



# Magic Pie (mgcpy)

Team Captain: Sambit Panda

Sandhya Ramachandran, Bear Xiong,  
Richard Guo, Satish Palaniappan

Date: 9/12/2018

# Visualization: edgelist

## Edgelist

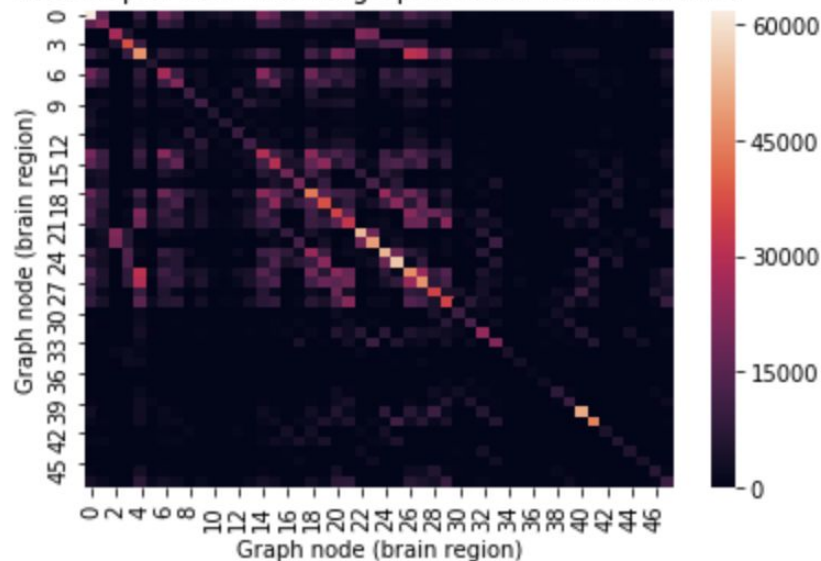
- Graph node with edge weight indicating the strength of correlation between two nodes
- Use to represent connectome
- sub-NDARAA536PTU\_acq-64dir\_dwi\_JHU\_res-1x1x1\_measure-spatial-ds.edgelist

42	42	44966
42	29	1707
42	24	1328
42	25	859
42	26	11215
42	27	2818
42	20	5116
42	21	2319
42	22	5459
42	23	4
42	47	48
42	28	6999
42	43	1

## Observation

- Diagonals
- Graph nodes with smaller indices (Interpretation?)

Heatmap of connectome graph of sub-NDARAA536PTU



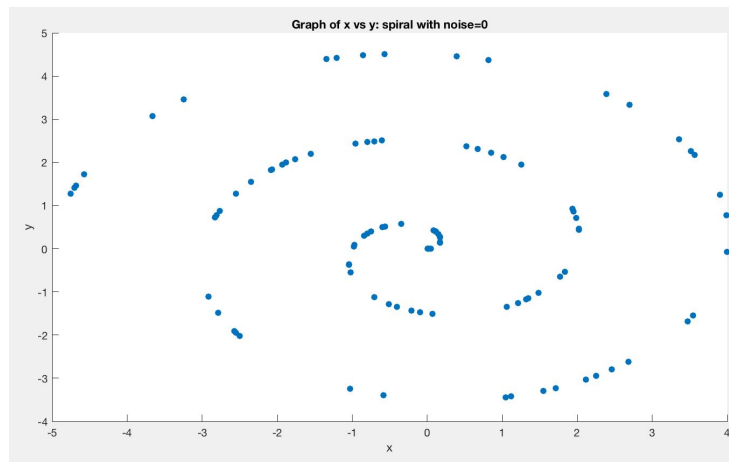
# Tool: mcorr, dcorr, mantel

- Input: data matrix (n by d)
- Process
  - Calculate distance matrix
  - Use distances between all pairs (global correlation) → can identify correlation on linear data, but not on nonlinear data
- Output: correlation coefficient between -1 and 1
  - 1: perfect positive correlation
  - 0: uncorrelated

```
>> [all_corr] = run_tools(8, 100, 1, 1, 0)
```

```
all_corr =
```

