Magic Pie (mgcpy)

Team Captain: Sambit Panda

Sandhya Ramachandran, Bear Xiong, Richard Guo, Satish Palaniappan, Ananya Swaminathan

Date: 9/12/2018

Sprint 1: Create *mgcpy* package

- Task 1: Implement dHSIC, HHG, Pearson, Spearman, and Mic into package (Sambit)
- Task 2: Implement MGC into the package (Satish)
- Task 3: Implement MDMR and FastMGC into package (Sandhya)
- Task 4: Implement MCORR, DCORR, and Mantel into package (Bear)
- Task 5: Implement Random Forest independence tests into package (Richard)
- Task 6: Implement 2- sample tests into package (Ananya)

Task 1 (Sambit): Implement dHSIC, HHG, Pearson, Spearman, and Mic into package

- Translated Pearson’s correlation data (RVCorr.m function in fastMGC) into python and sent pull request to development branch (pending approval)
- Tested code for the function (works for both lower dimensional and high dimensional tests)

Function is able to calculate correlations between distance matrices.

Matrix A

0	23	56	90	5	63	49
23	0	80	15	95	4	43
56	80	0	94	27	41	90
90	15	94	0	95	95	89
5	95	27	95	0	35	37

1. Compare A, A: **Correlation = 1.0**

Function is able to calculate correlations between distance matrices.

Matrix A

0	23	56	90	5	63	49
23	0	80	15	95	4	43
56	80	0	94	27	41	90
90	15	94	0	95	95	89
5	95	27	95	0	35	37

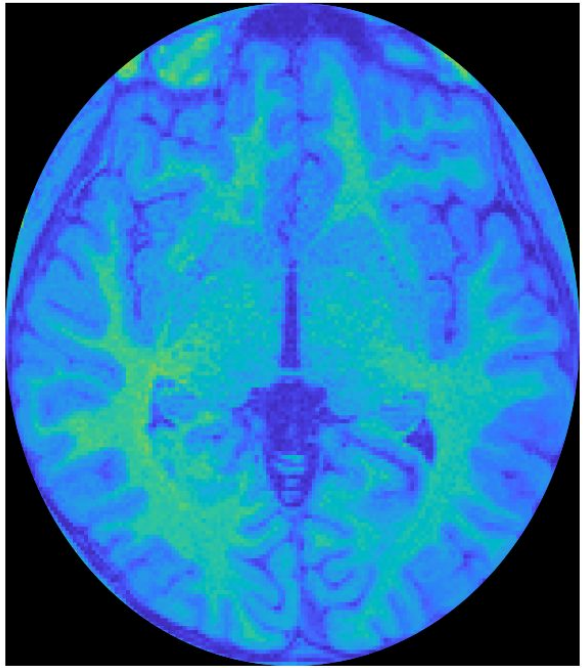
Matrix B

0	18	5	3	6	75	72
18	0	75	46	50	6	41
5	75	0	35	31	44	92
3	46	35	0	87	62	55
6	50	31	87	0	57	55

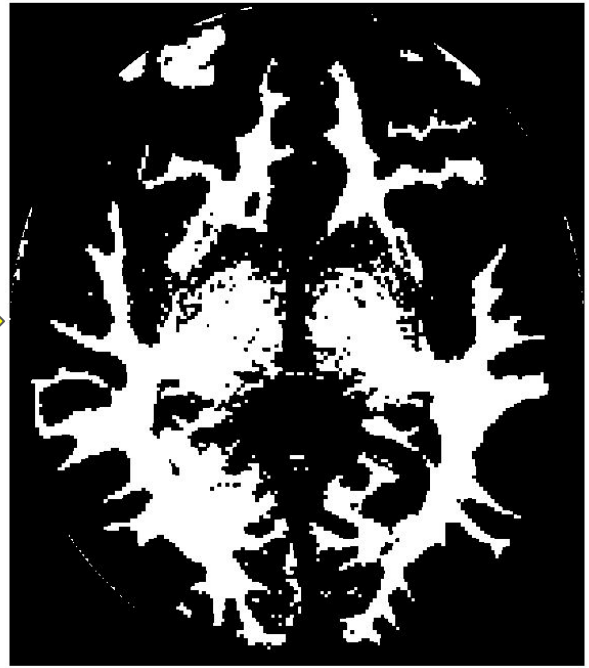
2. Compare A, B:
Correlation = 0.54479

Visualization of the week

sub-NDARAA536PTU_T1w.nii File



Cropped Section



Filtered White Matter

31.3%

White matter

Stuff to do this week

- More robust testing on higher dimensional data sets for RVCorr
- Begin working on dHSIC function (little bit more involved)
- Finalize PR with RVCorr function and merge with development branch
- Make RVCorr compliant with documentation guidelines

Task 2 (Satish): Implement MGC into the package

Last Week Accomplishments:

- Added the Apache 2.0 LICENSE file (and the batch to README.md)
- Added pre-commit git hooks to ensure PEP8 style guide is followed
  - Added a PEP8 badge as well
- Added draft version of the contribution guidelines (CONTRIBUTING.md)
- Setup code coverage (coveralls) and added a badge
- Implemented the DistRanks (in R MGC) method with unit test in *mgcpy*

All the changes above have been reviewed & merged to the *development* branch.

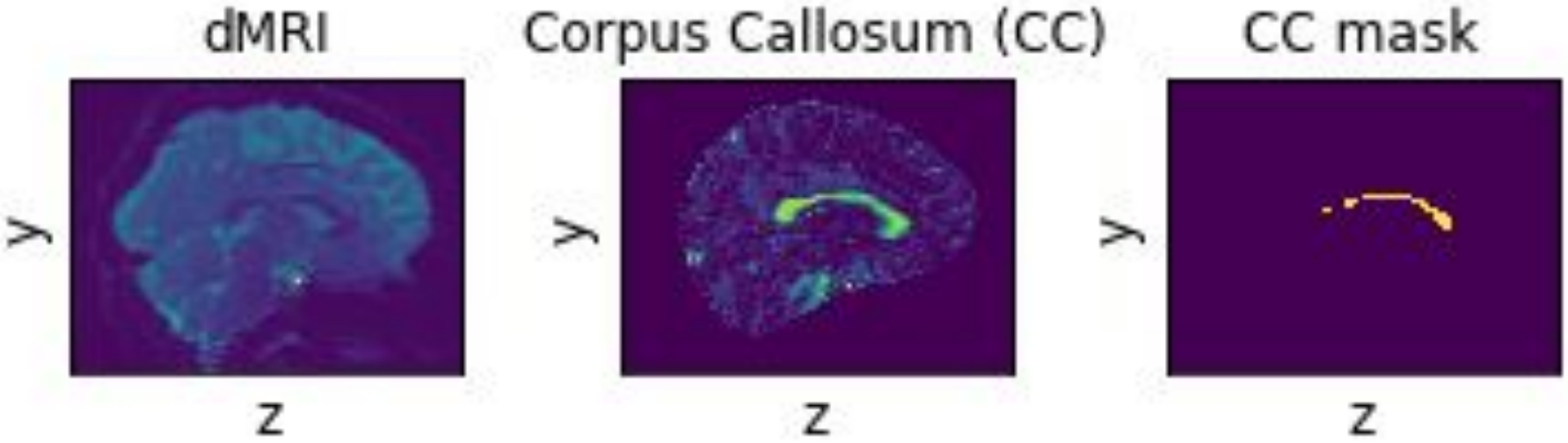
```r
> DistRanks <- function(dis) {
+     n=nrow(dis)
+     disRank=matrix(0,n,n)
+     for (i in (1:n)){
+         v=dis[,i]
+         tmp=rank(v,ties.method="min")
+         tu=unique(tmp)
+         if (length(tu)!=max(tmp)){
+             tu=sort(tu)
+             for (j in (1:length(tu))){
+                 kk=which(tmp==tu[j])
+                 tmp[kk]=j
+             }
+         }
+         disRank[,i]=tmp
+     }
+     return(disRank)
+ }
```

```python
def rank_distance_matrix(distance_matrix):
    """
    Sorts the entries within each column in ascending order

    For ties, the "minimum" ranking is used, e.g. if there are
    repeating distance entries, The order is like 1,2,2,3,3,4,...

    :param distance_matrix: a symmetric distance matrix.
    :return: column-wise ranked matrice of ``distance_matrix``
    """
    n_rows = distance_matrix.shape[0]
    ranked_distance_matrix = np.zeros(distance_matrix.shape)
    for i in range(n_rows):
        column = distance_matrix[:, i]
        ranked_column = np.array(scipy.stats.rankdata(column, "min"))
        sorted_unique_ranked_column = sorted(list(set(ranked_column)))
        if (len(ranked_column) != len(sorted_unique_ranked_column)):
            for j, rank in enumerate(sorted_unique_ranked_column):
                ranked_column[ranked_column == rank] = j + 1
        ranked_distance_matrix[:, i] = ranked_column
    return ranked_distance_matrix
```

```python
def test_rank_distance_matrix():
    a = np.array([[1, 4, 6],
                  [2, 5, 7],
                  [1, 4, 6]])
    ranked_a = np.array([[1, 1, 1],
                         [2, 2, 2],
                         [1, 1, 1]])
    assert np.array_equal(rank_distance_matrix(a), ranked_a)
```

Visualization of the week

Extracting the Corpus Callosum of Subject-NDARBN100LCD (using *dipy*)



Stuff to do this week

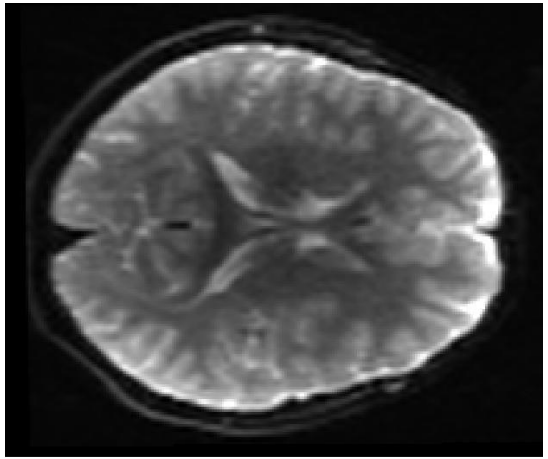
- Study the pseudo code in the MGC paper and the R code, and list down all the functions and its dependencies required to port to Python
- Create stubbed versions of all the above functions and add proper docstrings to define the inputs and outputs
- Implement one of the above function with tests in Python and raise a PR

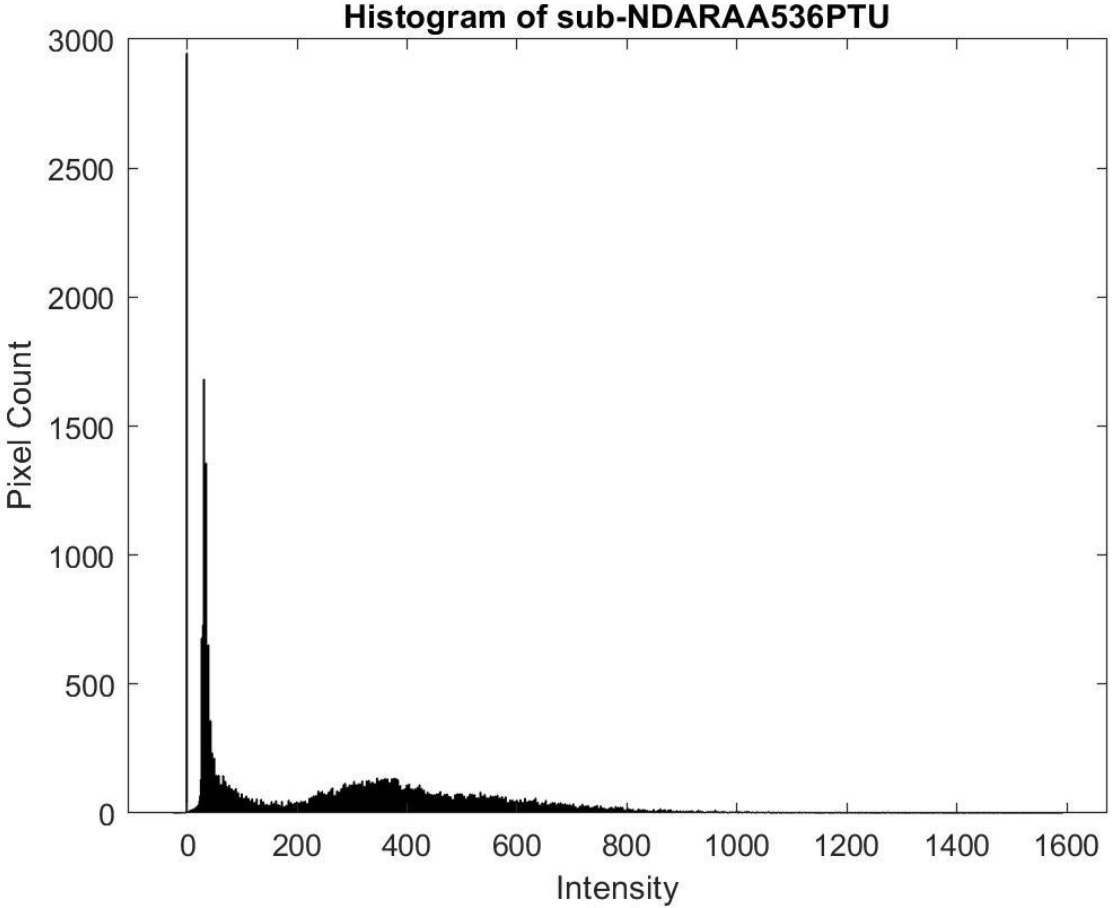
Task 3 (Sandhya): Implement MDMR and FastMGC into package

Last Week's Accomplishments:

- Attempted to run cython version of MDMR
  - Hard on windows: visual studios build issue, vcvarsall.bat
  - In general more accessible if not in cython
- Began to create a python version of MDMR (previously cython)
  - All base functions running in python
  - Mdmr function runs, but unclear input "columns"
- Attempted to contact author- no response yet
- Read up on MDMR from R edition's documentation

Visualization of the Week

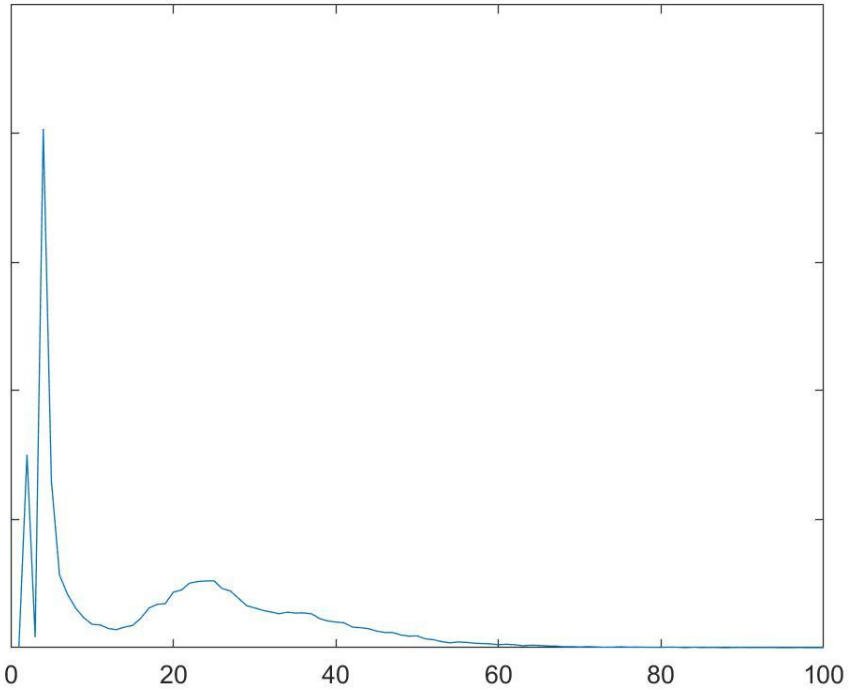


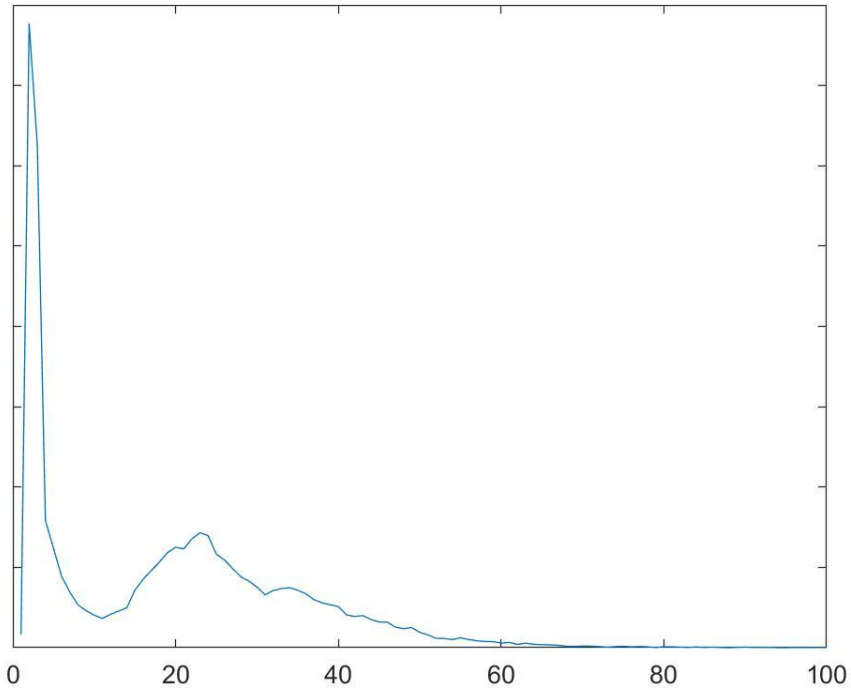


- sub-NDARAA536PTU
- Registered DWI

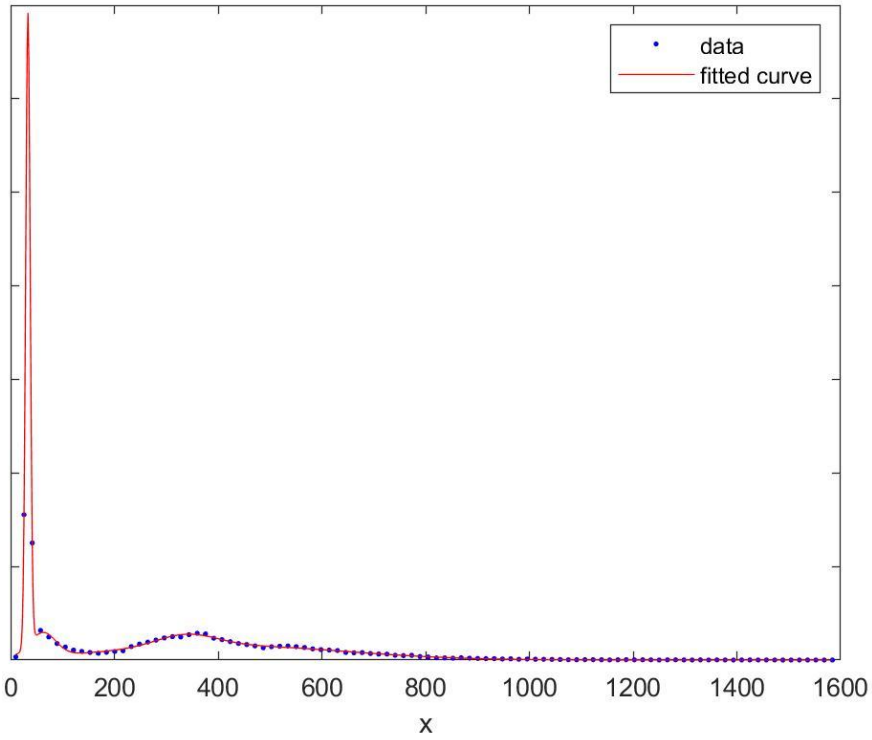
Background Strip:

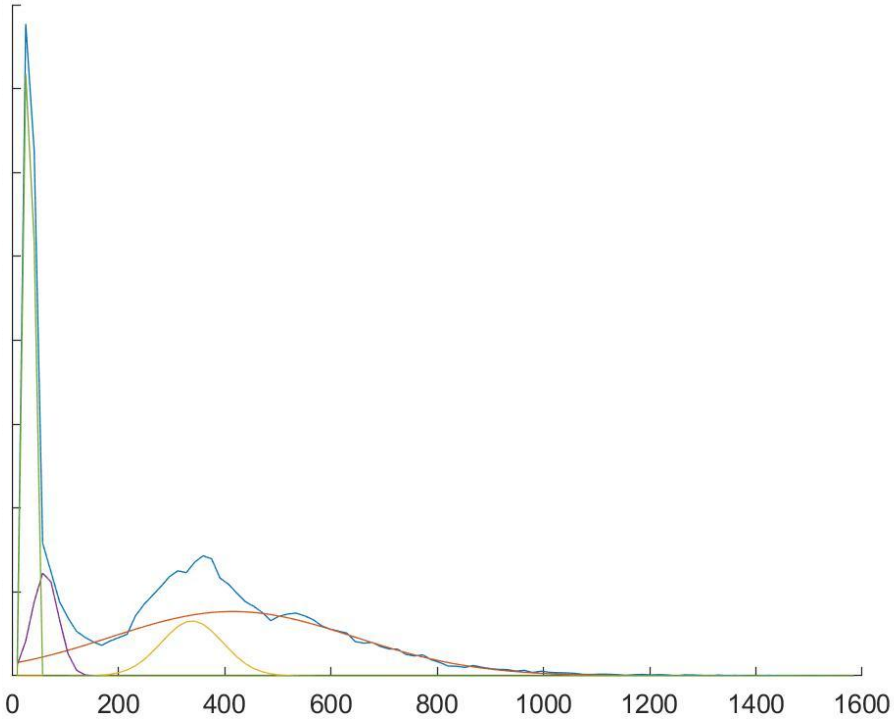
```
load('a.mat')
a = double(a(a>0));
[p,center] = hist(double(a(a>min(a(:)))),100);
p = p/sum(p);
figure; plot(p);
fitobject = fit(center',p','gauss4');
figure; plot(fitobject, center', p');
```





Gaussian Mixture Model





Next week:

- Convert remaining MDMR code to python
- Contact author of MDMR code for help
- If successful, run MDMR on spiral data, linear data
- If successful, convert MDMR code to our coding structure/documentation
    - Continue learning how git works and what our coding structure is from my teammates/google

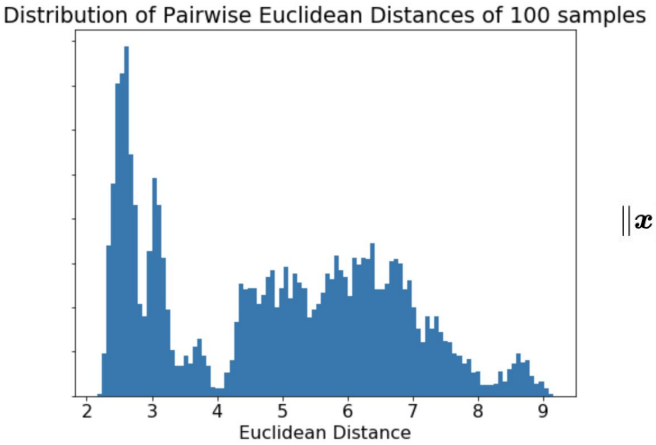
Task 4 (Bear): Implement MCORR, DCORR, and Mantel into package

Last Week's Accomplishments:

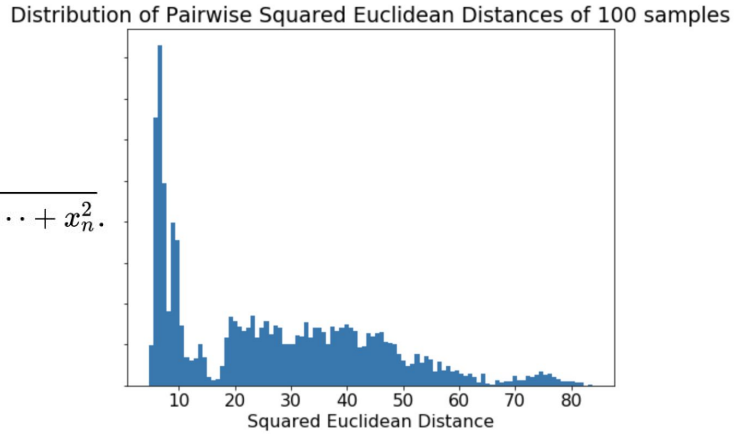
- Implemented DCORR. Results matched matlab code on simulated data
  - Specifics of simulated data: all 20 types of dependence, X, Y are 100 examples and 1 dimensional
  - DCORR test statistics matched original results with less than 1e-5 absolute error
- Initiated connectome preprocessing pipeline
  - n Edgelists → n d by d Adjacency matrices → one n by p data matrix (p is the dimension of each example)
  - Most naive approach: flatten the matrices!
    - 48 x 48 adjacency matrix → 2304-dimensional vector

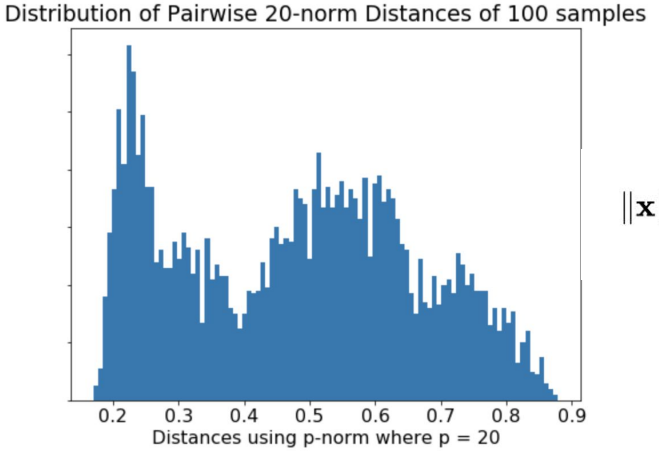
Pairwise distances between connectome graphs

- 100 “task-rest_bold_JHU_res-2x2x2_measure-correlation.edgelist” examples

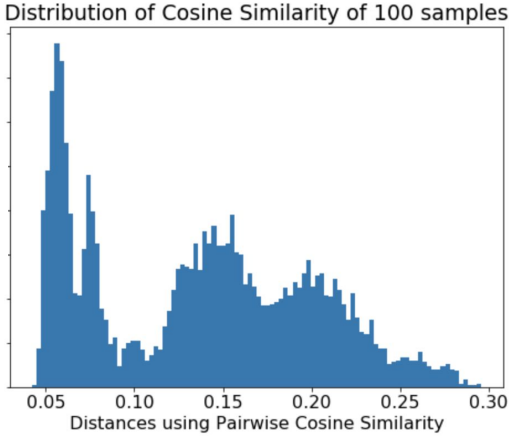


$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_n^2}.$$





$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$



$$1 - \frac{u \cdot v}{||u||_2 ||v||_2}$$

Next week

- Merge DCORR, MCORR, Mantel modules (with unit tests) into mgcpy
  - DoD: Closed PR
- Potential new tasks
  - Port simulation code into mgcpy
    - DoD: Scatter plots of different dependency
  - Build connectome-to-correlation-test data pipeline
    - DoD: Module which converts edgelist files into formats usable for mgcpy functions

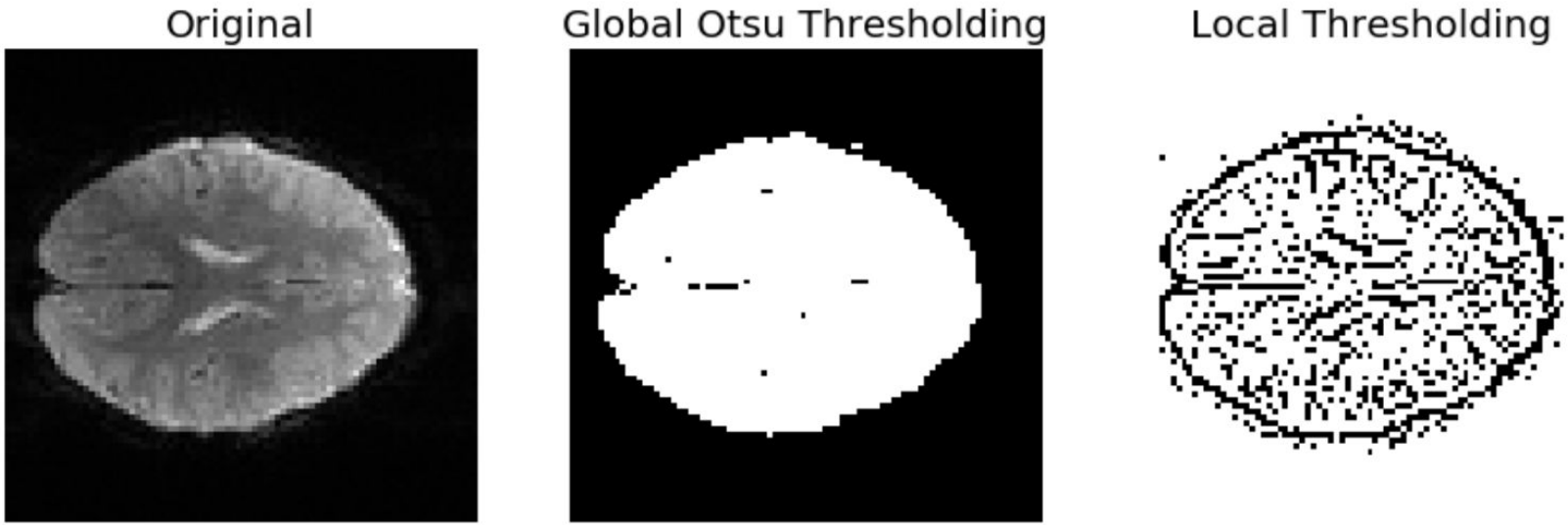
Task 5 (Richard): Random Forest Based Independence Test

Last Week’s Accomplishments:

- Setup travis-ci (merged into dev)
- Setup sphinx for documentation (PR ready)

Visualization of the week:

fMRI of Subject NDARMR277TT7 Watching Despicable Me



Next week

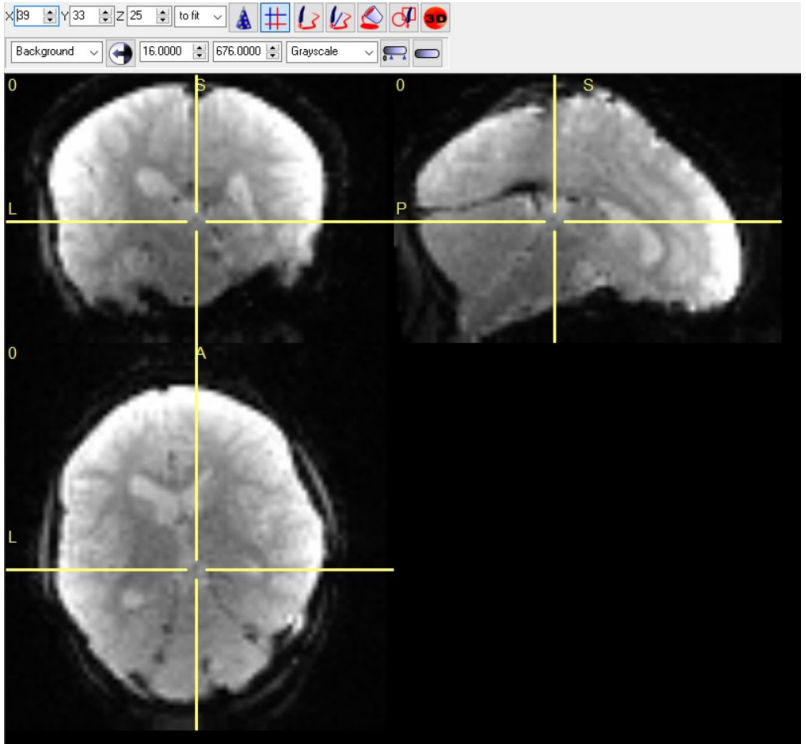
- Detailed guide on how to generate documentation and overall github workflow
  - DoD: docs in sphinx
- Pseudocode on using random forest to estimate conditional entropy
  - DoD: LaTeX write up
- Reach: Implement algorithm using scikit learn random forest embedding code
  - DoD: Jupyter Notebook showing results on easy simulated dataset

Task 6 (Ananya): Implement 2-sample tests into package

Last Week’s Accomplishments:

- Read and took notes on “Equivalence of Distance and Kernel Methods” to learn more about ENERGY and MMD

Visualization of the Week



Resting state fMRI of subject NDARMN450PEH

TP = 12

Next week

- Run ENERGY
  - DoD: Output of ENERGY
- Run MMD
  - DoD: Output of MMD

This branch is 25 commits ahead of master. Pull request Compare

tpsatish95 Merge pull request #29 from NeuroDataDesign/satish ··· Latest commit 6345471 a day ago

docs	Add docs folder	4 days ago
git-hooks	Add pre-commit hooks to ensure pep8	2 days ago
mgcpy	Add rank_distance_matrix function	2 days ago
.gitignore	Initialize the folder structure	2 days ago
.travis.yml	Add code coverage badge	2 days ago
CONTRIBUTING.md	Add contribution guidelines	2 days ago
LICENSE	Add LICENSE	4 days ago
README.md	Add code coverage badge	2 days ago
install-hooks.sh	Add pre-commit hooks to ensure pep8	2 days ago
requirements.txt	Add code coverage badge	2 days ago
setup.cfg	Add pre-commit hooks to ensure pep8	2 days ago

README.md

mgcpy

coverage 100% | build unknown | code style pep8 | license Apache 2.0

`mgcpy` is a Python package containing tools for multiscale graph correlation and other statistical tests, that is capable of dealing with high dimensional and multivariate data.

License

This project is covered under the **Apache 2.0 License**.

Collective Team Task:

Last week: Completed first draft of AWS proposal

This week: Visualization of data, Better second draft of AWS proposal, Make PR on something in repo