The results generated by `permkCCA_multipleRegion` can be summarized by the function `summary_kcca` for constructing a brain connectivity network for multiple subjects. The first argument is the result generated by `permkCCA_multipleRegion`. The second argument is the significance level. Typically it can be 0.05 or 0.01. The fourth argument indicates which format used to save table. For example, "markdown" indicates outputting summary information as R markdown table.

```
R> summary_kcca(kcca_object = result0, significance = 0.05, patientID = 1,
+  saveFormat = "markdown", threshold = 0.2)

 summary of kcca object  1  generated by 'permkCCA_multipleRegion' function:
```

```
|index1 |index2 |region1 |region2 |pvalue |
|:------|:------|:-------|:-------|:------|
|1      |60     |PreCG.L |SPG.R   |0      |
|1      |70     |PreCG.L |PCL.R   |0      |
|60     |70     |SPG.R   |PCL.R   |0.0367 |
```

Argument `patientID` specifies the indices of the subjects for a multiple-subject connectivity network construction. The option `threshold` specifies the minimum proportion of single-subject connections counted as a multiple-subject connection. When `patientID` has only one subject index, the single-subject network results will be generated. The option `saveFormat = "group"` can generate the multiple subject network, the format of which is the same as the output of `permkCCA_multipleRegion` for the single subject connectivity analysis.

```
R> group_data <- summary_kcca(kcca_object = result0, significance = 0.05,
+  saveFormat = "group", threshold = 0.2)
```

3.6. Brain Network Visualization

This section presents function `multipleRegion_plot` for visualizing the functional connectivity network.

```
R> multipleRegion_plot(input = result0, regionCodeProvided = FALSE,
+   view = "coronal", color = "blue", screenShot = "None")

 You have  2  patients' data
Which data you would like to see? 1
p value of kccc between  PreCG.L  and  SPG.R  is:  0
p value of kccc between  PreCG.L  and  PCL.R  is:  0
p value of kccc between  SPG.R  and  PCL.R  is:  0.0367

R> rgl.snapshot("result1.png")
```

The argument `input` accepts result generated by `permkCCA_multipleRegion`. The option `regionCodeProvided = FALSE` indicates that the user-defined region code has not been provided. The `view` can be specified as "coronal", "axial", "SL" (left sagittal) and "SR" (right sagittal). The argument `color` defines the color of connection lines. The resulting connectivity network will be generated and the function `rgl.snapshot` from package **rgl** can save the screenshot as png file. The resulting connectivity network is reported in Figure 4.

The function also can show multiple views of brain (Figure 5).

```
R> multipleRegion_plot(input = result0,
+  view = c("coronal", "axial", "SL", "SR"))

 You have  2  patients' data
Which data you would like to see? 1
p value of kccc between  PreCG.L  and  SPG.R  is:  0
p value of kccc between  PreCG.L  and  PCL.R  is:  0
p value of kccc between  SPG.R  and  PCL.R  is:  0.0367
```

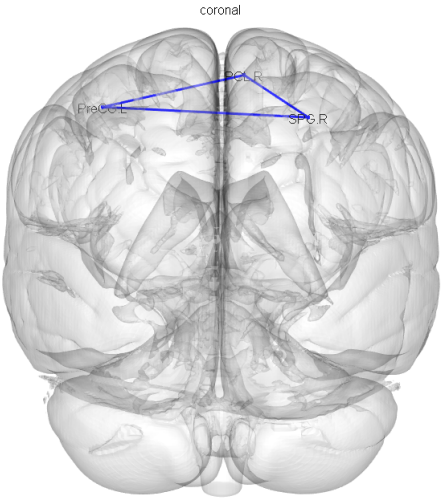

Figure 4: Regions-level connectivity network (coronal view).

```
R> rgl.snapshot("result2.png")
```

We also can show the group-level brain network by using:

```
R> multipleRegion_plot(input = group_data, significance = NA,
+  view = c("coronal", "axial", "SL", "SR"))
R> rgl.snapshot("result3.png")
```

The `significance = NA` means there is no p-value in the `group_data`.

4. Performance and Analysis

We demonstrate the performance of connectivity calculation in this section. We used AAL 90 brain regions with resolution 3mm, and performed multiple region connectivity calculation by 50 times of permutation. The running time was measured under the Portable Batch System (PBS) cluster environment by parallel computing with 16 processors.

```
R> result2 <- permkCCA_multipleRegion(imageDat = "UM_1_0050272_func_preproc",
+   region = c(1:90), resolution = "3mm", regionCode = "RegionList90.txt",
+   niiFile2 = "AAL_90_3mm.nii", parallel = TRUE, loc = "cluster")
```

The argument `parallel = TRUE` will initiate parallel computing and `loc = "cluster"` indicates that the codes will be executed in cluster. If users would like to run parallel computing in local computer, `loc = "local"` can be specified. The average running time is about 5 hours.