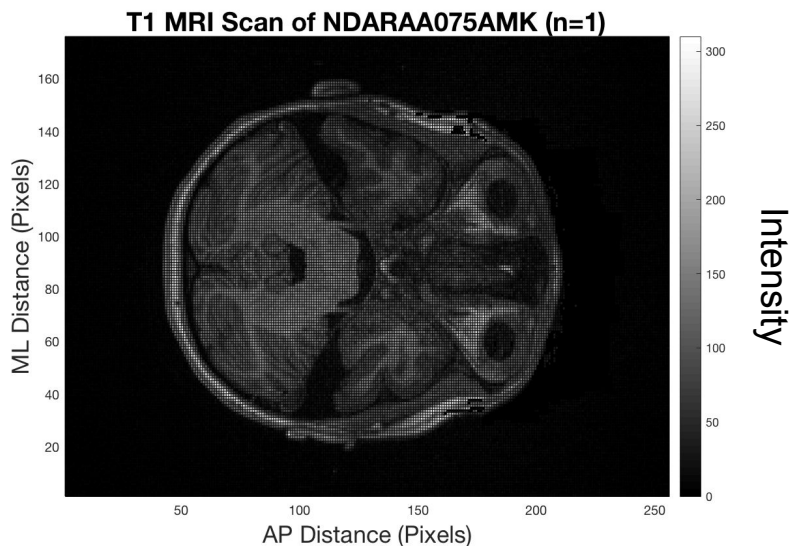
```
0.0343    0.0185    0.0345
```



```
% type: the type of data: linear, jointly normal, independent, etc.
% n: the number of data points generated
% dim: the dimensionality of the data
% dependent: indicating whether the two variables in the data are dependent
% noise: additive noise
% return corr: a number from -1 to 1, 1 indicating perfect positive correlation
% 0 indicating no correlation
function [all_corr] = run_tools(type,n,dim,dependent, noise)
    all_corr = zeros(3, 0);
    tool = ['mcorr', 'dcorr', 'mantel'];
    for i = 1:3
        [x, y] = CorrSampleGenerator(type,n,dim,dependent, noise);
        scatter(x, y, 'filled');
        xlabel('x');
        ylabel('y');
        title('Graph of x vs y: spiral with noise=0');
        [corr, ~, ~] = DCorr(x, y, tool(i));
        all_corr(i) = corr;
    end
end
```

# Visualization of HBN

- Video shows horizontal sections of slices as it goes inferior to superior
- Right side is anterior and left side of posterior



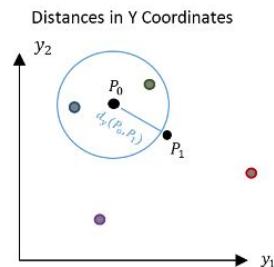
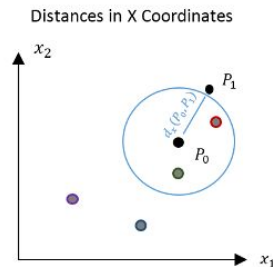
T1 MRI of NDARAA075MK (n=1) Animation

ML Distance (Pixels)

AP Distance (Pixels)

# Tool: HHG (Heller-Heller-Gorfine Tests Of Independence And Equality Of Distributions)

- Input: 2 distance matrices
- Output: HHG test statistic
  - HHG space partitioned by taking two points and seeing how close other points are in space relative to those two points
  - Chi square statistic is calculated from 2x2 matrix for each point
  - HHG stat is sum of sum all the the chi square statistics
- After: HHG stat run through permutation independence tests



$$S(i, j) = \frac{(N - 2)(A_{12}A_{21} - A_{11}A_{22})^2}{A_{1.}A_{2.}A_{.1}A_{.2}}$$

$$T = \sum_{i=1}^N \sum_{j=1, j \neq i}^N S(i, j)$$

```
>> HHG(A, B)
```

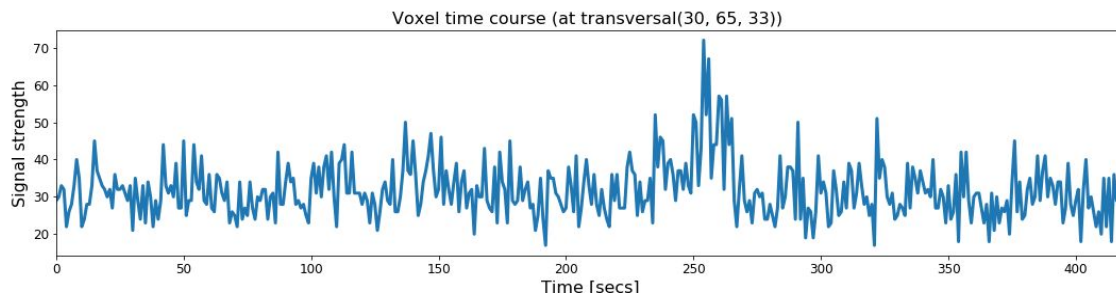
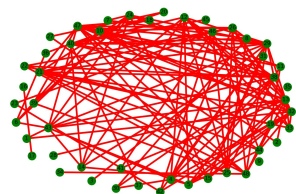
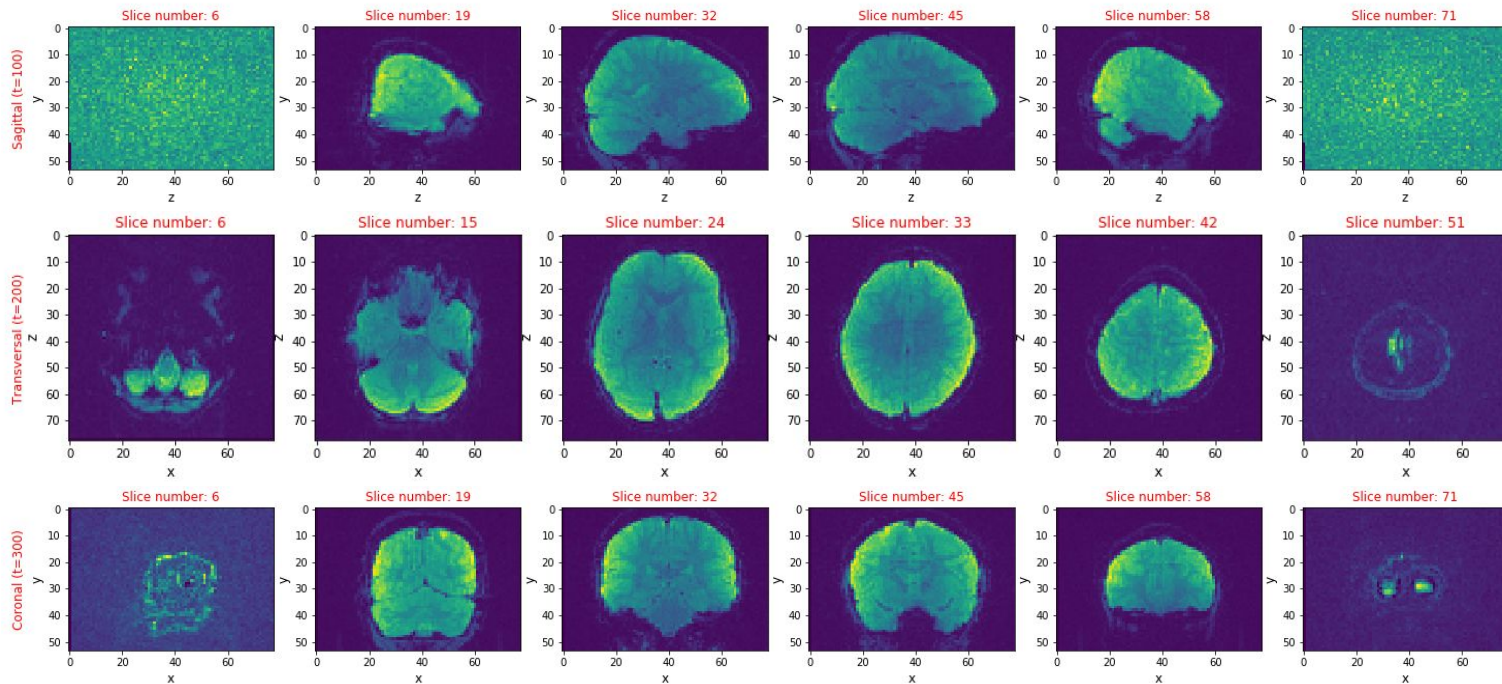
```
ans =
```

```
2.4963e+04
```

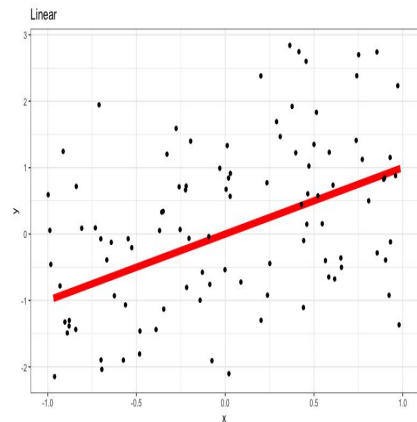
	$d_y(P_0, \cdot) \leq d_y(P_0, P_1)$	$d_y(P_0, \cdot) > d_y(P_0, P_1)$	
$d_x(P_0, \cdot) \leq d_x(P_0, P_1)$	$A_{11}(P_0, P_1)$	$A_{12}(P_0, P_1)$	$A_{1.}(P_0, P_1)$
$d_x(P_0, \cdot) > d_x(P_0, P_1)$	$A_{21}(P_0, P_1)$	$A_{22}(P_0, P_1)$	$A_{2.}(P_0, P_1)$
	$A_{.1}(P_0, P_1)$	$A_{.2}(P_0, P_1)$	$N - 2$

Table 1: The observed counts in an HHG partition of the sample space.

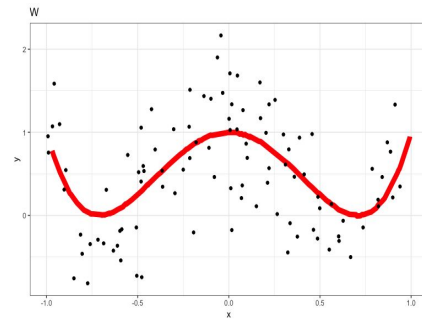
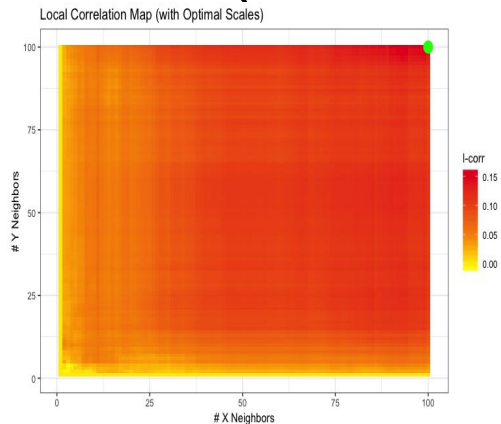
# Visualization of HBN Data (fMRI of Subject-NDARBN100LCD)



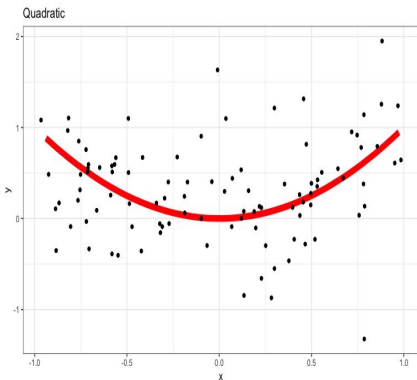
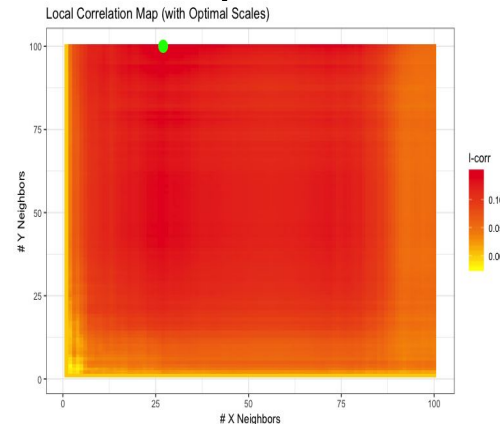
# Tool: MGC (Multiscale Graph Correlation)



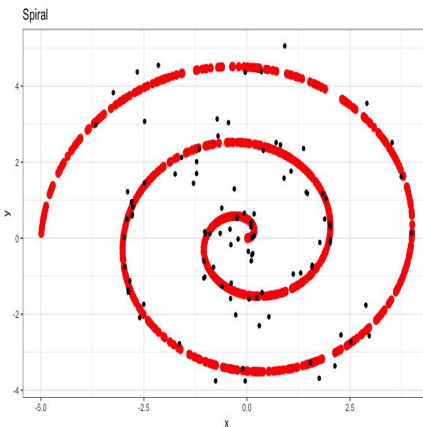
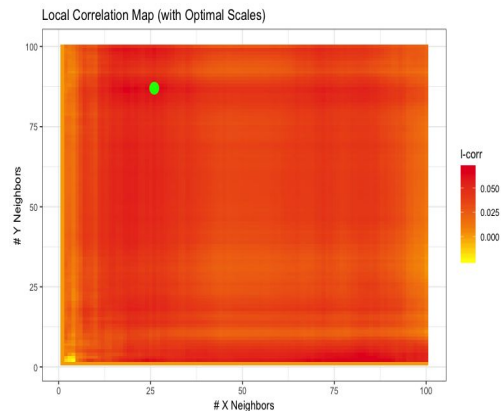
**Linear: pMGC = 0 and statMGC = 0.1422364**



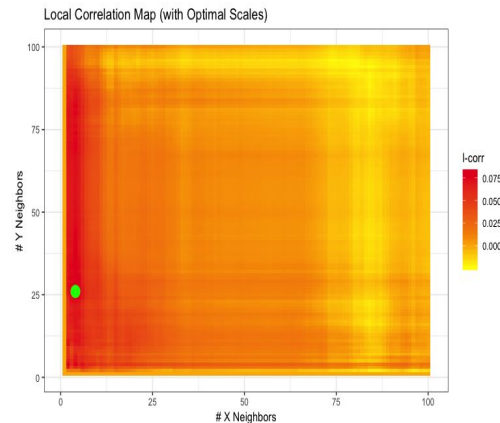
**W: pMGC = 0 and statMGC = 0.1486704**



**Quadratic: pMGC = 0.05 and statMGC = 0.06660724**

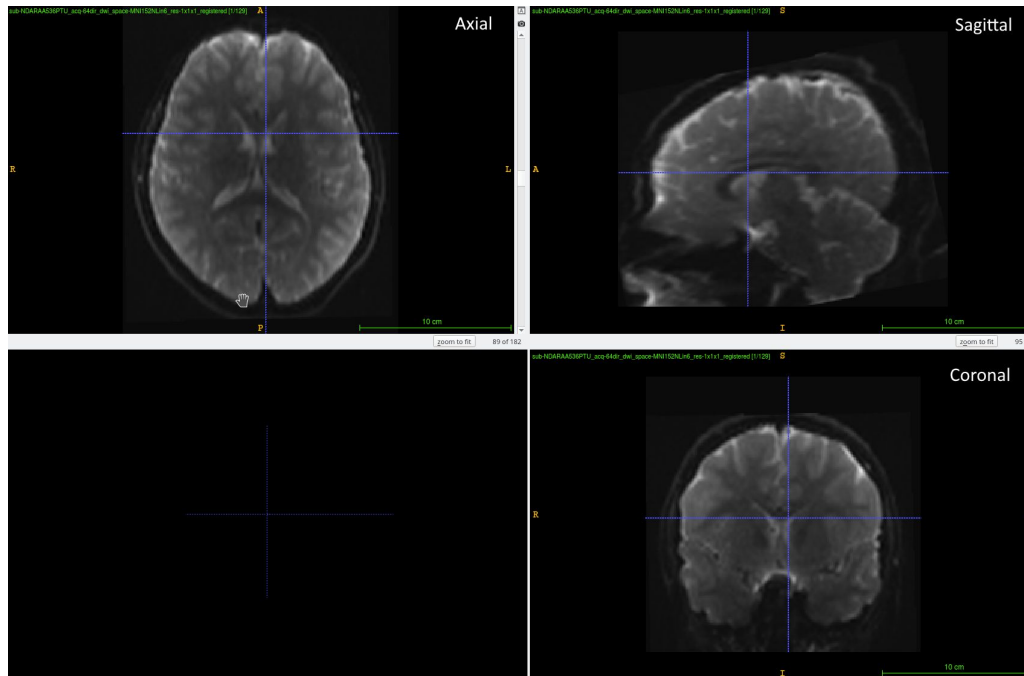


**Spiral: pMGC = 0.05 and statMGC = 0.08159789**





# Visualization of HBN: ITK-SNAP



- Visualization of the dwi MRI of sub NDARAA536PTU, registered, and represented in greyscale
- Axial, sagittal, and coronal views shown
- Shows differentiation between white and grey matter, ventricles



# Tool: fastmgc used on demo data

100x100 double

	1	2	3	4	5	6
1	0	2.2038	3.1008	0.5331	4.4869	0.1713
2	2.2038	0	5.3046	2.7369	6.6907	2.3750
3	3.1008	5.3046	0	2.5677	1.3861	2.9296
4	0.5331	2.7369	2.5677	0	3.9538	0.3618
5	4.4869	6.6907	1.3861	3.9538	0	4.3157
6	0.1713	2.3750	2.9296	0.3618	4.3157	0

100x1 double

	1
1	1.7872
2	3.9910
3	-1.3136
4	1.2541
5	-2.6997
6	1.6159
7	-1.9499
8	-2.9204
9	-2.8747
10	-1.0310

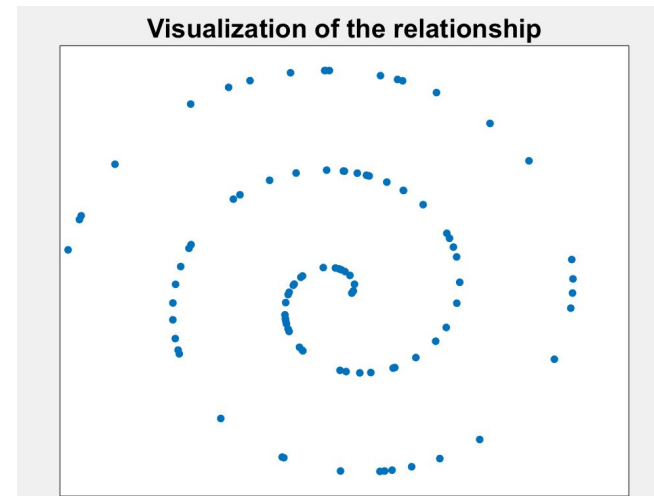
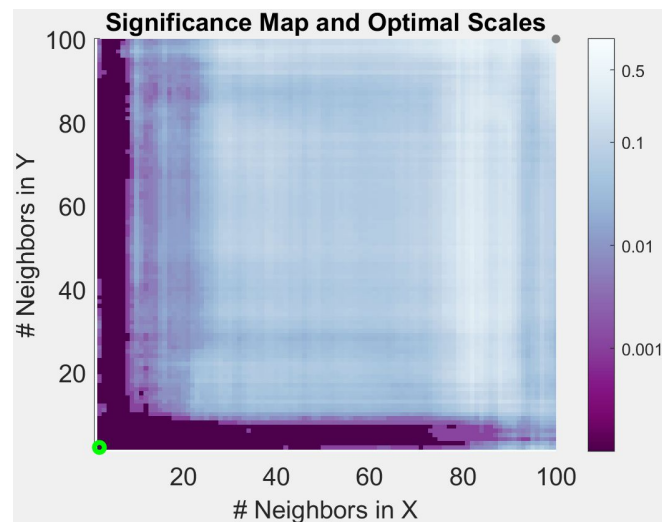
Command Window

```
1x1 cell array

{'The p-value and test statistic of MGC are 0 and 0.22509'}

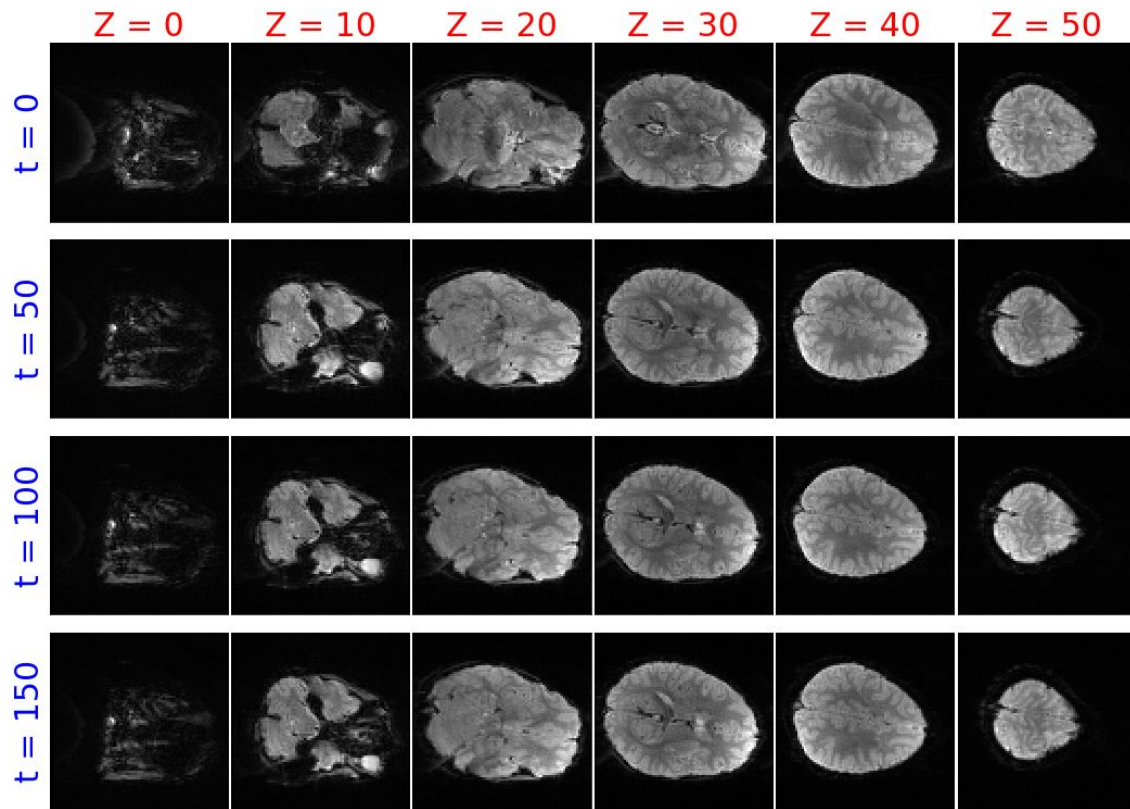
For p-value, the smaller the more significant
fx>>
```

- A and B are 100x100 distance matrices calculated for sample data
- Finds a perfect relationship (artificial data, expected) despite the data being nonlinear



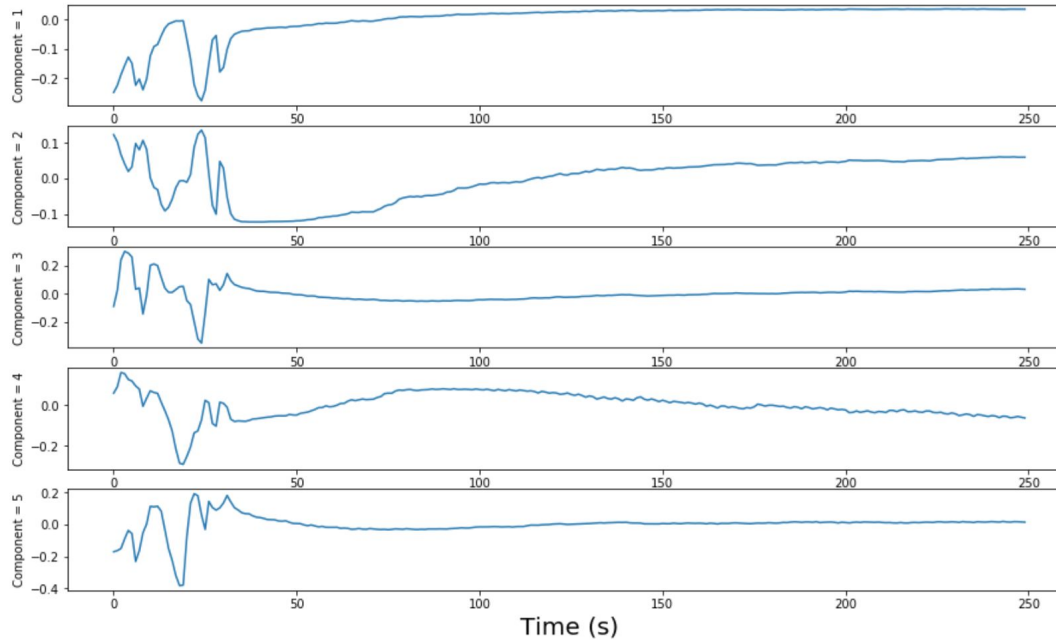
# Looking at it

fMRI of Subject-NDARAD224CRB During Movie Watching

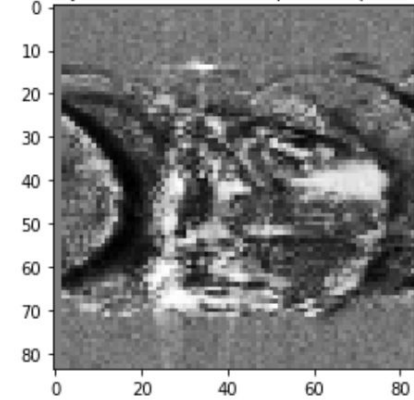


# Exploratory Data Analysis

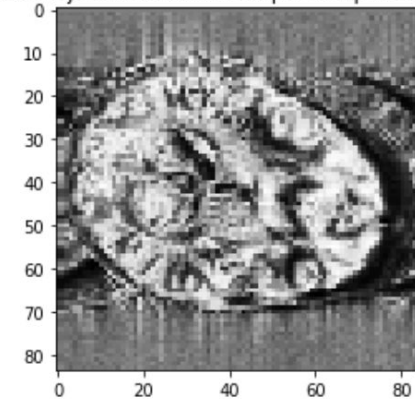
## PCA on the time dimension



Basis Projection of First Principal Component  $z = 0$

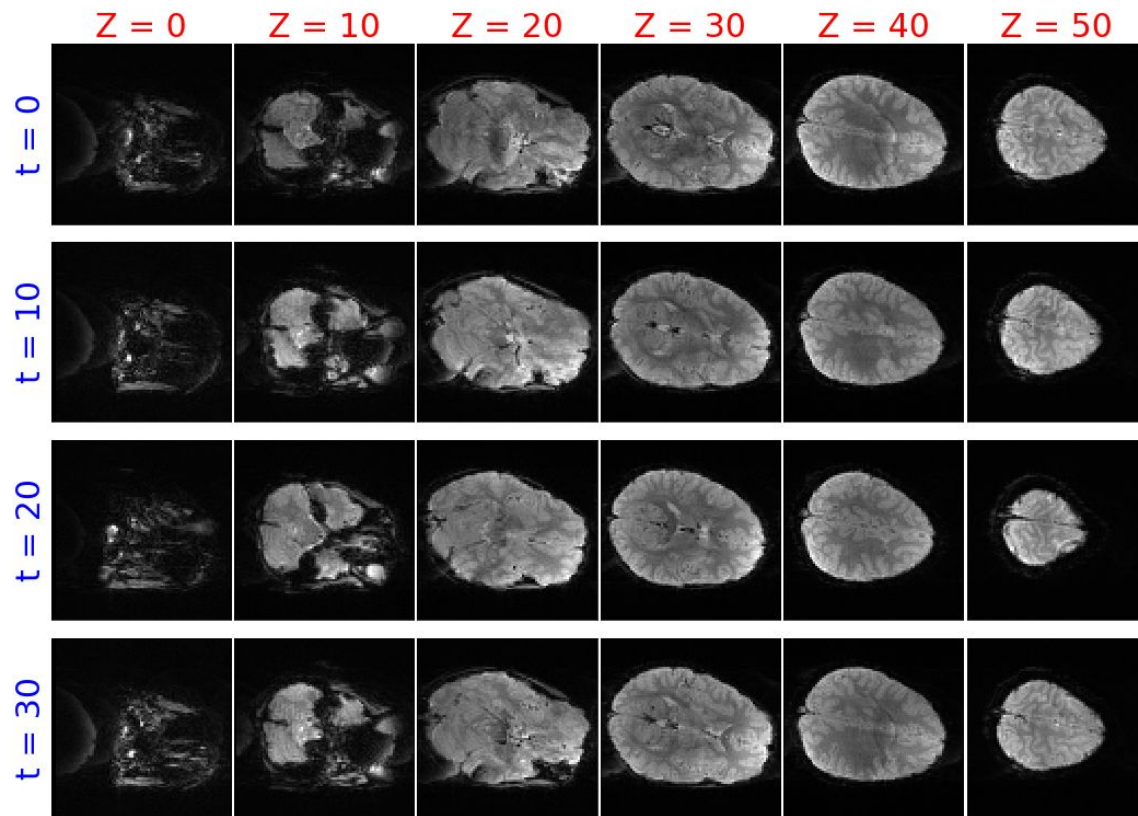


Basis Projection of First Principal Component  $z = 30$

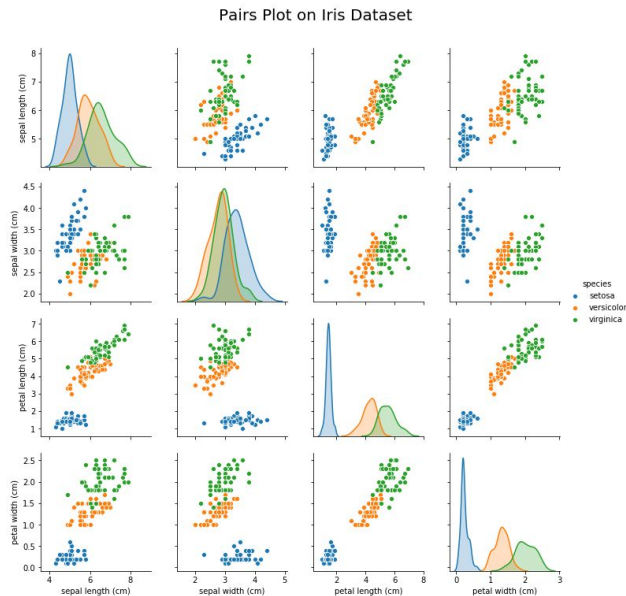


# Looking more at fMRI during the start of the task

fMRI of Subject-NDARAD224CRB During Movie Watching



# Making sure I can run Random Forest



```
In [106]: from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(df[iris.feature_names], iris.target, test_size=0.4)

rf = RandomForestClassifier(n_estimators=20)
rf.fit(X_train, y_train)

predictions = rf.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print(f'Accuracy score: {accuracy}')
```

Accuracy score: 0.9666666666666667

# Acknowledgements

- Jovo
- Vikram
- Healthy Brain Network people