We performed a case study of 60 patients (with 34 autism and 26 controls) by using cluster parallel computing in High Performance Computing (HPC) cluster and jobs were submitted via PBS files, where the commands and cluster resources used for jobs were defined.

For each PBS job, the following R codes were recursively used:

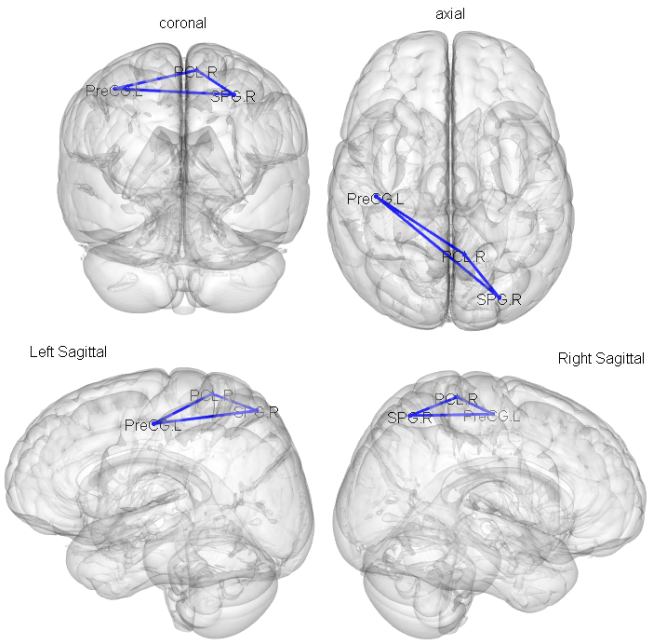

Figure 5: Multiple regions connectivity network (multiple views) .

```
R> array_ID <- as.numeric(Sys.getenv('PBS_ARRAYID'))
R> #set working directory to your imaging files
R> setwd("./UM_1")
R> fileName <- gsub(".nii.gz", "", list.files())[array_ID]
R> setwd("..")
R> testcase <- nii2RData(niiFile1 = fileName, resolution = "3mm",
+  regionCode = "RegionList90.txt", niiFile2 = "AAL_90_3mm.nii",
+  imgPath = "./UM_1", datPath = "./UM_1_result")
R> result <- permkCCA_multipleRegion(imageDat = testcase, region=c(1:90),
+  resolution = "3mm", regionCode = "RegionList90.txt",
+  niiFile2 = "AAL_90_3mm.nii", parallel = TRUE, loc = "cluster")
R> tableTemp <- summary_kcca(result, 0.05, 1, "excel")
R> write.csv(tableTemp, paste(fileName, ".csv", sep = ""), col.names = FALSE)
R> group_data_2 <- meanConnection(path=getwd(), threshold=0.2)
```

60 csv files were generated and the function `meanConnection` was used to process all these files.

Users are supposed to separate cases and controls csv files. Users can specify locations of their csv files in argument `path`. The percentage of connection is calculated based on significance in each region pair. The function `meanConnection` will return a data frame with significant region index and region name (without p-value) and the function `multipleRegion_plot` can accept this dataframe as an argument and generate region-level connectivity networks (Figure 6, Figure 7). Users need to set `significance = NA` in this case.

```
R> multipleRegion_plot(input = group_data_2, significance = NA,
```

```
+  view = c("coronal", "axial", "SL", "SR"))
R> rgl.snapshot("result3.png")
```

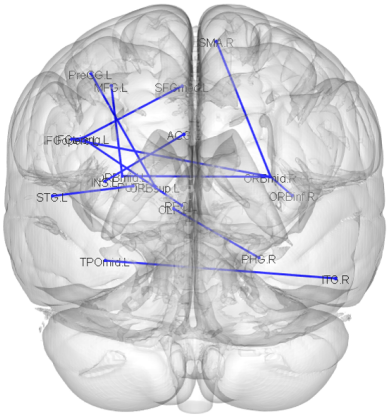

Figure 6: Multiple regions connectivity network for case.

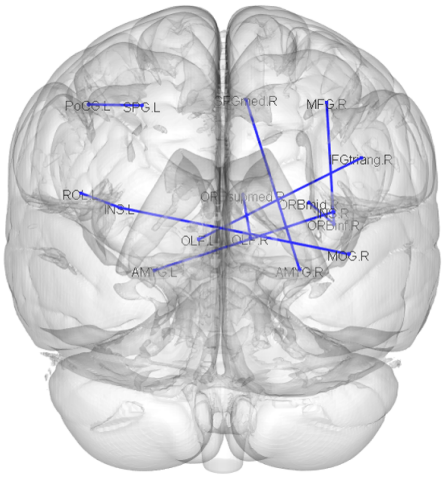

Figure 7: Multiple regions connectivity network for control.

5. Conclusion and Future Work

In this article, we introduce a new R package, **brainKCCA**, for building up the region-level brain networks using voxel-level information with flexible brain network visualization tools. The core algorithm is implemented with the parallel computing option, which is scalable to the analysis of massive neuroimaging data. The tutorials on how to analyze the resting state fMRI data have been provided and an example to the analysis of functional connectivity networks in the Autism Brain Imaging Data Exchange (ABIDE) study has been presented. The permutation test requires intensive computations, a scalable Bayesian kernel canonical

correlation analysis can be developed for large-scale network construction. We will pursue this elsewhere.

References

Akaho S (2006). "A kernel method for canonical correlation analysis." *arXiv preprint cs/0609071*. URL https://arxiv.org/abs/cs/0609071.

Calhoun V, Adali T, Pekar J, Pearlson G (2003). "Latency (in) sensitive ICA: group independent component analysis of fMRI data in the temporal frequency domain." *NeuroImage*, **20**(3), 1661–1669. doi:10.1016/S1053-8119(03)00411-7.

Dai T, Guo Y, Initiative ADN, *et al.* (2017). "Predicting individual brain functional connectivity using a Bayesian hierarchical model." *NeuroImage*, **147**, 772–787. doi:10.1016/j.neuroimage.2016.11.048.

Hardoon DR, Mourao-Miranda J, Brammer M, Shawe-Taylor J (2007). "Unsupervised analysis of fMRI data using kernel canonical correlation." *NeuroImage*, **37**(4), 1250–1259. doi:10.1016/j.neuroimage.2007.06.017.

Hotelling H (1936). "Relations between two sets of variates." *Biometrika*, **28**(3/4), 321–377. doi:10.1093/biomet/28.3-4.321.

Kang J, Bowman FD, Mayberg H, Liu H (2016). "A depression network of functionally connected regions discovered via multi-attribute canonical correlation graphs." *NeuroImage*, **141**, 431–441. doi:10.1016/j.neuroimage.2016.06.042.

Lai PL, Fyfe C (2000). "Kernel and nonlinear canonical correlation analysis." *International Journal of Neural Systems*, **10**(05), 365–377. doi:10.1142/S012906570000034X.

Mayberg HS (2003). "Modulating dysfunctional limbic-cortical circuits in depression: towards development of brain-based algorithms for diagnosis and optimised treatment." *British medical bulletin*, **65**(1), 193–207. doi:10.1093/bmb/65.1.193.

Muschelli J, Sweeney E, Crainiceanu C (2014). "brainR: interactive 3 and 4D images of high resolution neuroimage data." *The R journal*, **6**(1), 41. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4911196/.

Power JD, Barnes KA, Snyder AZ, Schlaggar BL, Petersen SE (2012). "Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion." *Neuroimage*, **59**(3), 2142–2154. doi:10.1016/j.neuroimage.2011.10.018.

Rinck P (2014). "Magnetic resonance: a critical peer-reviewed introduction." In *Magnetic resonance in medicine. The basic textbook of the European magnetic resonance forum,*, pp. 21–01.

Takagi Y, Sakai Y, Lisi G, Yahata N, Abe Y, Nishida S, Nakamae T, Morimoto J, Kawato M, Narumoto J, *et al.* (2017). "A neural marker of obsessive-compulsive disorder from whole-brain functional connectivity." *Scientific reports*, **7**(1), 7538. doi:10.1038/s41598-017-07792-7.

Zeng LL, Shen H, Liu L, Wang L, Li B, Fang P, Zhou Z, Li Y, Hu D (2012). "Identifying major depression using whole-brain functional connectivity: a multivariate pattern analysis." *Brain*, **135**(5), 1498–1507. doi:10.1093/brain/aws059.

Zheng W, Zhou X, Zou C, Zhao L (2006). "Facial expression recognition using kernel canonical correlation analysis (KCCA)." *IEEE transactions on neural networks*, **17**(1), 233–238. doi:10.1109/TNN.2005.860849.

Affiliation:

Jian Kang
Associate Professor
Biostatistics Department
Kidney Epidemiology and Cost Center
School of Public Health
University of Michigan, Ann Arbor
4055 SPH II, 1415 Washington Heights, Ann Arbor, Michigan 48109-2029
E-mail: jiankang@umich.edu
URL: http://www-personal.umich.edu/~jiankang/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd