GRETNA

Sandy Wang

February 11, 2016

Contents

1 Overview

GRETNA toolbox has been designed for the **graph-theoretical network analysis** of fMRI data. It is a suite of MATLAB functions and some MATLAB-based Interface to perform the process of conventional fMRI preprocessing, as well as to calculate most frequently used network metrics, like small world, efficiency, degree, betweenness, assortativity, hierarchy, synchronization and modularity.

2 Licence

GRETNA is distributed under the terms of the GUN General Public Licence as published by the Free Software Foundation (version 3) and the details on ”copyleft” can be found at http://www.gnu.org/copyleft/.

3 Prerequisites

You need the following to run **GRETNA** on your computer:

- MATLAB: A high level numerical mathematics environment developed by MathWorks, Inc. Natick, MA, USA. GRETNA requires MATLAB2010a or later version
- SPM8: SPM is made freely available to the (neuro)imaging community, to promote collaboration and a common analysis scheme across laboratories. The software represents the implementation of the theoretical concepts of Statistical Parametric Mapping in a complete analysis package.
- MRICroN's dcm2nii: GRETNA included this package in its distribution. So you do not need download MRICroN's dcm2nii again.
- MatlabBGL: MatlabBGL is a MATLAB package for working with graphs. It uses the Boost Graph Library to efficiently implement the graph algorithms. GRETNA included this package in its distribution. So you do not need download MatlabBGL again.
- PSOM: The pipeline system for GNU Octave and Matlab(r) (PSOM) is a lightweght library to manage complex multi-stage data processing. A pipeline is a collection of jobs, i.e. Matlab or Octave codes with a well identified

set of options that are using files for inputs and outputs. GRETNA included this package in its distribution. So you do not need download PSOM again.

4 Installation

Warning: Please ensure your GRETNA path do not include blank!.

4.1 Command-line

If you do not have write permission for the path of GRETNA, please add GRETNA to MATLAB's path with the following command every time you launch MATLAB:

```
>> addpath(genpath("/usr/local/share/software/GRETNA"));
```

Where **"/usr/local/share/software/GRETNA"** is the location of your GRETNA.

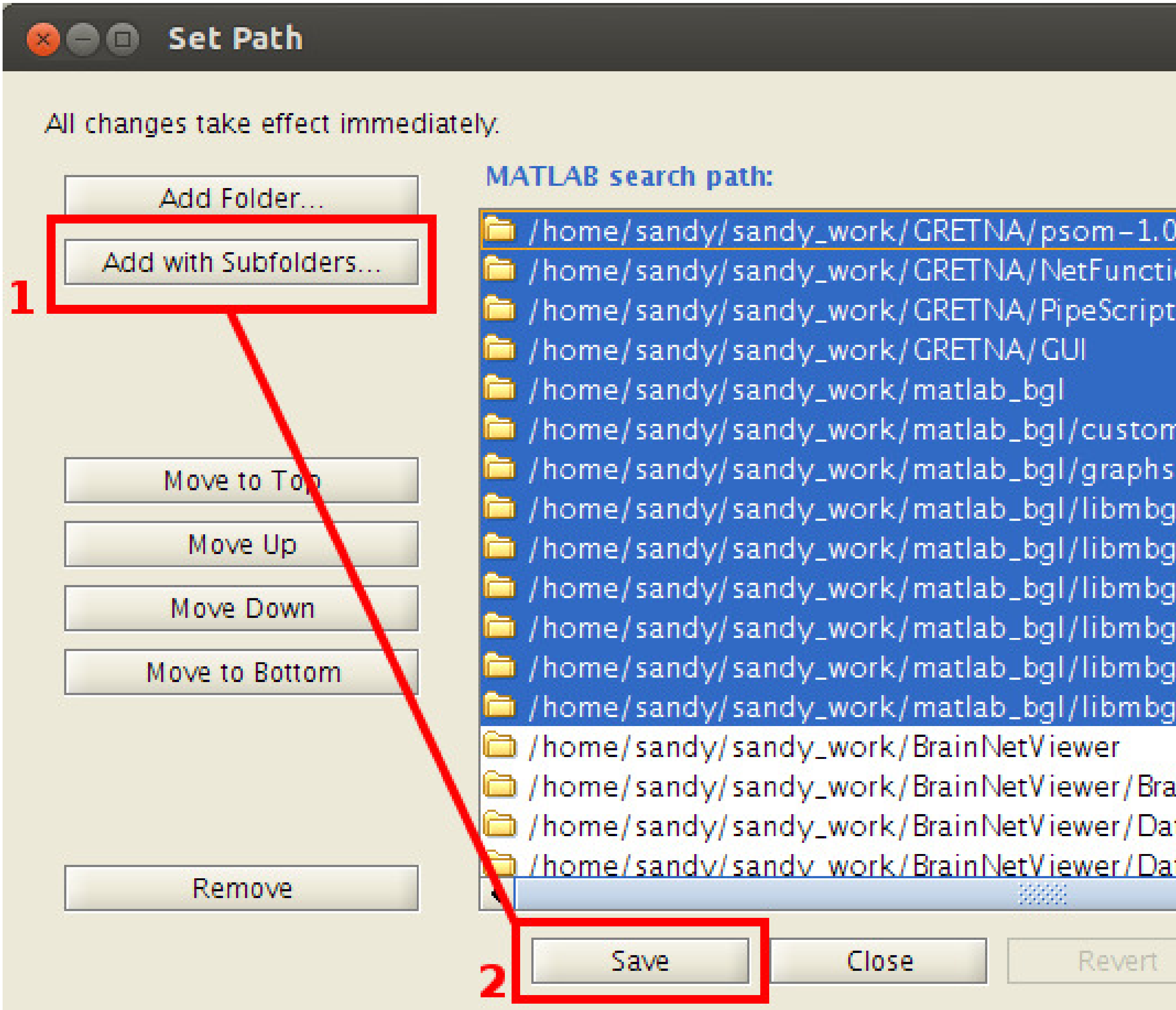
4.2 Interface

To ensure GRETNA is automatically on your MATLAB path in the future, you need launch MATLAB and run the following command to manage your MATLAB path:

```
>> pathtool;
```

Then:

1. Click "Add with Subfolders..." button and select your GRETNA path, i.e. "/usr/local/software/GRETNA"



All changes take effect immediately.

MATLAB search path:

```
/home/sandy/sandy_work/GRETNA/psom-1.0
/home/sandy/sandy_work/GRETNA/NetFuncti
/home/sandy/sandy_work/GRETNA/PipeScript
/home/sandy/sandy_work/GRETNA/GUI
/home/sandy/sandy_work/matlab_bgl
/home/sandy/sandy_work/matlab_bgl/custom
/home/sandy/sandy_work/matlab_bgl/graphs
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/matlab_bgl/libmbg
/home/sandy/sandy_work/BrainNetViewer
/home/sandy/sandy_work/BrainNetViewer/Bra
/home/sandy/sandy_work/BrainNetViewer/Da
/home/sandy/sandy_work/BrainNetViewer/Da
```

2. Save your change. If you do not have the permission to save your change in GRETNA folder, please save **pathdef.m** to another location where you will often launch MATLAB.

5 Starting

GRETNA include three sub-module:

1. Network Construction: The fMRI pre-processing and the estimation of ROI-based correlation matrix.
2. Network Analysis: The estimation of network and nodal metrics.
3. Network Comparison: The statistics of network, nodal and edge metrics.

you open GRETNA interface by

```
>> gretna;
```

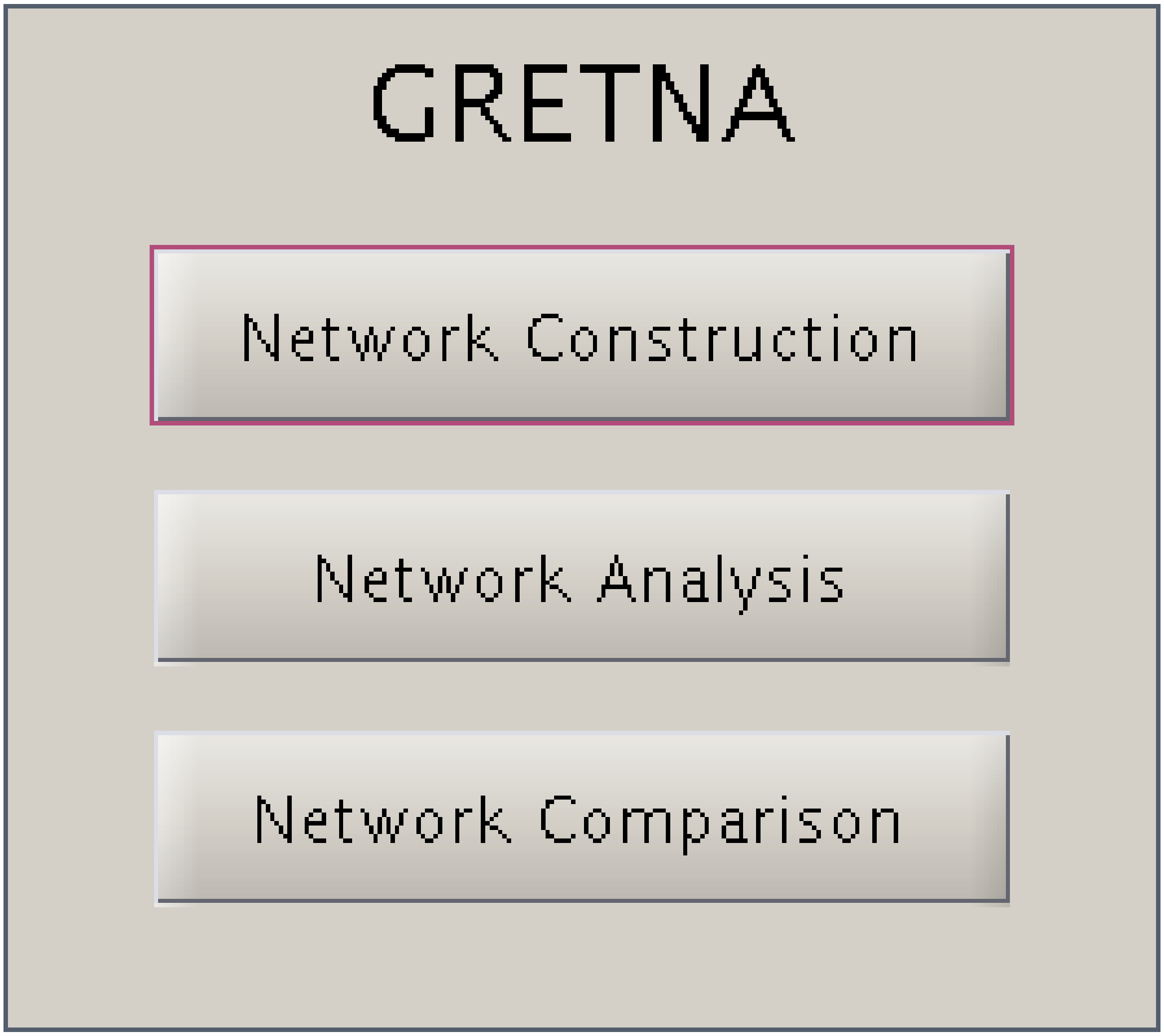
6 Toolbar

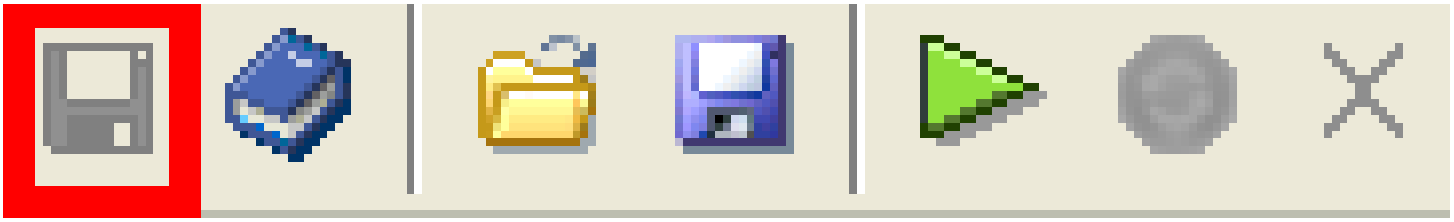
6.1 Save Default Configure

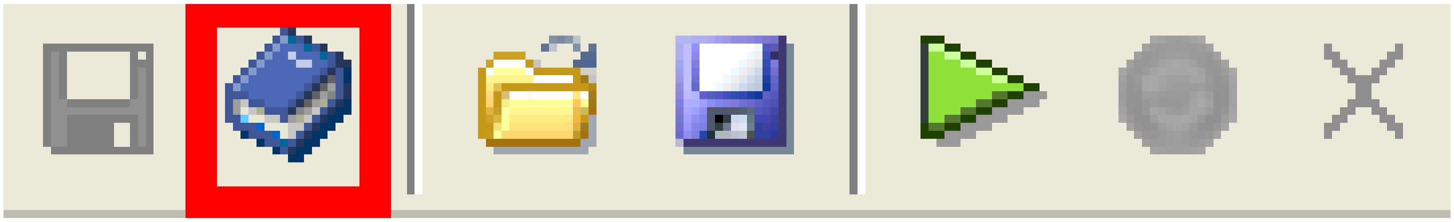
You can save your customized configuration for **GRETNA**, it will be the configure you want every time you open **GRETNA**.

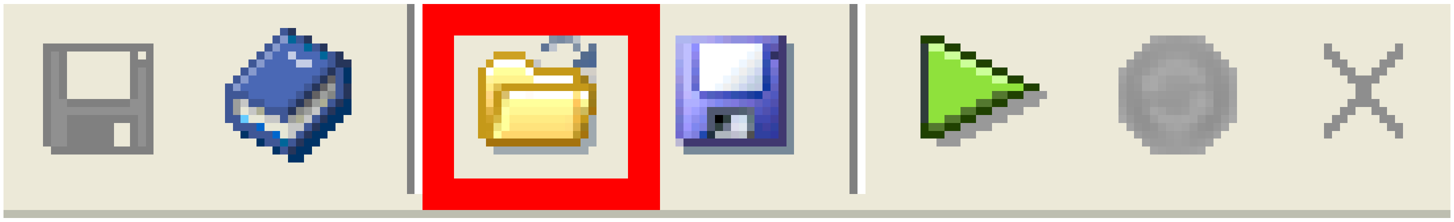
6.2 Manual

Click to open this manual.


GRETNA
Network Construction
Network Analysis
Network Comparison







6.3 Load Configure

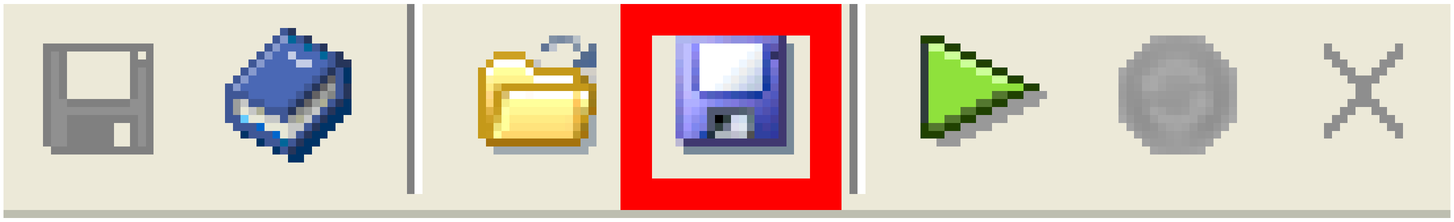
To load the configuration you save.

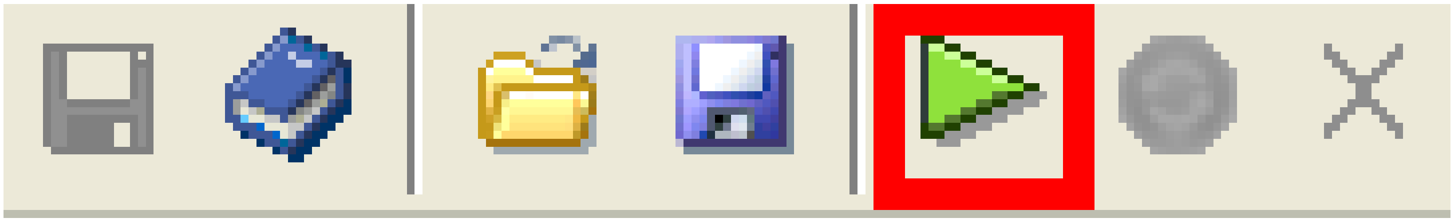
6.4 Save Configure

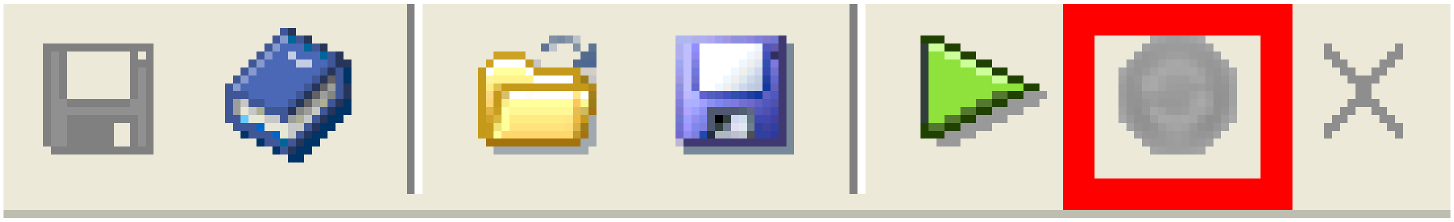
To save the current configuration.

6.5 Run

To run the pipeline.





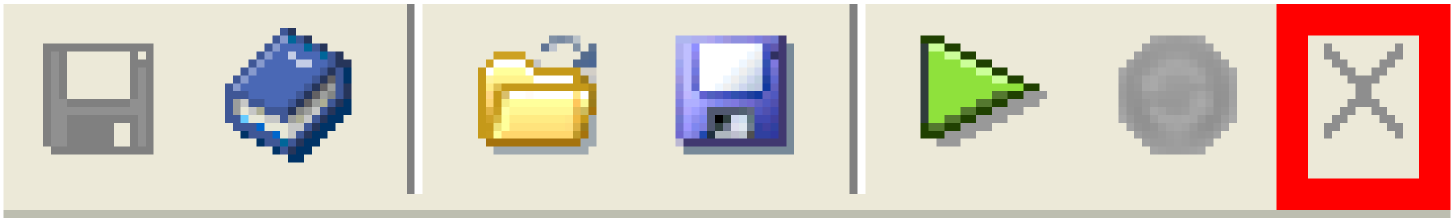


6.6 Refresh

As usual, the status of pipeline will be refreshed automatically. If you found any error message in MATLAB's command-line window, that is **not** the errors for pipeline but the errors for the refresh process. Please click this button to get the status of pipeline.

6.7 Stop

Before you close the interfaces of **GRETNA**, **please** click this button **if the pipeline is still running**.



7 Input Format

For **GRETNA**, you can import functional image sequences to get network matrices, or you can import network matrices directly.

7.1 Brain Image

GRETNA support fMRI sequences with raw DICOM data from scanners or 3D/4D NIfTI-1 format. You need select the directory where you store all subjects, and also the **key word** as what you do in SPM8.

These fMRI data should be stored in 4 given rule:

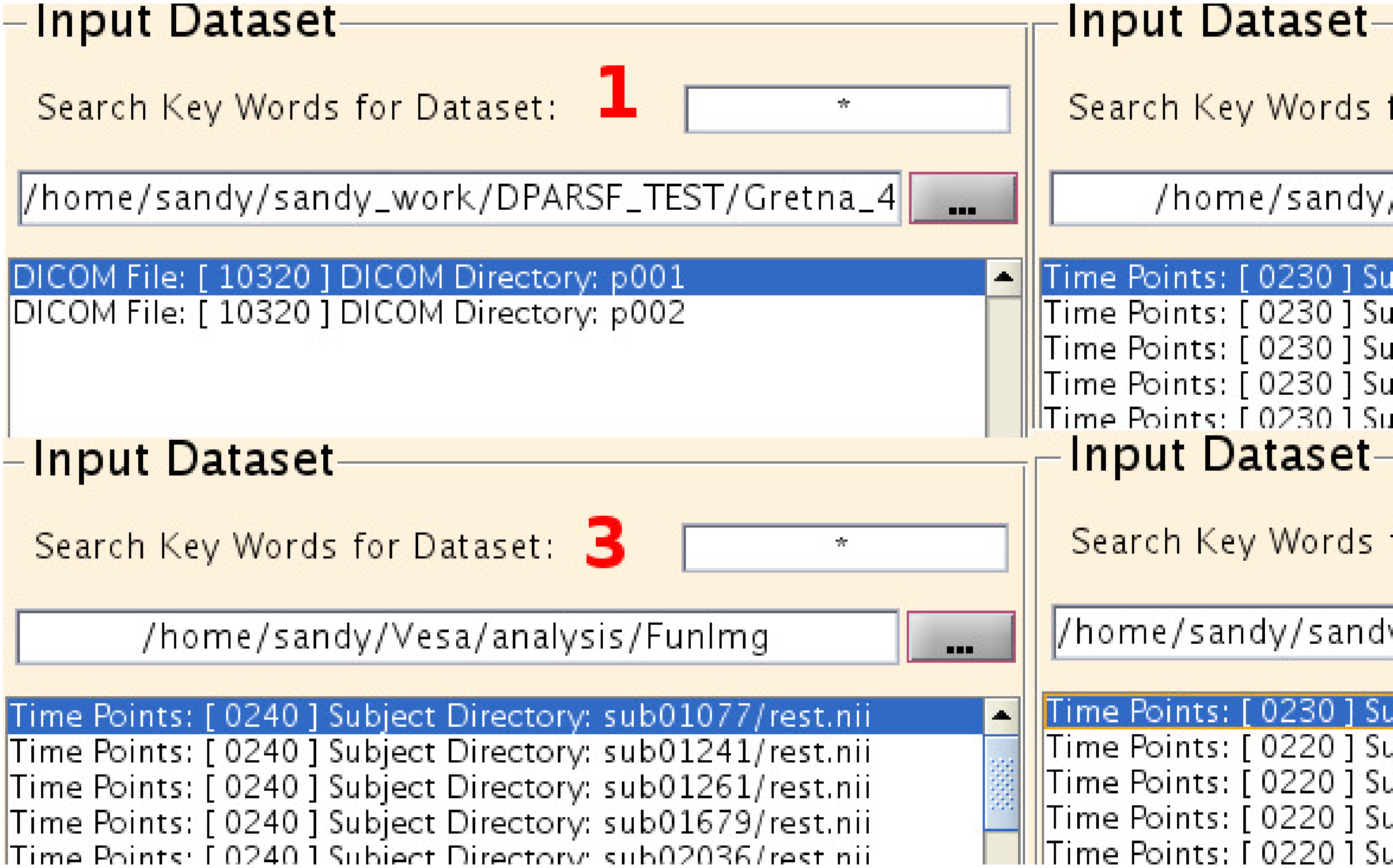
1. The sub-folders of subjects for DICOM.
2. The sub-folders of subjects for 3D NIfTI-1 files.
3. The sub-folders of subjects for 4D NIfTI-1 files.
4. 4D NIfTI-1 files with subjects' name.

7.2 Network Matrix

GRETNA support network matrix with MATLAB's MAT-file or TEXT file. For MAT-file, you can use MATLAB's norm or sparse matrix to save your network. You can check your data by double-click in GRETNA's interface.

These files should be stored in 3 given rule:

- The network matrices with different field name in MAT-file.
- The N × 1 cell of network metrices in MAT-file



Input Dataset

Search Key Words for Dataset: **1** *

/home/sandy/sandy_work/DPARSF_TEST/Gretna_4 ...

```
DICOM File: [ 10320 ] DICOM Directory: p001
DICOM File: [ 10320 ] DICOM Directory: p002
```

Input Dataset

Search Key Words f

/home/sandy/

```
Time Points: [ 0230 ] Su
Time Points: [ 0230 ] Su
Time Points: [ 0230 ] Su
Time Points: [ 0230 ] Su
Time Points: [ 0230 ] Su
```

Input Dataset

Search Key Words for Dataset: **3** *

/home/sandy/Vesa/analysis/FunImg ...

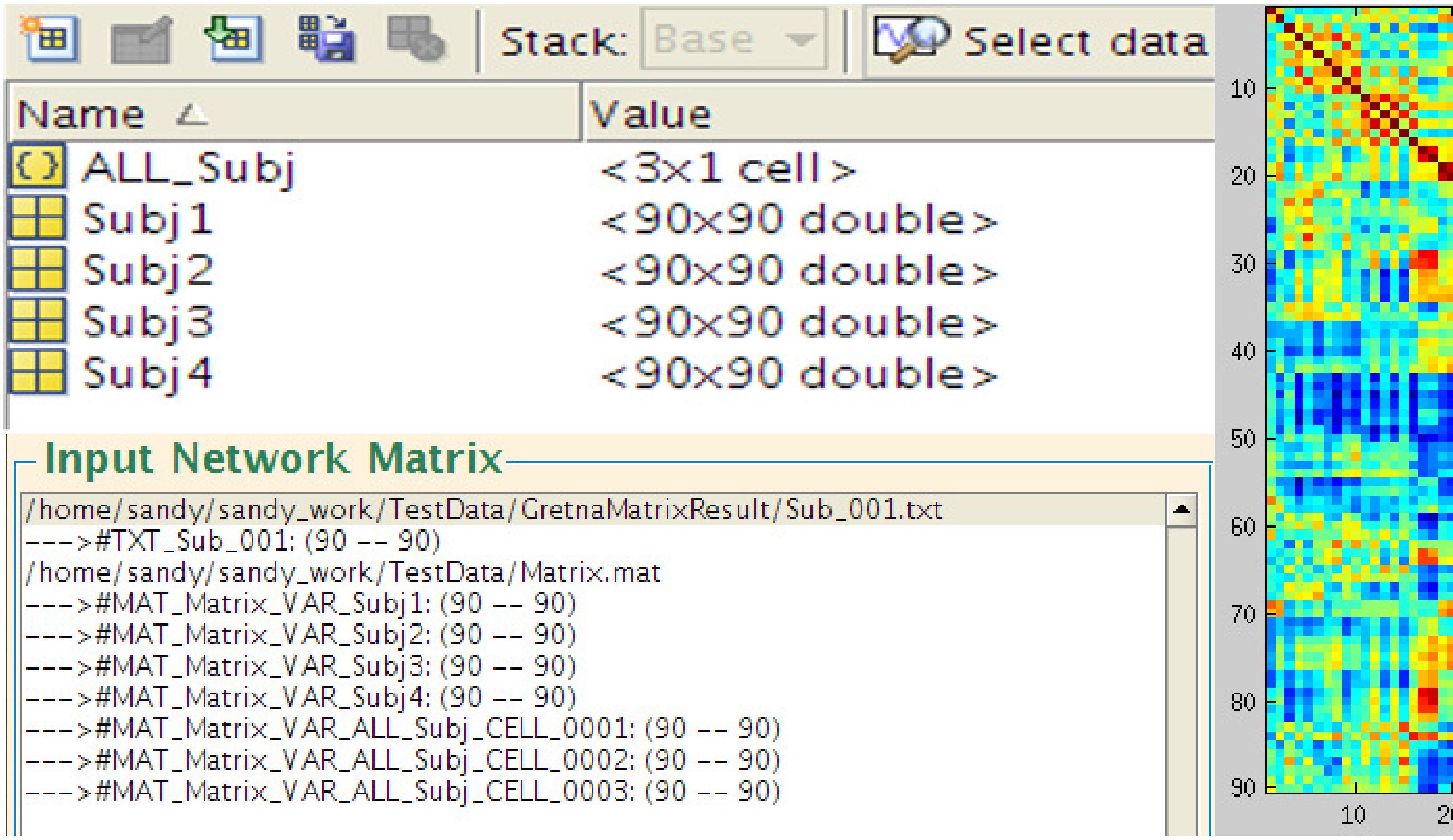
```
Time Points: [ 0240 ] Subject Directory: sub01077/rest.nii
Time Points: [ 0240 ] Subject Directory: sub01241/rest.nii
Time Points: [ 0240 ] Subject Directory: sub01261/rest.nii
Time Points: [ 0240 ] Subject Directory: sub01679/rest.nii
Time Points: [ 0240 ] Subject Directory: sub02036/rest.nii
```

Input Dataset

Search Key Words f

/home/sandy/sandy

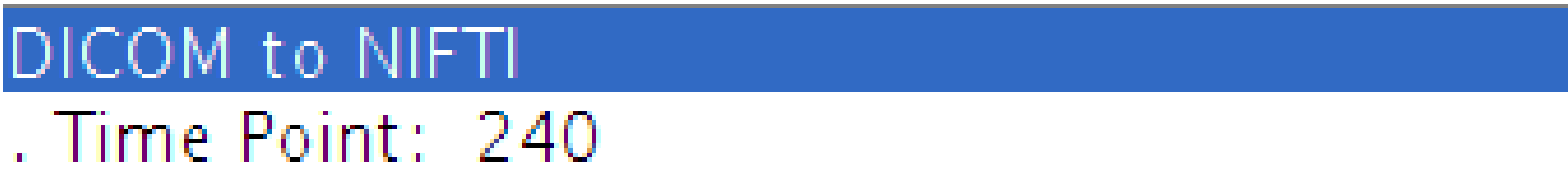
```
Time Points: [ 0230 ] Su
Time Points: [ 0220 ] Su
Time Points: [ 0220 ] Su
Time Points: [ 0220 ] Su
Time Points: [ 0220 ] Su
```



Name	Value
ALL_Subj	<3x1 cell>
Subj1	<90x90 double>
Subj2	<90x90 double>
Subj3	<90x90 double>
Subj4	<90x90 double>

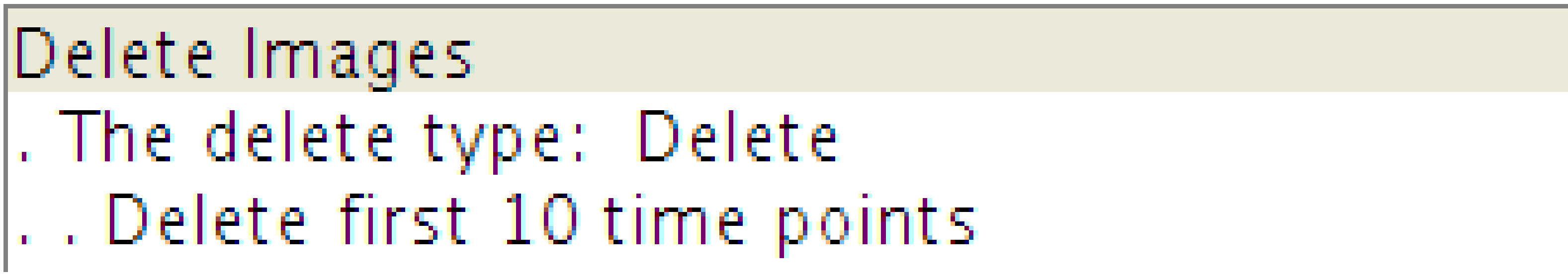
Input Network Matrix

```
/home/sandy/sandy_work/TestData/GretnaMatrixResult/Sub_001.txt
--->#TXT_Sub_001: (90 -- 90)
/home/sandy/sandy_work/TestData/Matrix.mat
--->#MAT_Matrix_VAR_Subj1: (90 -- 90)
--->#MAT_Matrix_VAR_Subj2: (90 -- 90)
--->#MAT_Matrix_VAR_Subj3: (90 -- 90)
--->#MAT_Matrix_VAR_Subj4: (90 -- 90)
--->#MAT_Matrix_VAR_ALL_Subj_CELL_0001: (90 -- 90)
--->#MAT_Matrix_VAR_ALL_Subj_CELL_0002: (90 -- 90)
--->#MAT_Matrix_VAR_ALL_Subj_CELL_0003: (90 -- 90)
```



DICOM to NIFTI

. Time Point: 240



Delete Images

. The delete type: Delete

. . Delete first 10 time points

- The network matrices in TEXT file.

8 Network Construction

For **Slice Timing, Realign, Normalization, Smooth**, you can read SPM manual to know more details.

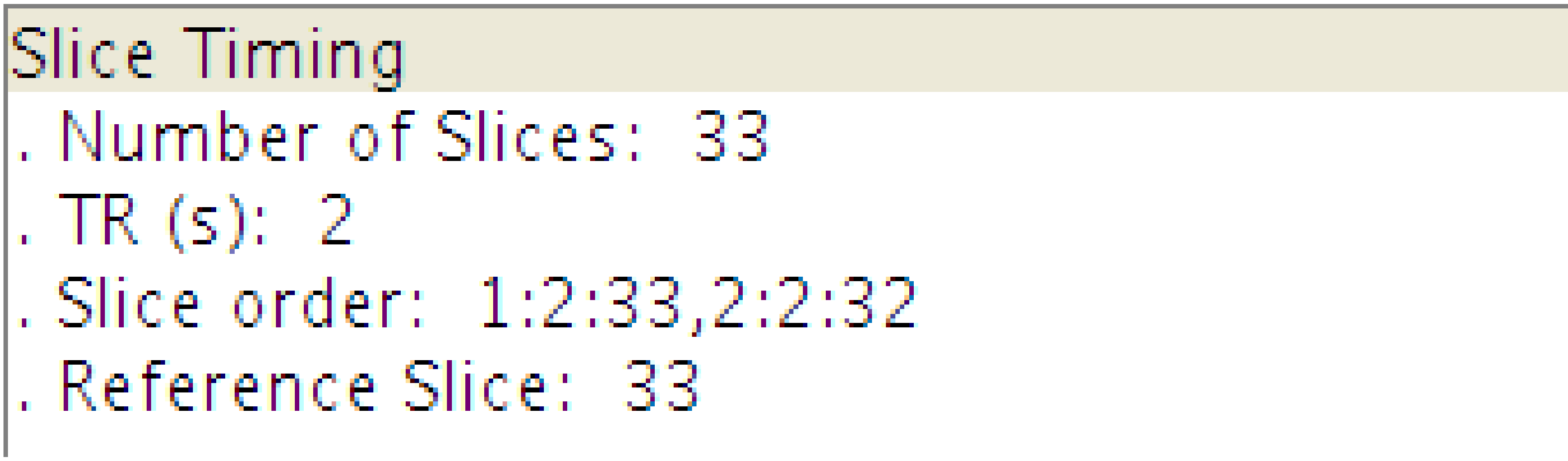
8.1 DICOM to NIfTI

- **Time Point**: The number of time points for your data.

8.2 Delete Images

The first n volumes can be discarded for the signal equilibrium and participants' adaptation to the scanning noise.

- **The delete type**: Select "Delete" or "Retain" by double-click to choose the type for delete images.



Slice Timing

. Number of Slices: 33

. TR (s): 2

. Slice order: 1:2:33,2:2:32

. Reference Slice: 33

- Select the **number of image sequences** to delete or retain by double-click.

8.3 Slice Timing

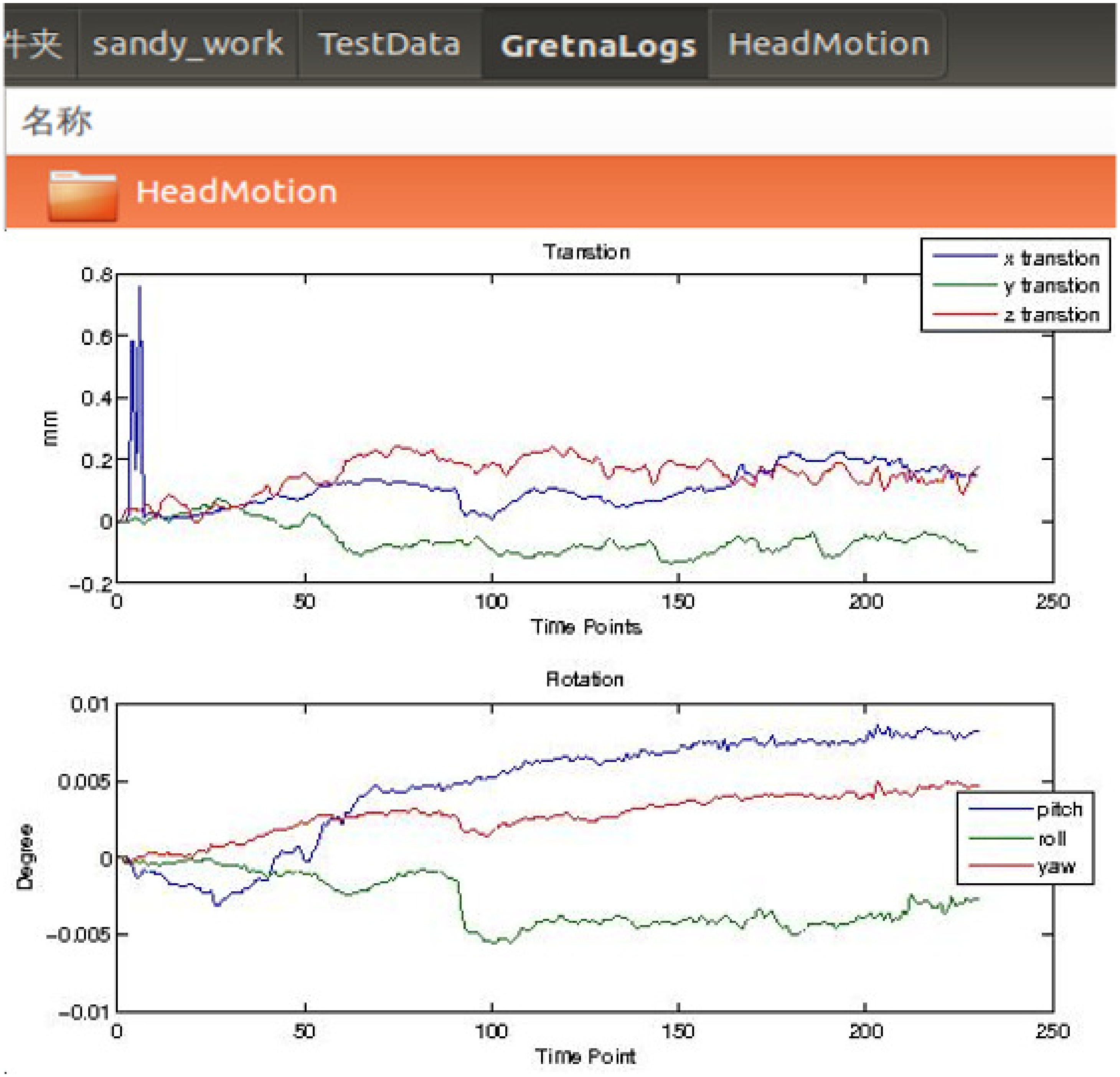
- **Number of Slices**: The number of slices in one volumn.
- **TR (s)**: The time of repeat of fMRI signal.
- **Slice Order**: The sequence of Slice, e.g. interleaved (bottom − > up) [1:2:nslices 2:2:nslices].
- **Reference Slice**: The slice number as reference.

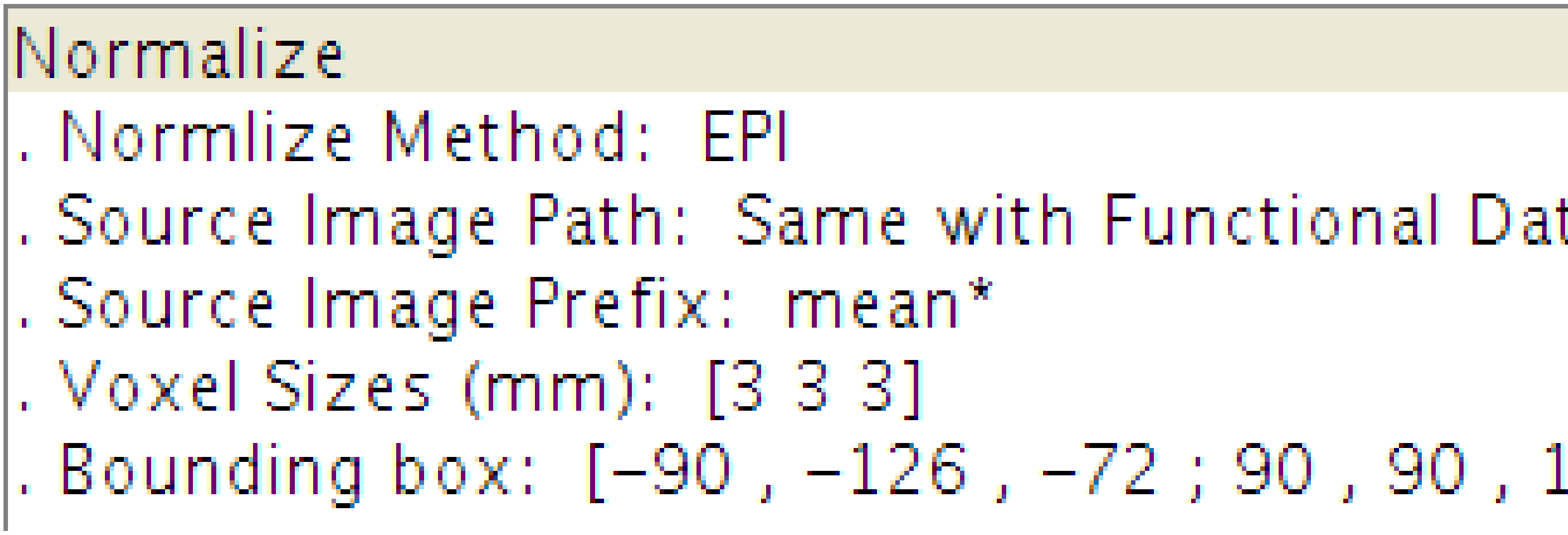
8.4 Realign

You can check subjects' head motion parameter in the "GretnaLogs/HeadMotion" folder.

8.5 Normalize EPI

- **Source Image Path**: The subjects' parent directory of image that is warped to match the templates.


件夹
sandy_work
TestData
GretnaLogs
HeadMotion
名称
HeadMotion
Transtion
x transtion
y transtion
z transtion
mm
0.8
0.6
0.4
0.2
0
−0.2
0
50
100
150
200
250
Time Points
Rotation
pitch
roll
yaw
Degree
0.01
0.005
0
−0.005
−0.01
0
50
100
150
200
250
Time Point


Normalize

. Normlize Method: EPI

. Source Image Path: Same with Functional Dat

. Source Image Prefix: mean*

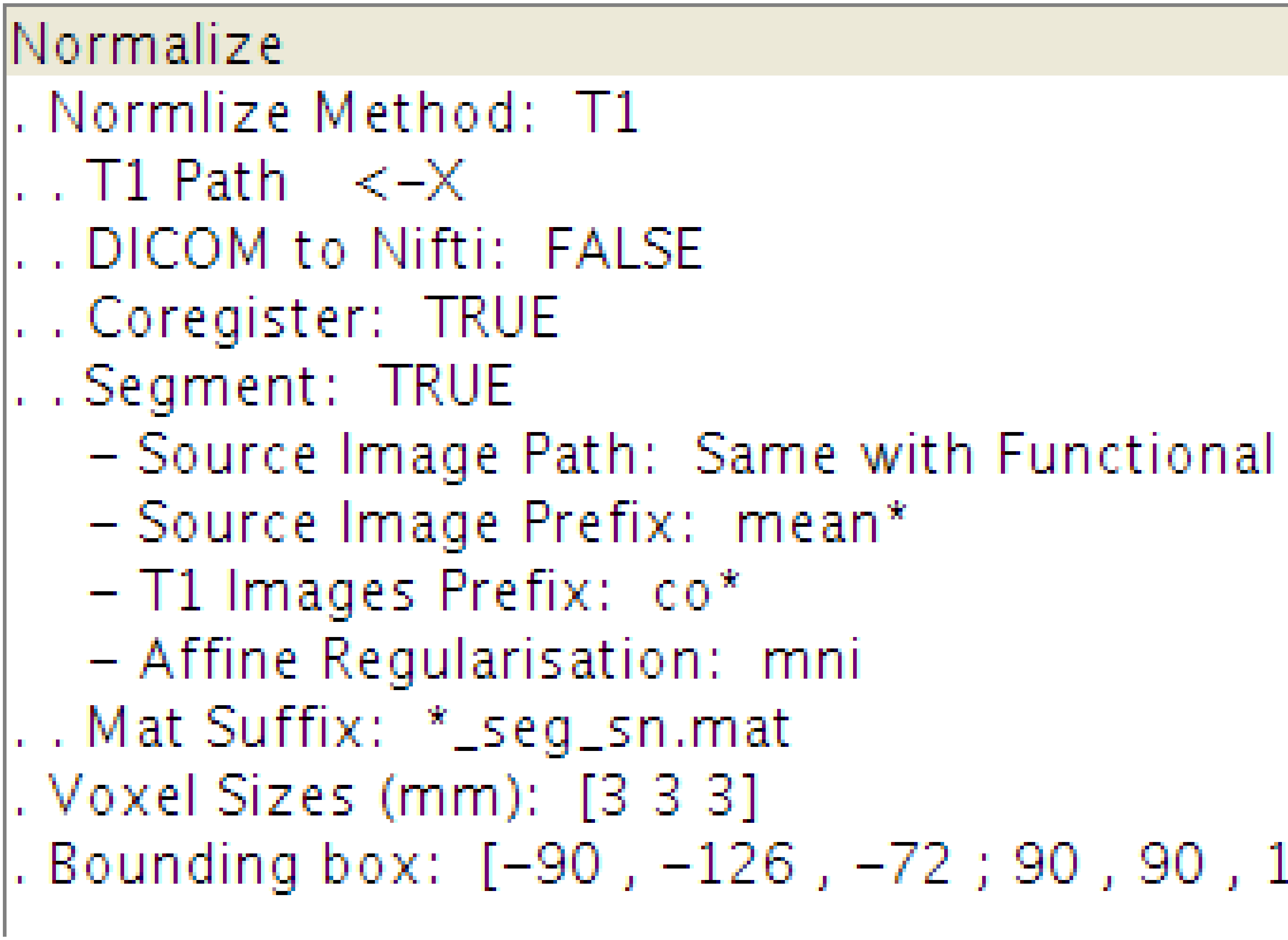
. Voxel Sizes (mm): [3 3 3]

. Bounding box: [-90 , -126 , -72 ; 90 , 90 , 1

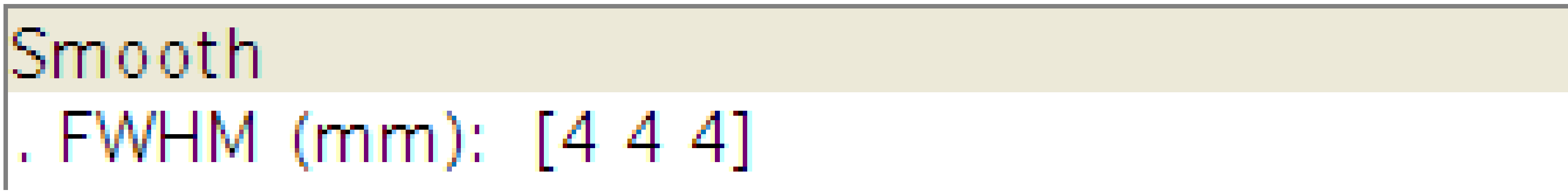
- **Source Image Prefix**: The prefix of image that is warped to match the templates.
- **Voxel Sizes (mm)**: The voxel size of the written normalised images.
- **Bounding Box**: The bounding box (in mm) of the volume which is to be written.

8.6 Normalize T1

- **T1 Path**: The subjects' parent directory of T1 image.
- **DICOM to NIfTI**: Execute DICOM to NIfTI or not.
- **Coregister**: Execute coregister T1 image to image that is warped to match the templates or not.
- **Segment**: Execute segment T1 image or not.
  - **Source Image Path**: The subjects' parent directory of image that is warped to match the templates.

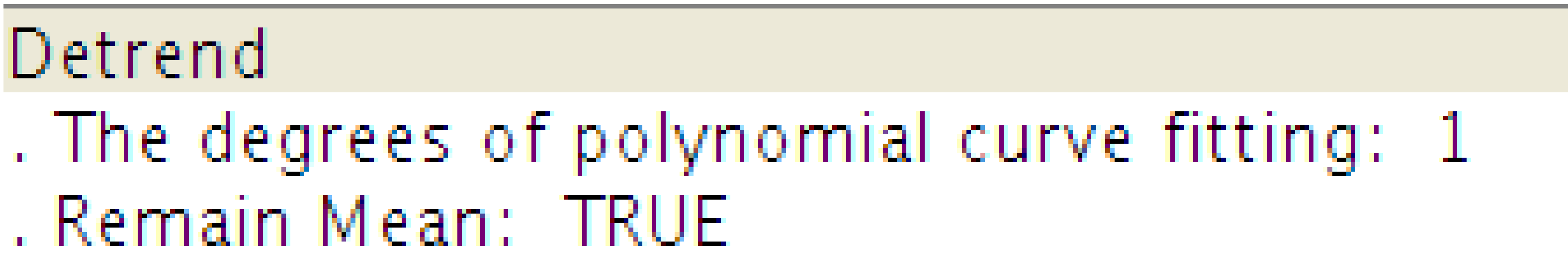


```
Normalize
. Normlize Method:  T1
. . T1 Path   <-X
. . DICOM to Nifti:  FALSE
. . Coregister:  TRUE
. . Segment:  TRUE
    – Source Image Path:  Same with Functional
    – Source Image Prefix:  mean*
    – T1 Images Prefix:  co*
    – Affine Regularisation:  mni
. . Mat Suffix:  *_seg_sn.mat
. Voxel Sizes (mm):  [3 3 3]
. Bounding box:  [-90 , -126 , -72 ; 90 , 90 , 1
```



Smooth

. FWHM (mm): [4 4 4]



Detrend

. The degrees of polynomial curve fitting: 1

. Remain Mean: TRUE

  - **Source Image Prefix**: The prefix of image that is warped to match the templates.
  - **T1 Image Prefix**: The prefix of T1 image.
  - **Affine Regularisation**: Select ”mni” or ”estern”.

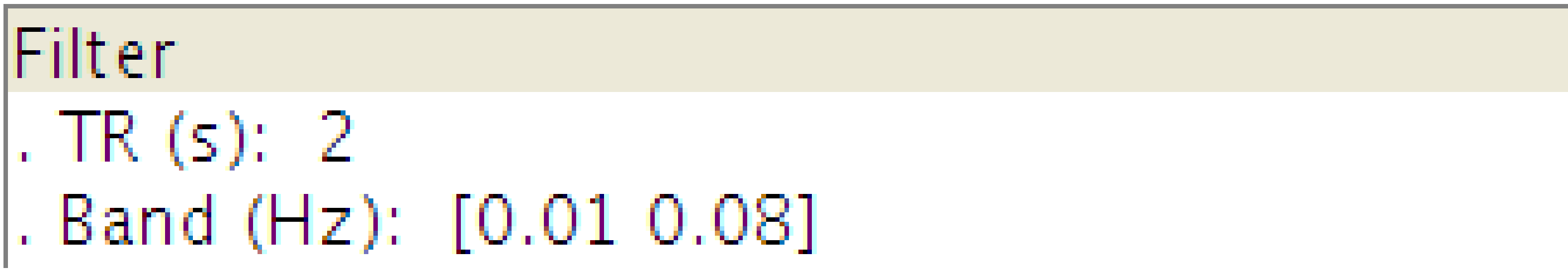
- **Mat Suffix**: The suffix of transation matrix.

- **Voxel Sizes (mm)**: The voxel size of the written normalised images.

- **Bounding Box**: The bounding box (in mm) of the volume which is to be written.

8.7 Smooth

- **FWHM (mm)**: The full width half maximum of kernel.



Filter

. TR (s): 2

. Band (Hz): [0.01 0.08]

8.8 Detrend

- **The Degree of Polynomial Curve fitting**: The degree of trend.

- **Remain Mean**: Remain the mean of time courses or not.

8.9 Filter

- **TR (s)**: The time of repeat of fMRI signal.

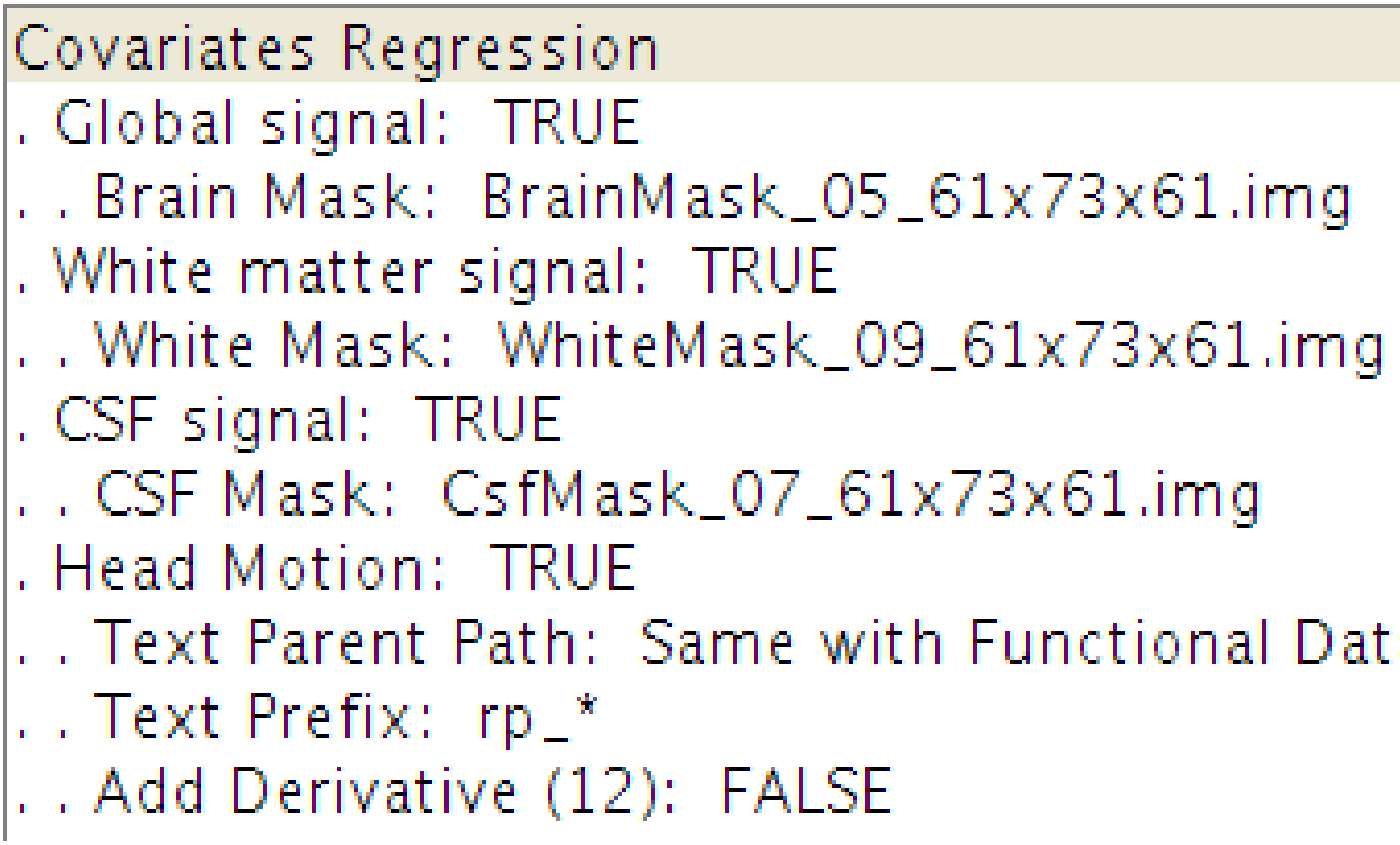
- **Band (Hz)**: The frequency band for temporal filtering.

8.10 Covariates Regression

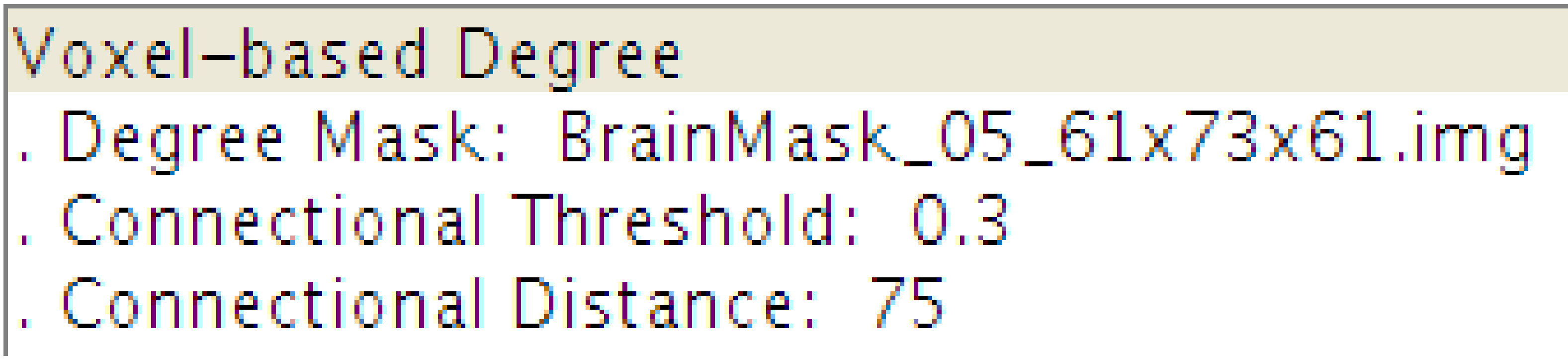
- **Global Signal**: Regress out global signal or not.
  - **Brain Mask**: The mask of whole brain.

- **White Matter Signal**: Regress out white matter signal or not.
  - **White Mask**: The mask of white matter.

- **CSF Signal**: Regress out cerebrospinal fluid signal or not.



```
Covariates Regression
. Global signal:  TRUE
. . Brain Mask:  BrainMask_05_61x73x61.img
. White matter signal:  TRUE
. . White Mask:  WhiteMask_09_61x73x61.img
. CSF signal:  TRUE
. . CSF Mask:  CsfMask_07_61x73x61.img
. Head Motion:  TRUE
. . Text Parent Path:  Same with Functional Dat
. . Text Prefix:  rp_*
. . Add Derivative (12):  FALSE
```



Voxel-based Degree

. Degree Mask: BrainMask_05_61x73x61.img

. Connectional Threshold: 0.3

. Connectional Distance: 75

  - **CSF Mask**: The mask of cerebrospinal fluid.

- **Head Motion**: Regress out head motion parameters or not.

  - **Text Parent Path**: The subjects' parent directory of head motion parameters' text file.

  - **Text Prefix**: The prefix of head motion parameters' text file.

  - **Add Derivative (12)**: Regress out derivative head motion or not.

8.11 Voxel-based Degree

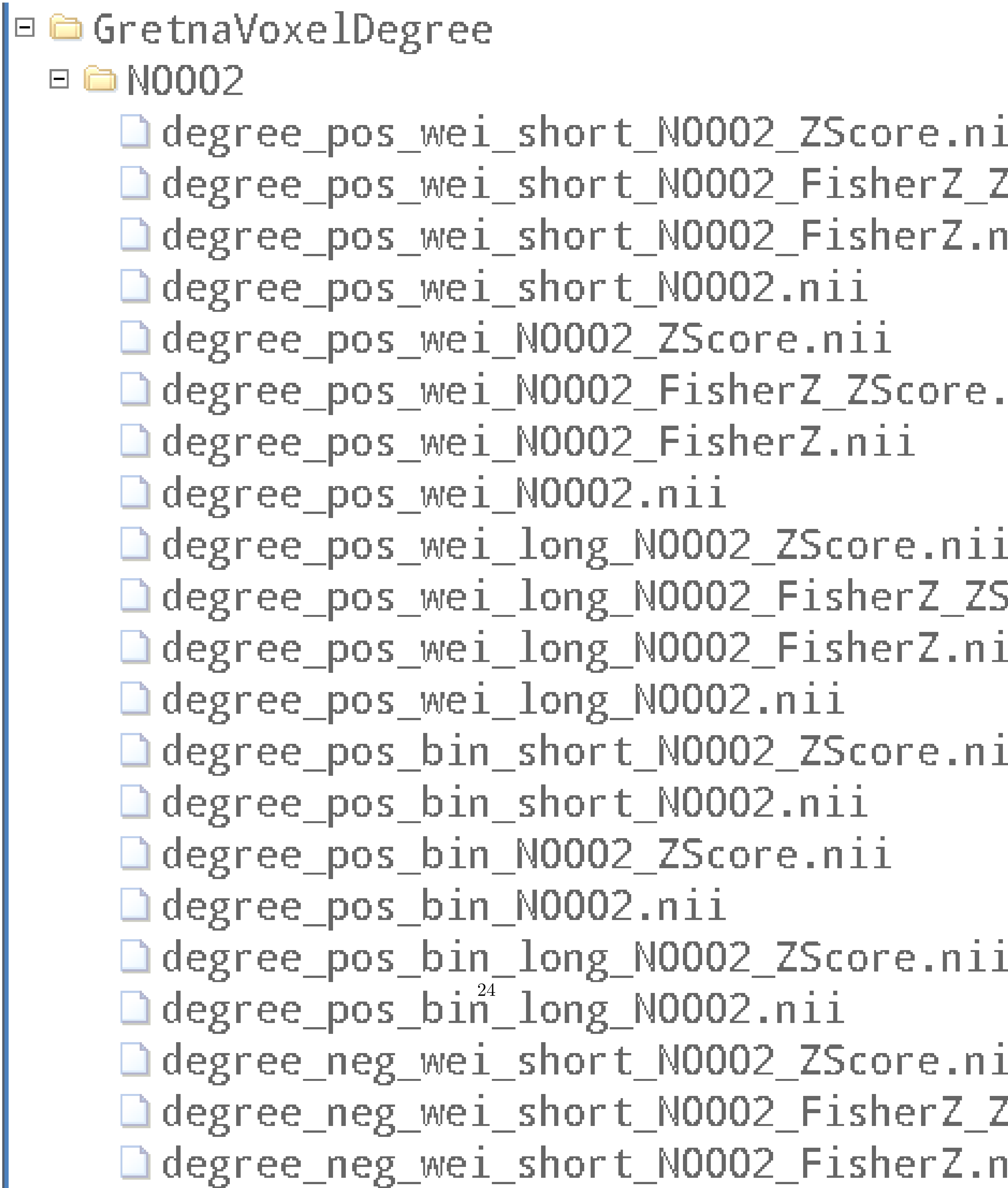
- **Degree Mask**: The mask that include all network nodes you want.

- **Connectional Threshold**: The threshold for correlation coefficient.

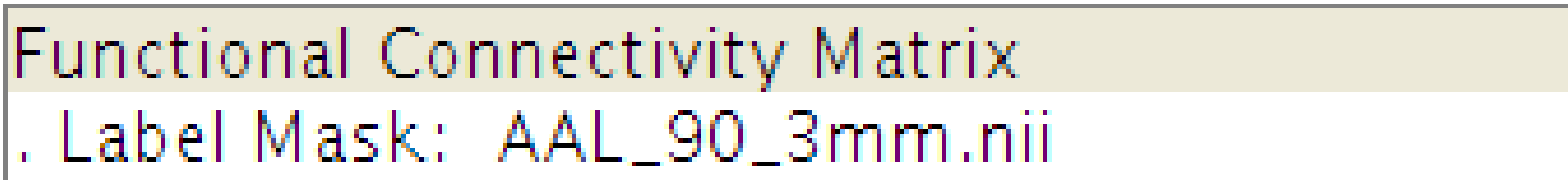
- **Connectional Distance**: The euclidean distance between nodes.

All results of voxel-based degree will be stored in the subjects' sub-folder of "GretnaVoxelDegree" folder with the following specific label:

- *wei*, *bin*: using weighted or binary connections to estimate voxel-based degree.



```
⊟ GretnaVoxelDegree
  ⊟ N0002
      degree_pos_wei_short_N0002_ZScore.ni
      degree_pos_wei_short_N0002_FisherZ_Z
      degree_pos_wei_short_N0002_FisherZ.n
      degree_pos_wei_short_N0002.nii
      degree_pos_wei_N0002_ZScore.nii
      degree_pos_wei_N0002_FisherZ_ZScore.
      degree_pos_wei_N0002_FisherZ.nii
      degree_pos_wei_N0002.nii
      degree_pos_wei_long_N0002_ZScore.nii
      degree_pos_wei_long_N0002_FisherZ_ZS
      degree_pos_wei_long_N0002_FisherZ.ni
      degree_pos_wei_long_N0002.nii
      degree_pos_bin_short_N0002_ZScore.ni
      degree_pos_bin_short_N0002.nii
      degree_pos_bin_N0002_ZScore.nii
      degree_pos_bin_N0002.nii
      degree_pos_bin_long_N0002_ZScore.nii
      degree_pos_bin_long_N0002.nii
      degree_neg_wei_short_N0002_ZScore.ni
      degree_neg_wei_short_N0002_FisherZ_Z
      degree_neg_wei_short_N0002_FisherZ.n
```



Functional Connectivity Matrix

. Label Mask: AAL_90_3mm.nii

- *pos*, *neg* or *abs*: using positive, negative or absolute connections to estimate voxel-based degree.
- *short* or *long*: using short or long connections which were defined in GUI to estimate voxel-based degree.
- *Fisher*: executing Fisher's Z transformation instead of R to estimate voxel-based degree.
- *ZScore*: standardizing results with z-score.

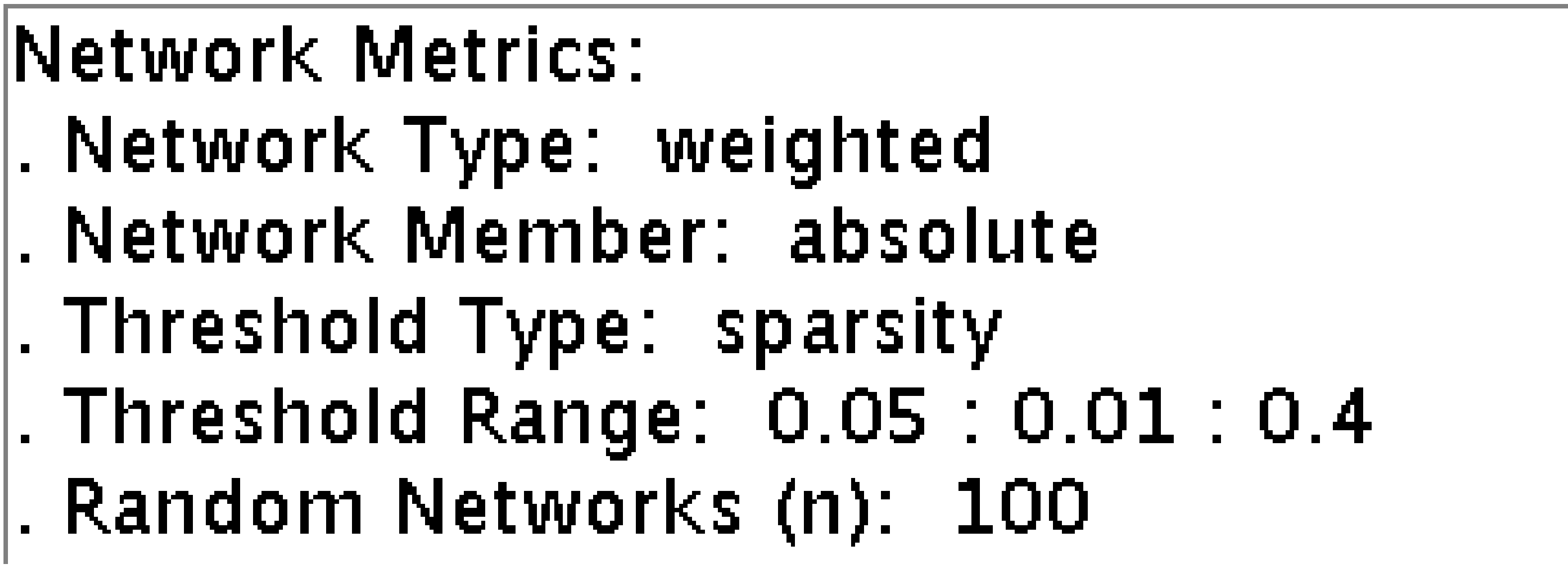
8.12 Functional Connectivity Matrix

- **Label Mask**: The template of brain which have different number for different region, e.g. AAL90.

9 Network Analysis

9.1 Network Configurations

- **Network Type**: Select ”weighted” or ”binary” network.
- **Network Member**: Remove the negative value in matrix (Positive), remove the positive value in matrix (Negative) or use the absolute value (Absolute).
- **Threshold Type**: Select the method to cut the network matrices, ”sparsity” or ”similarity threshold”. If DTI's network matrix, select ”similarity threshold” and input the threshold of fiber number. If BOLD-fMRI's network



```
Network Metrics:
. Network Type:  weighted
. Network Member:  absolute
. Threshold Type:  sparsity
. Threshold Range:  0.05 : 0.01 : 0.4
. Random Networks (n):  100
```

matrix, select ”similarity threshold” for ”correlation coefficient” or ”r value”.

- **Threshold Range**: Select the range of threshold, it could be one value or a sequence.

- **Random Network**: The number of random network.

9.2 Network - Small World

9.3 Network - Efficiency

9.4 Network - Rich Club

9.5 Network - Modularity

9.6 Network - Assortativity

9.7 Network - Hierarchy

9.8 Network - Synchronization

9.9 Node - Degree

9.10 Node - Efficiency

9.11 Node - Betweenness

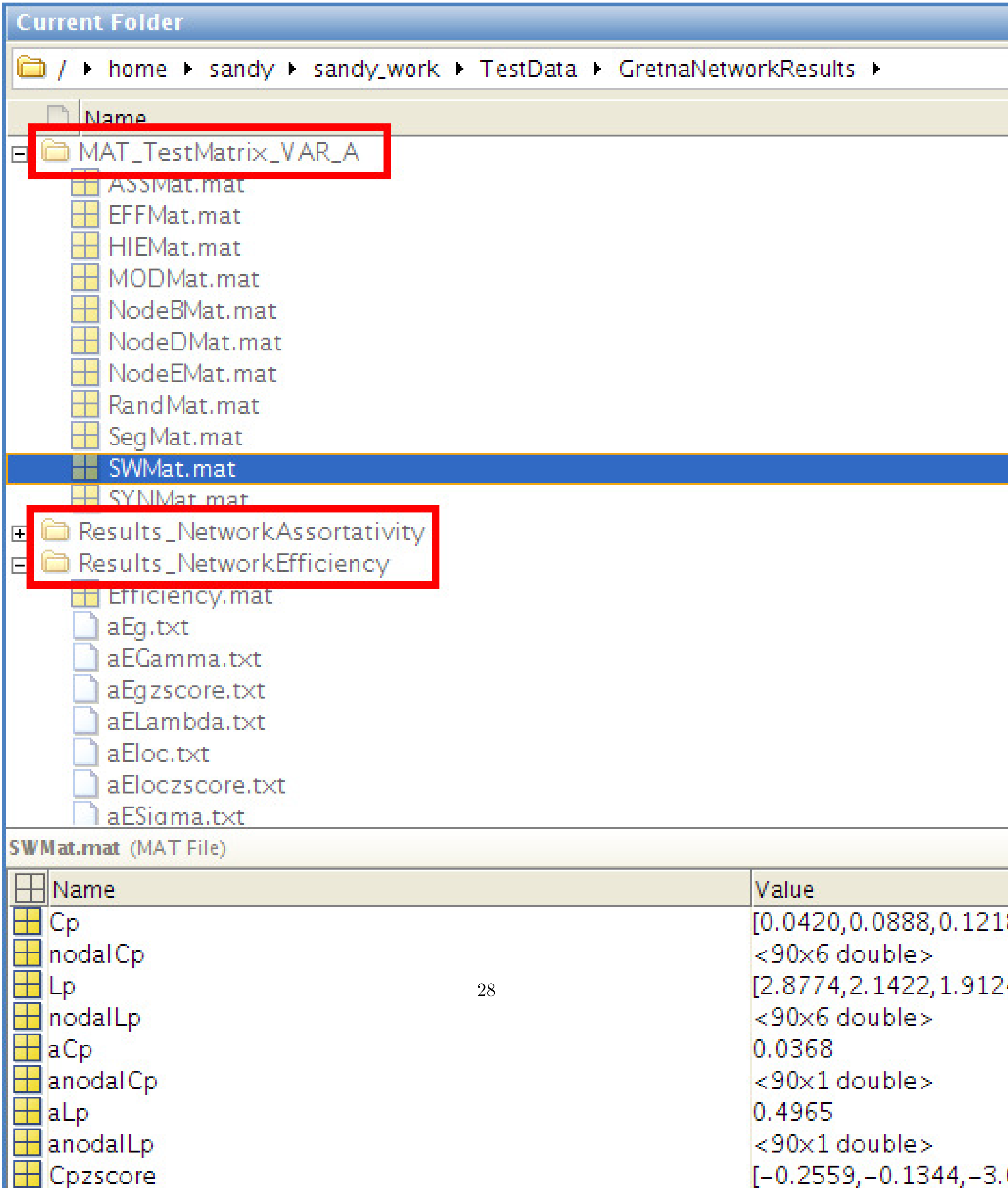
10 Network Results

All results of network metric will be stored with MAT-file and TEXT-file both. You can pick these metrics from subjects' diretory (e.g. "MAT_TestMatrix_VAR_A") one by one or use the integrated results (Results_*).

10.1 Individual Results

10.1.1 Network - Small World

You can load "SWMat.mat" to get the following metrics.



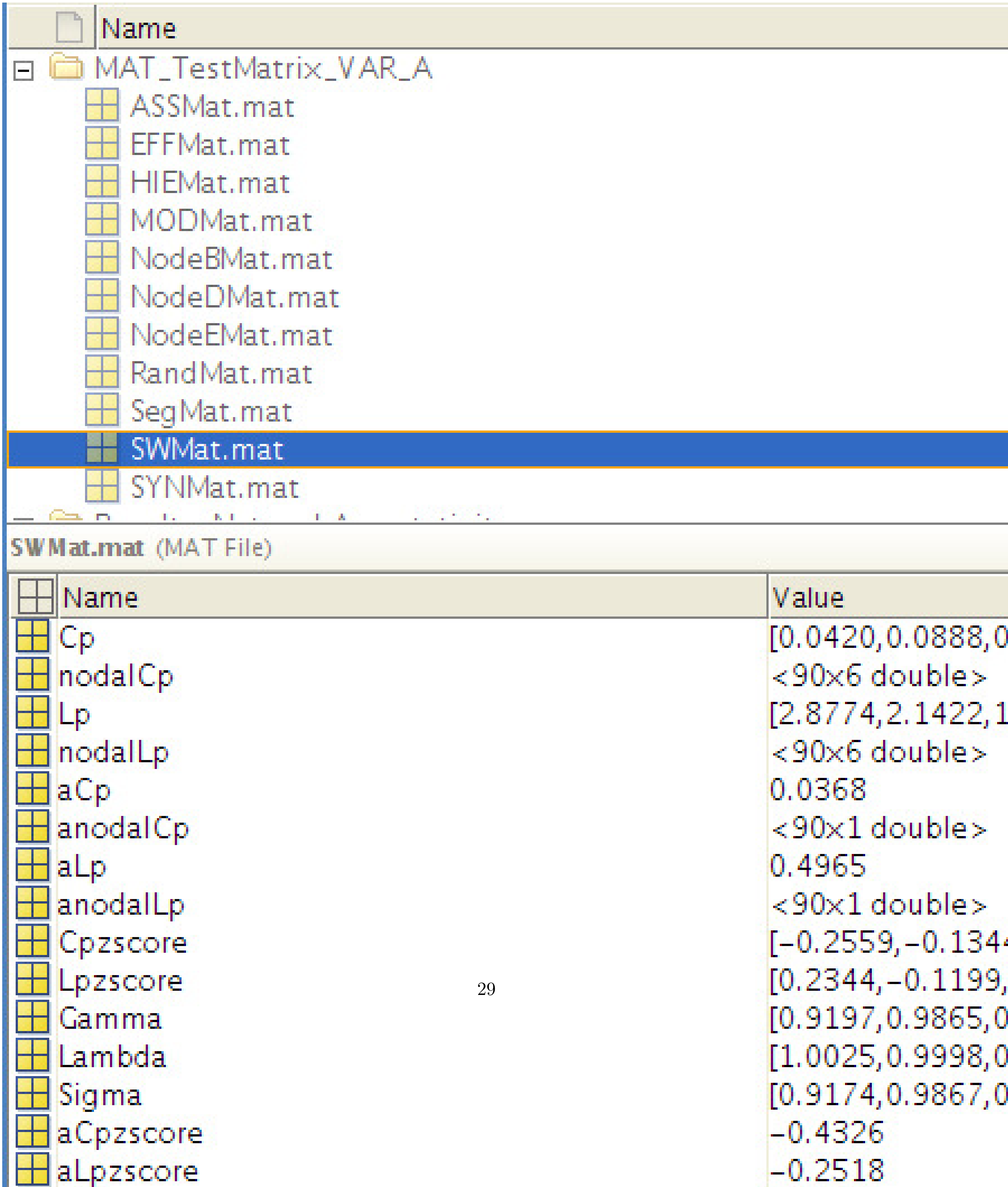
Current Folder

/ ▸ home ▸ sandy ▸ sandy_work ▸ TestData ▸ GretnaNetworkResults ▸

```
Name
⊟ MAT_TestMatrix_VAR_A
     ASSMat.mat
     EFFMat.mat
     HIEMat.mat
     MODMat.mat
     NodeBMat.mat
     NodeDMat.mat
     NodeEMat.mat
     RandMat.mat
     SegMat.mat
     SWMat.mat
     SYNMat.mat
⊞ Results_NetworkAssortativity
⊟ Results_NetworkEfficiency
     Efficiency.mat
     aEg.txt
     aEGamma.txt
     aEgzscore.txt
     aELambda.txt
     aEloc.txt
     aEloczscore.txt
     aESigma.txt
```

SWMat.mat (MAT File)

Name	Value
Cp	[0.0420,0.0888,0.121
nodalCp	<90x6 double>
Lp	[2.8774,2.1422,1.912
nodalLp	<90x6 double>
aCp	0.0368
anodalCp	<90x1 double>
aLp	0.4965
anodalLp	<90x1 double>
Cpzscore	[-0.2559,-0.1344,-3.



Name
MAT_TestMatrix_VAR_A
ASSMat.mat
EFFMat.mat
HIEMat.mat
MODMat.mat
NodeBMat.mat
NodeDMat.mat
NodeEMat.mat
RandMat.mat
SegMat.mat
SWMat.mat
SYNMat.mat

SWMat.mat (MAT File)

Name	Value
Cp	[0.0420,0.0888,0
nodalCp	<90x6 double>
Lp	[2.8774,2.1422,1
nodalLp	<90x6 double>
aCp	0.0368
anodalCp	<90x1 double>
aLp	0.4965
anodalLp	<90x1 double>
Cpzscore	[-0.2559,-0.134
Lpzscore	[0.2344,-0.1199,
Gamma	[0.9197,0.9865,0
Lambda	[1.0025,0.9998,0
Sigma	[0.9174,0.9867,0
aCpzscore	-0.4326
aLpzscore	-0.2518

- **Cp**: Clustering coefficient of network. 1×N array, N is the number of threshold sequences.

- **Lp**: Shortest path length of network. 1×N array, N is the number of threshold sequences.

- **nodalCp**: Clustering coefficient of node. M×N array, M is the number of nodes, N is the number of threshold sequences.

- **nodalLp**: Shortest path length of node. M×N array, M is the number of nodes, N is the number of threshold sequences.

- **Cpzscore**: The z-score of clustering coefficient of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Cpzscore = \frac{Cp - mean(Cprand)}{std(Cprand)}$$

  Cprand is a R×1 array, R is the number of randomized network. It is the clustering coefficient of randomized network.

- **Lpzscore**: The z-score of shortest path length of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Lpzscore = \frac{Lp - mean(Lprand)}{std(Lprand)}$$

  Lprand is a R×1 array, R is the number of randomized network. It is the shortest path length of randomized network.

- **Gamma**: Gamma is the ratio of Cp and mean value of Cprand, 1×N array, N is the number of threshold sequences.

The formula is following:

$$Gamma = \frac{Cp}{mean(Cprand)}$$

- **Lambda**: Lambda is the ratio of Lp and mean value of Lprand, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Lambda = \frac{Lp}{mean(Lprand)}$$

- **Sigma**: Sigma is the ratio of Gamma and Lambda. 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Sigma = \frac{Gamma}{Lambda}$$

- **aCp**: The AUC (area under curve) of Cp.

- **aLp**: The AUC of Lp.

- **anodalCp**: The AUC of nodalCp. M×1 array, M is the number of nodes.

- **anodalLp**: The AUC of nodalLp. M×1 array, M is the number of nodes.

- **aCpzscore**: The AUC of Cpzscore.

- **aLpzscore**: The AUC of Lpzscore.

- **aGamma**: The AUC of Gamma.

- **aLambda**: The AUC of Lambda.

- **aSigma**: The AUC of Sigma.

10.1.2 Network - Efficiency

You can load "EFFMat.mat" to get the following metrics.

- **Eloc**: Local efficiency of network. 1×N array, N is the number of threshold sequences.

- **Eg**: Global efficiency of network. 1×N array, N is the number of threshold sequences.

- **nodalEloc**: Local efficiency of node. M×N array, M is the number of nodes, N is the number of threshold sequences.

- **nodalEg**: Global efficiency of node. M×N array, M is the number of nodes, N is the number of threshold sequences.

- **Eloczscore**: The z-score of local efficiency of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Eloczscore = \frac{Eloc - mean(Elocrand)}{std(Elocrand)}$$

  Cprand is a R×1 array, R is the number of randomized network. It is the local efficiency of randomized network.

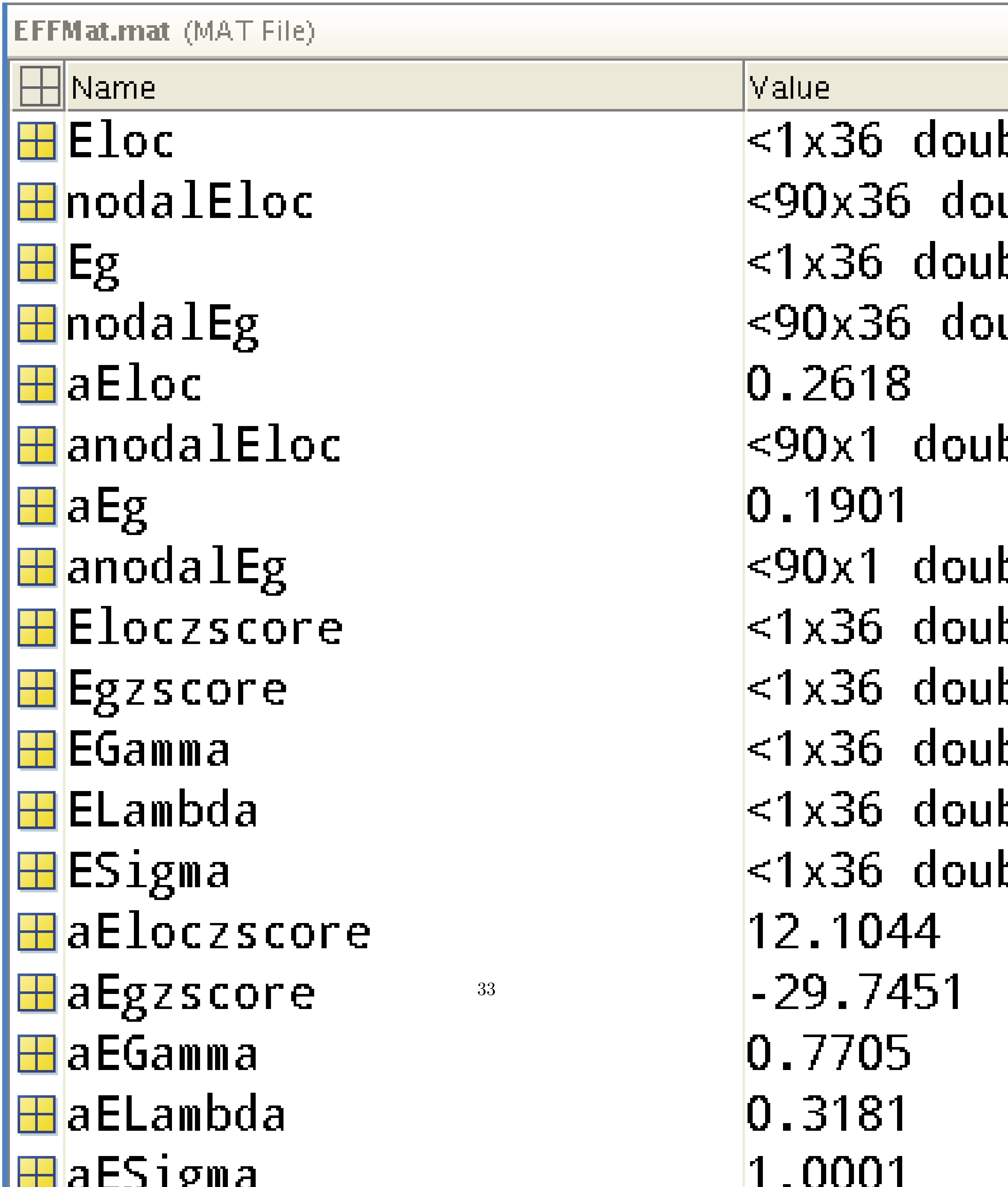
- **Egzscore**: The z-score of global efficiency of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$Egzscore = \frac{Eg - mean(Egrand)}{std(Egrand)}$$

  Egrand is a R×1 array, R is the number of randomized network. It is the global efficiency of randomized network.

- **EGamma**: EGamma is the ratio of Eloc and mean value of Elocrand, 1×N array, N is the number of threshold sequences.



EFFMat.mat (MAT File)

Name	Value
Eloc	<1x36 doub
nodalEloc	<90x36 dou
Eg	<1x36 doub
nodalEg	<90x36 dou
aEloc	0.2618
anodalEloc	<90x1 doub
aEg	0.1901
anodalEg	<90x1 doub
Eloczscore	<1x36 doub
Egzscore	<1x36 doub
EGamma	<1x36 doub
ELambda	<1x36 doub
ESigma	<1x36 doub
aEloczscore	12.1044
aEgzscore	-29.7451
aEGamma	0.7705
aELambda	0.3181
aESigma	1.0001

The formula is following:

$$EGamma = \frac{Eloc}{mean(Elocrand)}$$

- **ELambda**: ELambda is the ratio of Eg and mean value of Egrand, 1×N array, N is the number of threshold sequences.

  The formula is following:

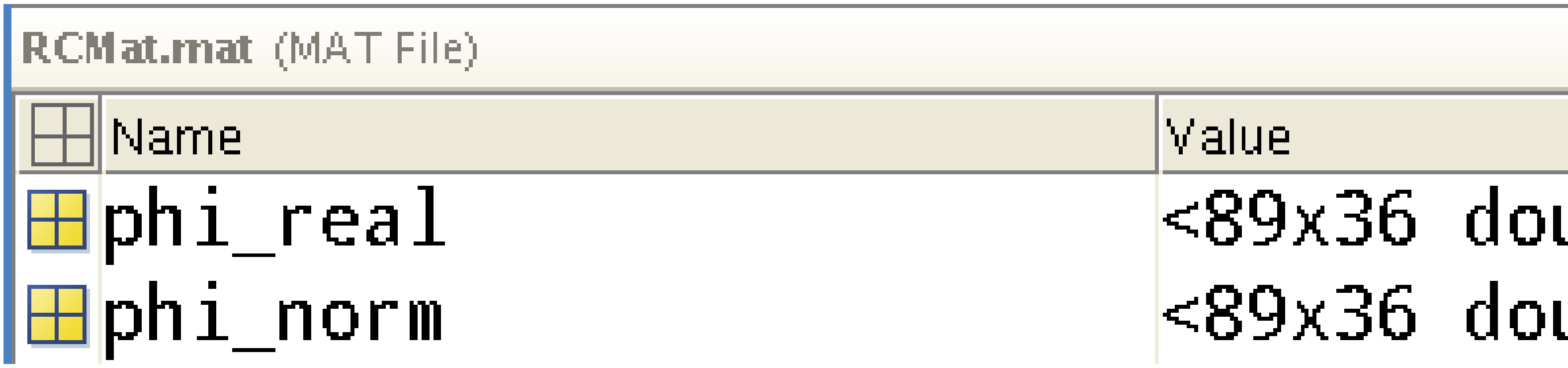
$$ELambda = \frac{Eg}{mean(Egrand)}$$

- **ESigma**: ESigma is the ratio of EGamma and ELambda. 1×N array, N is the number of threshold sequences.

  The formula is following:

$$ESigma = \frac{EGamma}{ELambda}$$

- **aEloc**: The AUC (area under curve) of Eloc.
- **aEg**: The AUC of Eg.
- **anodalEloc**: The AUC of nodalEloc. M×1 array, M is the number of nodes.
- **anodalEg**: The AUC of nodalEg. M×1 array, M is the number of nodes.
- **aEloczscore**: The AUC of Eloczscore.
- **aEgzscore**: The AUC of Egzscore.
- **aEGamma**: The AUC of EGamma.
- **aELambda**: The AUC of ELambda.



RCMat.mat (MAT File)

Name	Value
phi_real	<89x36 dou
phi_norm	<89x36 dou

- **aESigma**: The AUC of ESigma.

10.1.3 Network - Rich Club

You can load ”RCMat.mat” to get the following metrics.

- **phi_real**: The rich club coefficient of real network. K×N array, K is the number of binary node degree, from 1 to Node-1, N is the number of threshold sequences.

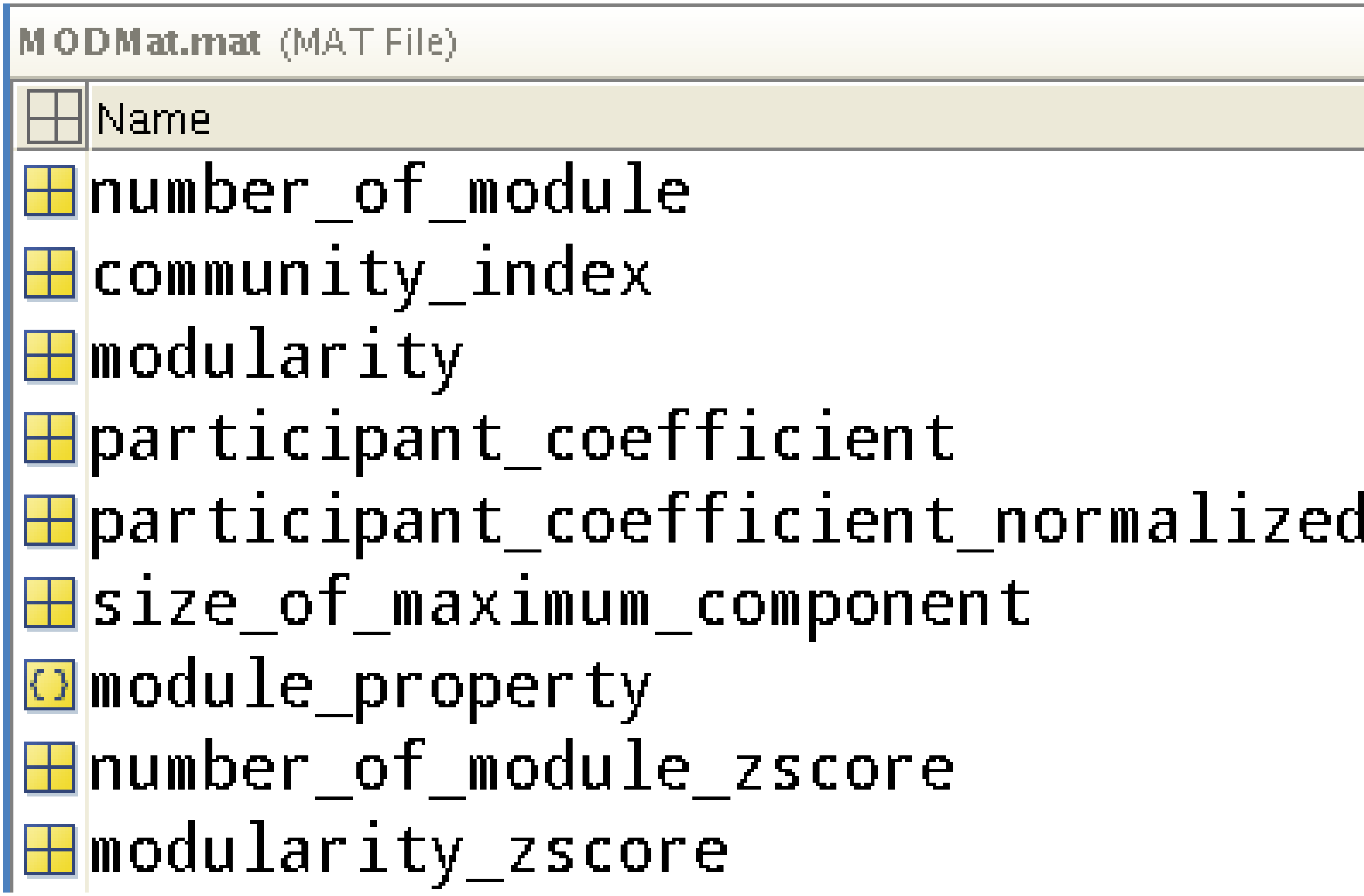
- **phi_norm**: The normalized rich club coefficient of real network. K×N array, K is the number of binary node degree, from 1 to Node-1, N is the number of threshold sequences.

10.1.4 Network - Modularity

You can load ”MODMat.mat” to get the following metrics.

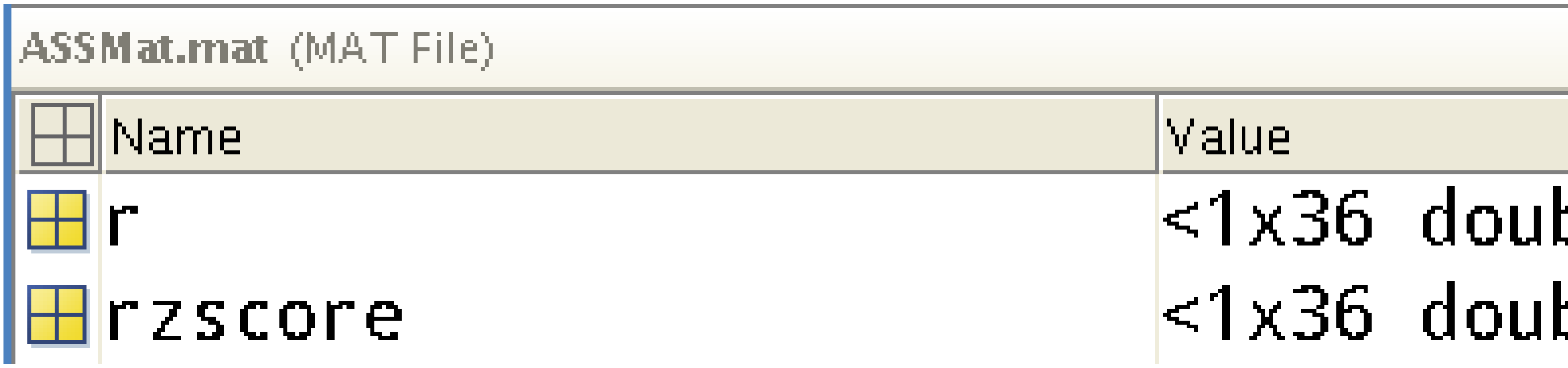
- **community_index**: The community (listed for each node), M×N array, M is the number of nodes, N is the number of threshold sequences.

- **number_of_module**: The number of module in network, 1×N array, N is the number of threshold sequences.



MODMat.mat (MAT File)

Name
number_of_module
community_index
modularity
participant_coefficient
participant_coefficient_normalized
size_of_maximum_component
module_property
number_of_module_zscore
modularity_zscore



ASSMat.mat (MAT File)

Name	Value
r	<1x36 doub
rzscore	<1x36 doub

- **modularity**: Modularity value of network, 1×N array, N is the number of threshold sequences.

- **modularity_zscore**: The z-score of modularity of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$modularity\_zscore = \frac{modularity - mean(modrand)}{std(modrand)}$$

  modrand is a R×1 array, R is the number of randomized network. It is the modularity of randomized network.

10.1.5 Network - Assortativity

You can load ”ASSMat.mat” to get the following metrics.

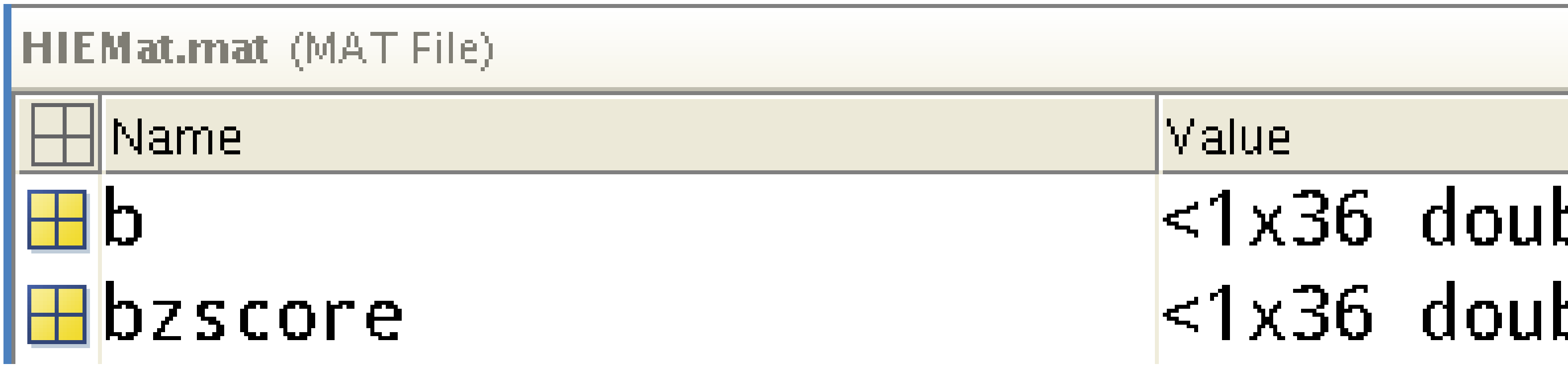
- **r**: Assortativity of network, 1×N array, N is the number of threshold sequences.

- **rzscore**: The z-score of assortativity of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$rzscore = \frac{r - mean(rrand)}{std(rrand)}$$

  rrand is a R×1 array, R is the number of randomized network. It is the assortativity of randomized network.



HIEMat.mat (MAT File)

Name	Value
b	<1x36 doub
bzscore	<1x36 doub

10.1.6 Network - Hierarchy

You can load ”HIEMat.mat” to get the following metrics.

- **b**: Hierarchy of network, 1×N array, N is the number of threshold sequences.

- **bzscore**: The z-score of hierarchy of network, 1×N array, N is the number of threshold sequences.

  The formula is following:

  $$bzscore = \frac{b - mean(brand)}{std(brand)}$$

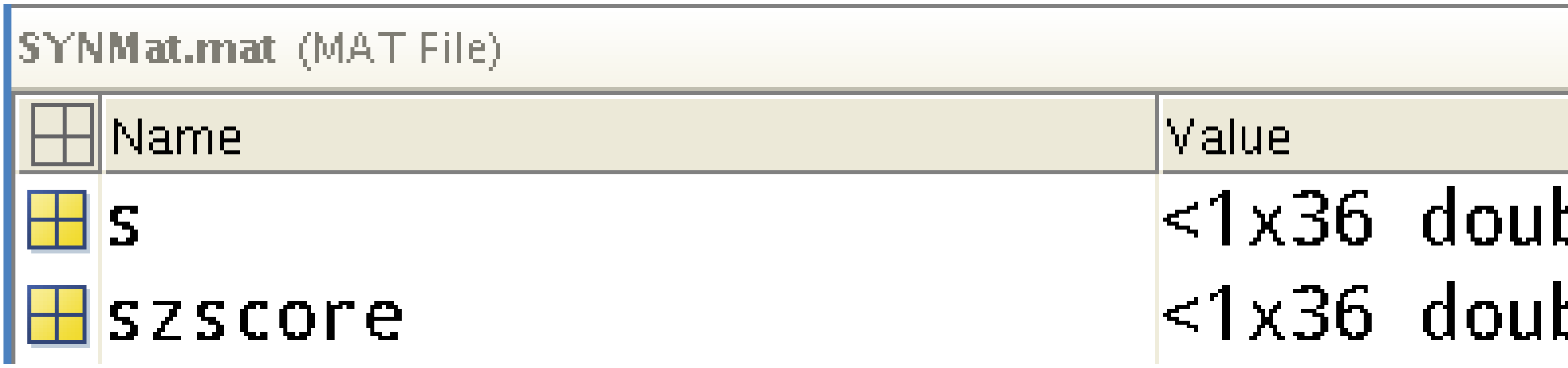
  brand is a R×1 array, R is the number of randomized network. It is the hierarchy of randomized network.

10.1.7 Network - Synchronization

You can load ”SYNMat.mat” to get the following metrics.

- **s**: Synchronization of network, 1×N array, N is the number of threshold sequences.

- **szscore**: The z-score of synchronization of network, 1×N array, N is the number of threshold sequences.



SYNMat.mat (MAT File)

Name	Value
s	<1x36 doub
szscore	<1x36 doub

The formula is following:

$$szscore = \frac{s - mean(srand)}{std(srand)}$$

srand is a R×1 array, R is the number of randomized network. It is the synchronization of randomized network.

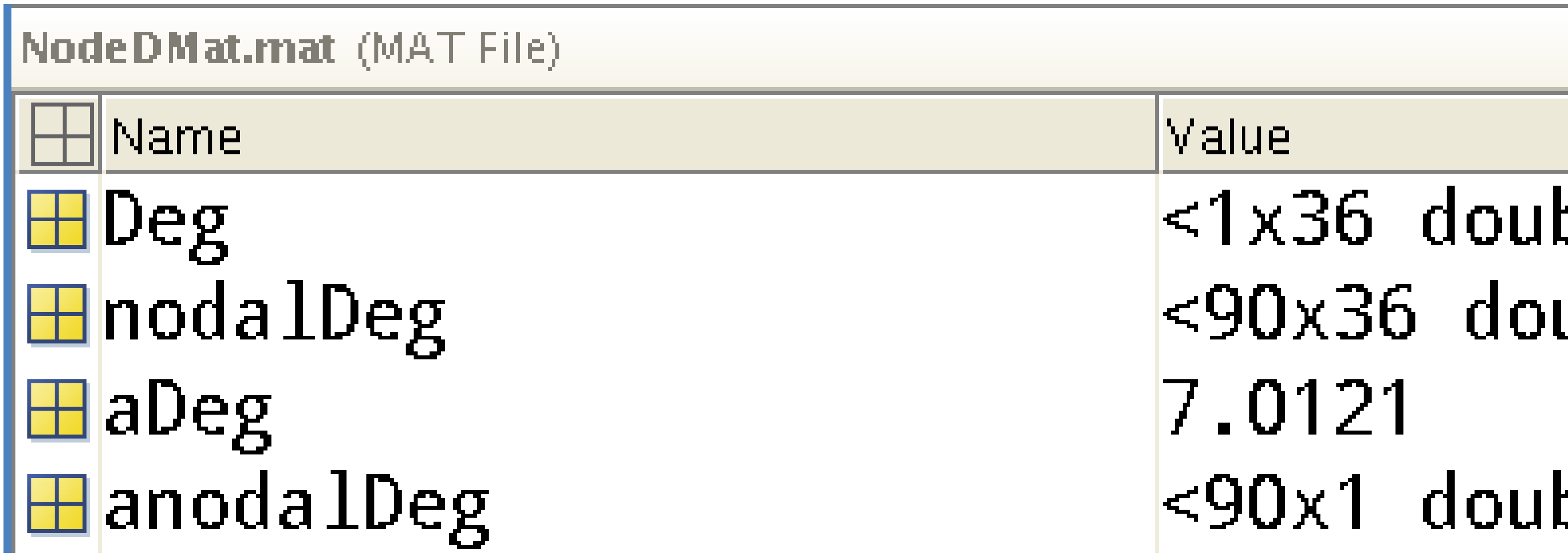
10.1.8 Node - Degree

You can load ”NodeDMat.mat” to get the following metrics.

- **Deg**: The degree of network, 1×N array, N is the number of threshold sequences.
- **nodalDeg**: The degree number of nodes, M×N array, M is the number of nodes, N is the number of threshold sequences.
- **aDeg**: The AUC of Deg, 1×N array, N is the number of threshold sequences.
- **anodalDeg**: The AUC of nodalDeg, M×N array, M is the number of nodes, N is the number of threshold sequences.

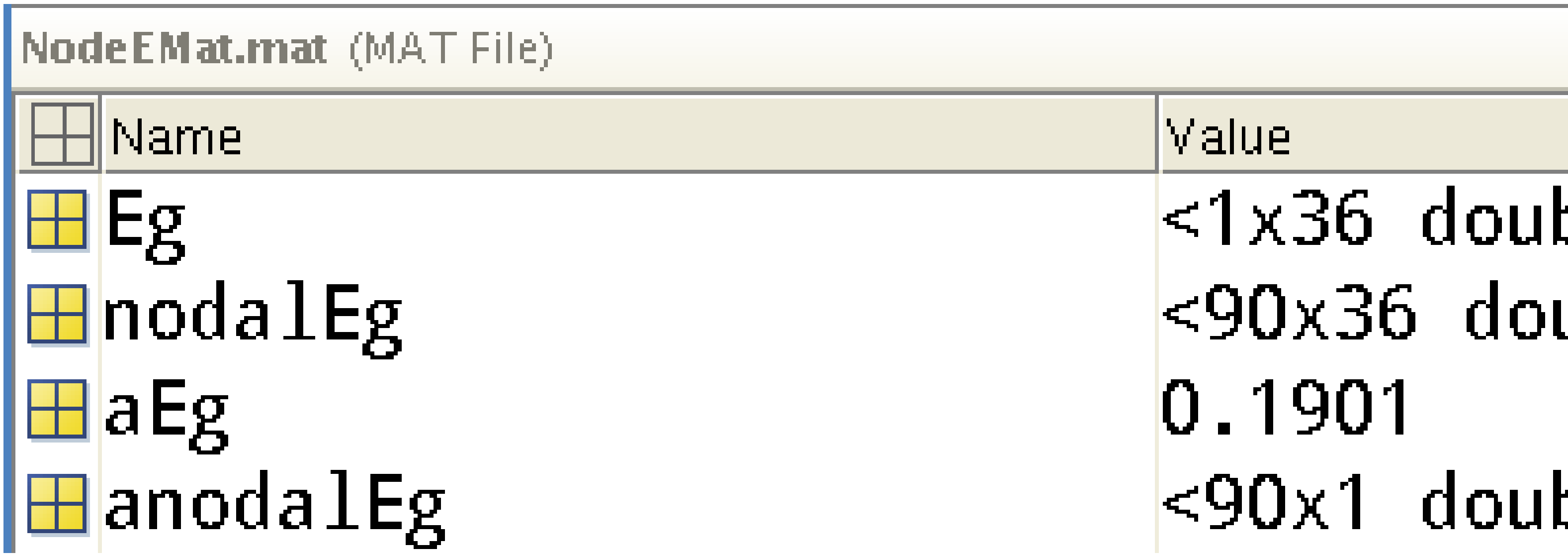
10.1.9 Node - Efficiency

You can load ”NodeEMat.mat” to get the following metrics.



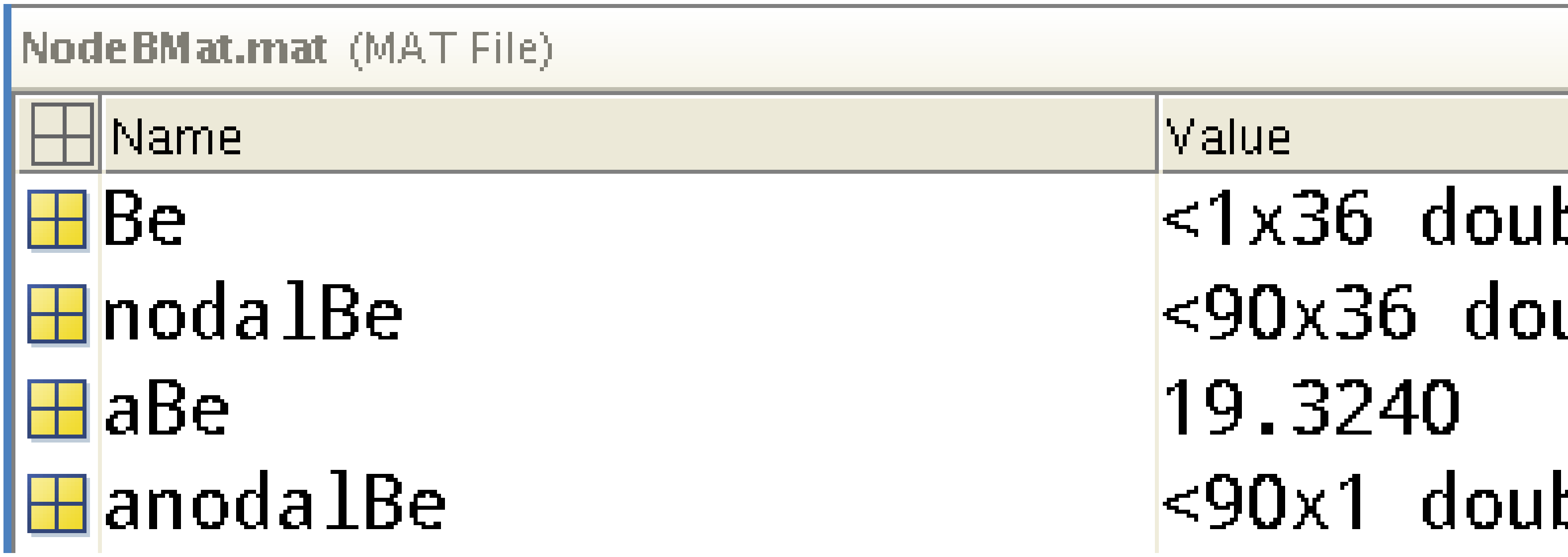
NodeDMat.mat (MAT File)

Name	Value
Deg	<1x36 doub
nodalDeg	<90x36 dou
aDeg	7.0121
anodalDeg	<90x1 doub



NodeEMat.mat (MAT File)

Name	Value
Eg	<1x36 doub
nodalEg	<90x36 dou
aEg	0.1901
anodalEg	<90x1 doub



NodeBMat.mat (MAT File)

Name	Value
Be	<1x36 dou
nodalBe	<90x36 do
aBe	19.3240
anodalBe	<90x1 dou

- **Eg**: The global efficiency of network, 1×N array, N is the number of threshold sequences.
- **nodalEg**: The global efficiency number of nodes, M×N array, M is the number of nodes, N is the number of threshold sequences.
- **aEg**: The AUC of Eg, 1×N array, N is the number of threshold sequences.
- **anodalEg**: The AUC of nodalEg, M×N array, M is the number of nodes, N is the number of threshold sequences.

10.1.10 Node - Betweenness

You can load ”NodeBMat.mat” to get the following metrics.

- **Be**: The betweenness of network, 1×N array, N is the number of threshold sequences.
- **nodalBe**: The betweenness number of nodes, M×N array, M is the number of nodes, N is the number of threshold sequences.

- **aBe**: The AUC of Be, 1×N array, N is the number of threshold sequences.

- **anodalBe**: The AUC of nodalBe, M×N array, M is the number of nodes, N is the number of threshold sequences.

10.2 Integrated Results

10.2.1 Network - Small World

You can load "SmallWorld.mat" or TEXT file which have the same names of metric to get the following metrics.

- **Cp(_All_Threshold)**: Clustering coefficient of network. M×N array, M is the number of subjects, N is the number of threshold sequences.

- **Lp(_All_Threshold)**: Shortest path length of network. M×N array, M is the number of subjects, N is the number of threshold sequences.

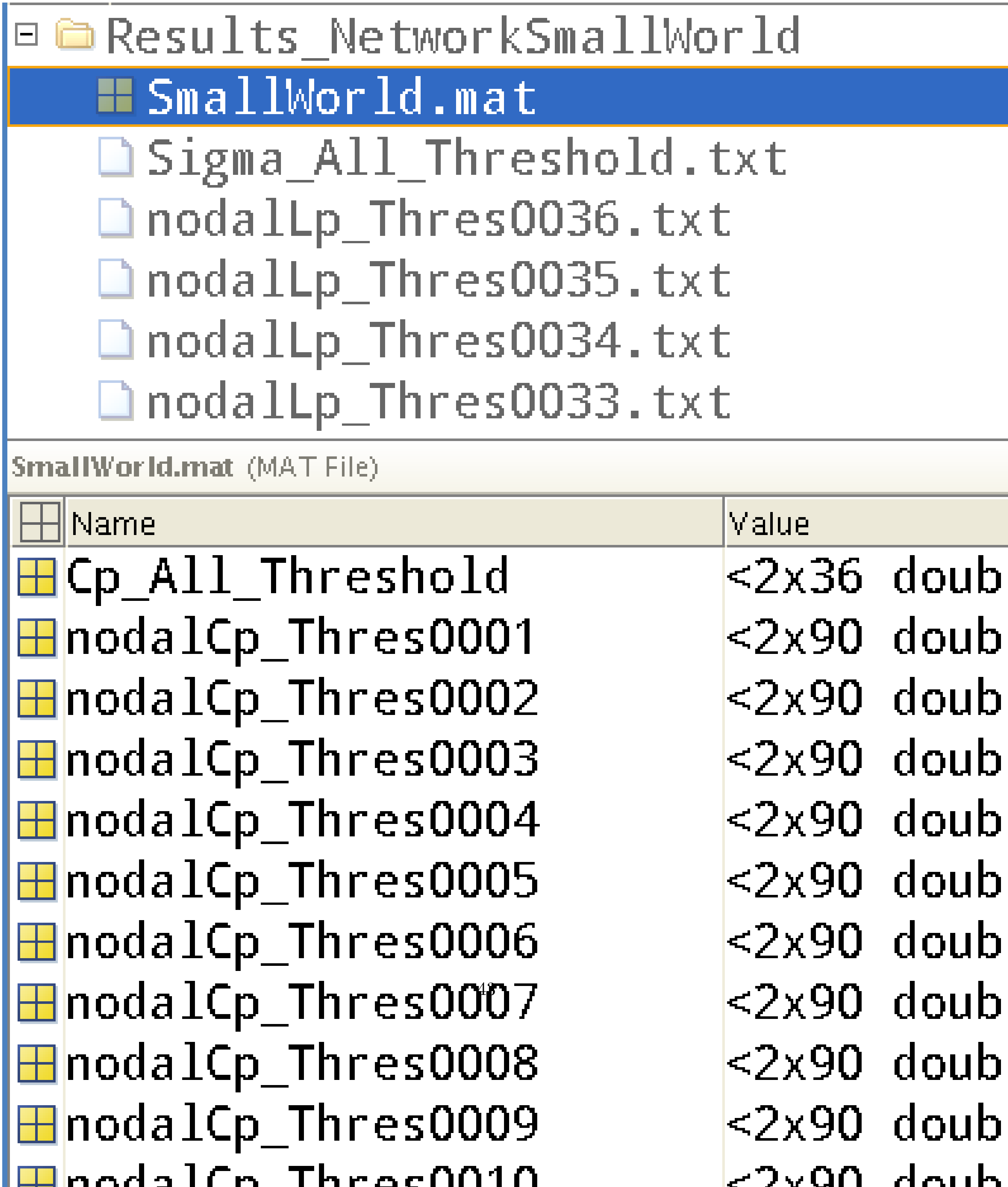
- **nodalCp_Thres***: Clustering coefficient number of nodes, "*" is the label of threshold sequence, M×N array, M is the number of subjects, N is the number of nodes.

- **nodalLp_Thres***: Shortest path length number of nodes, "*" is the label of threshold sequence, M×N array, M is the number of subjects, N is the number of nodes.

- **Cpzscore(_All_Threshold)**: The z-score of clustering coefficient of network, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

  $$Cpzscore = \frac{Cp - mean(Cprand)}{std(Cprand)}$$

  Cprand is a R×1 array, R is the number of randomized network. It is the clustering coefficient of randomized network.



```
⊟ Results_NetworkSmallWorld
      SmallWorld.mat
      Sigma_All_Threshold.txt
      nodalLp_Thres0036.txt
      nodalLp_Thres0035.txt
      nodalLp_Thres0034.txt
      nodalLp_Thres0033.txt
```

SmallWorld.mat (MAT File)

Name	Value
Cp_All_Threshold	<2x36 doub
nodalCp_Thres0001	<2x90 doub
nodalCp_Thres0002	<2x90 doub
nodalCp_Thres0003	<2x90 doub
nodalCp_Thres0004	<2x90 doub
nodalCp_Thres0005	<2x90 doub
nodalCp_Thres0006	<2x90 doub
nodalCp_Thres0007	<2x90 doub
nodalCp_Thres0008	<2x90 doub
nodalCp_Thres0009	<2x90 doub
nodalCp_Thres0010	<2x90 doub

- **Lpzscore(_All_Threshold)**: The z-score of shortest path length of network, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Lpzscore = \frac{Lp - mean(Lprand)}{std(Lprand)}$$

  Lprand is a R×1 array, R is the number of randomized network. It is the shortest path length of randomized network.

- **Gamma(_All_Threshold)**: Gamma is the ratio of Cp and mean value of Cprand, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Gamma = \frac{Cp}{mean(Cprand)}$$

- **Lambda(_All_Threshold)**: Lambda is the ratio of Lp and mean value of Lprand, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Lambda = \frac{Lp}{mean(Lprand)}$$

- **Sigma(_All_Threshold)**: Sigma is the ratio of Gamma and Lambda. M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Sigma = \frac{Gamma}{Lambda}$$

- **aCp**: The AUC (area under curve) of Cp. M×1 array, M is the number of subjects.

- **aLp**: The AUC of Lp. M×1 array, M is the number of subjects.

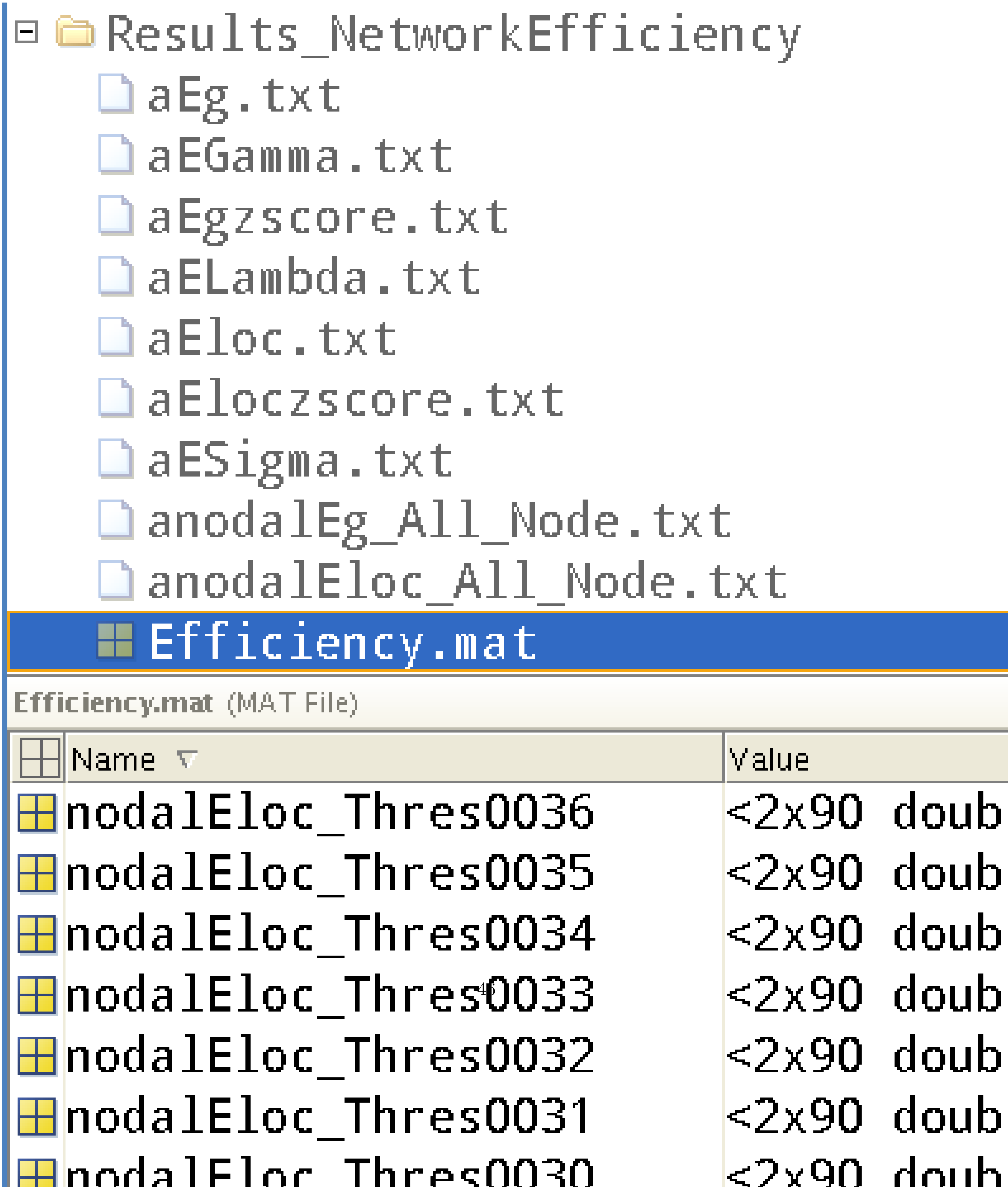
- **anodalCp_All_Node**: The AUC of nodalCp. M×N array, M is the number of subjects, N is the number of nodes.

- **anodalLp_All_Node**: The AUC of nodalLp. M×N array, M is the number of subjects, N is the number of nodes.
- **aCpzscore**: The AUC of Cpzscore. M×1 array, M is the number of subjects.
- **aLpzscore**: The AUC of Lpzscore. M×1 array, M is the number of subjects.
- **aGamma**: The AUC of Gamma. M×1 array, M is the number of subjects.
- **aLambda**: The AUC of Lambda. M×1 array, M is the number of subjects.
- **aSigma**: The AUC of Sigma. M×1 array, M is the number of subjects.

10.2.2 Network - Efficiency

You can load ”Efficiency.mat” or TEXT file which have the same names of metric to get the following metrics.

- **Eloc(_All_Threshold)**: Local efficiency of network. M×N array, M is the number of subjects, N is the number of threshold sequences.
- **Eg(_All_Threshold)**: Global efficiency of network. M×N array, M is the number of subjects, N is the number of threshold sequences.
- **nodalEloc_Thres***: Local efficiency number of nodes, ”*” is the label number of threshold sequences, M×N array, M is the number of subjects, N is the number of nodes.
- **nodalEg_Thres***: Global efficiency number of nodes, ”*” is the label number of threshold sequences, M×N array, M is the number of subjects, N is the number of nodes.



⊟ Results_NetworkEfficiency
- aEg.txt
- aEGamma.txt
- aEgzscore.txt
- aELambda.txt
- aEloc.txt
- aEloczscore.txt
- aESigma.txt
- anodalEg_All_Node.txt
- anodalEloc_All_Node.txt
- Efficiency.mat

Efficiency.mat (MAT File)

Name ▽	Value
nodalEloc_Thres0036	<2x90 doub
nodalEloc_Thres0035	<2x90 doub
nodalEloc_Thres0034	<2x90 doub
nodalEloc_Thres0033	<2x90 doub
nodalEloc_Thres0032	<2x90 doub
nodalEloc_Thres0031	<2x90 doub
nodalEloc_Thres0030	<2x90 doub

- **Eloczscore(_All_Threshold)**: The z-score of local efficiency of network, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Eloczscore = \frac{Eloc - mean(Elocrand)}{std(Elocrand)}$$

  Cprand is a R×1 array, R is the number of randomized network. It is the local efficiency of randomized network.

- **Egzscore(_All_Threshold)**: The z-score of global efficiency of network, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$Egzscore = \frac{Eg - mean(Egrand)}{std(Egrand)}$$

  Egrand is a R×1 array, R is the number of randomized network. It is the global efficiency of randomized network.

- **EGamma(_All_Threshold)**: EGamma is the ratio of Eloc and mean value of Elocrand, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$EGamma = \frac{Eloc}{mean(Elocrand)}$$

- **ELambda(_All_Threshold)**: ELambda is the ratio of Eg and mean value of Egrand, M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$ELambda = \frac{Eg}{mean(Egrand)}$$

- **ESigma(_All_Threshold)**: ESigma is the ratio of EGamma and ELambda. M×N array, M is the number of subjects, N is the number of threshold sequences. The formula is following:

$$ESigma = \frac{EGamma}{ELambda}$$

- **aEloc**: The AUC (area under curve) of Eloc. M×1 array, M is the number of subjects.

- **aEg**: The AUC of Eg. M×1 array, M is the number of subjects.

- **anodalEloc_All_Node**: The AUC of nodalEloc. M×N array, M is the number of subjects, N is the number of nodes.

- **anodalEg_All_Node**: The AUC of nodalEg. M×N array, M is the number of subjects, N is the number of nodes.

- **aEloczscore**: The AUC of Eloczscore. M×1 array, M is the number of subjects.

- **aEgzscore**: The AUC of Egzscore. M×1 array, M is the number of subjects.

- **aEGamma**: The AUC of EGamma. M×1 array, M is the number of subjects.

- **aELambda**: The AUC of ELambda. M×1 array, M is the number of subjects.

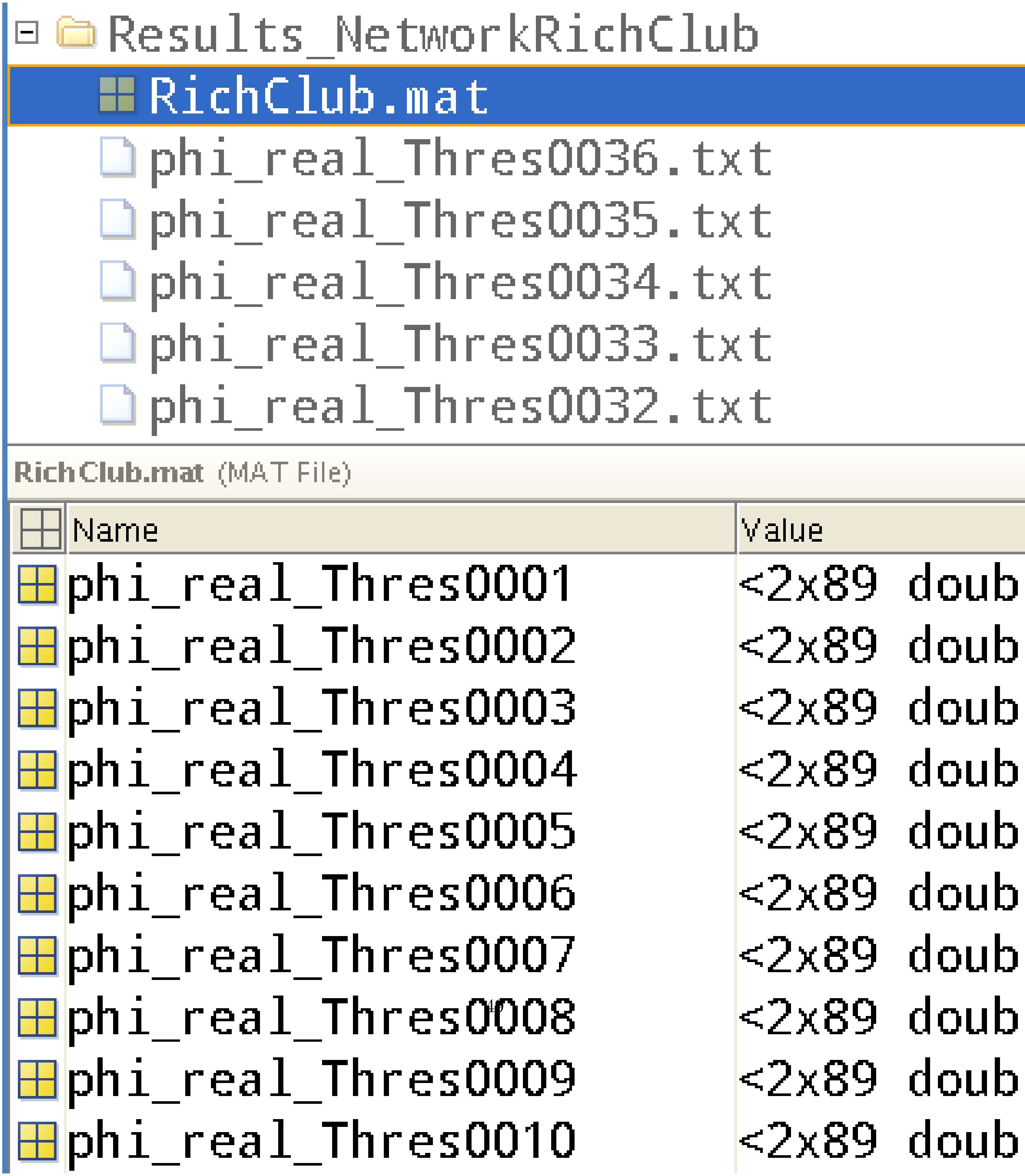
- **aESigma**: The AUC of ESigma. M×1 array, M is the number of subjects.

10.2.3 Network - Rich Club

You can load "RichClub.mat" or TEXT file which have the same names of metric to get the following metrics.

- **phi_real_Thres***: The rich club coefficient of real network, "*" is the label number of threshold sequences, M×K array, M is the number of subjects, K is the number of binary degree, from 1 to Node-1.

- **phi_norm_Thres***: The normalized rich club coefficient of real network, "*" is the label number of threshold sequences, M×K array, M is the number of subjects, K is the number of binary degree, from 1 to Node-1.



```
⊟ Results_NetworkRichClub
      RichClub.mat
      phi_real_Thres0036.txt
      phi_real_Thres0035.txt
      phi_real_Thres0034.txt
      phi_real_Thres0033.txt
      phi_real_Thres0032.txt
```

RichClub.mat (MAT File)

Name	Value
phi_real_Thres0001	<2x89 doub
phi_real_Thres0002	<2x89 doub
phi_real_Thres0003	<2x89 doub
phi_real_Thres0004	<2x89 doub
phi_real_Thres0005	<2x89 doub
phi_real_Thres0006	<2x89 doub
phi_real_Thres0007	<2x89 doub
phi_real_Thres0008	<2x89 doub
phi_real_Thres0009	<2x89 doub
phi_real_Thres0010	<2x89 doub

10.2.4 Network - Modularity

You can load ”Modularity.mat” or TEXT file which have the same names of metric to get the following metrics.

- **community_index_Thres***: The community (listed for each node), ”*” is the number of threshold sequences, M×N array, M is the number of subjects, N is the number of nodes.
- **number_of_module**: The number of module in network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **modularity**: Modularity value of network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **modularity_zscore**: The z-score of modularity of network, M×N array, M is the number of subjects, N is the number of threshold sequences.

  The formula is following:

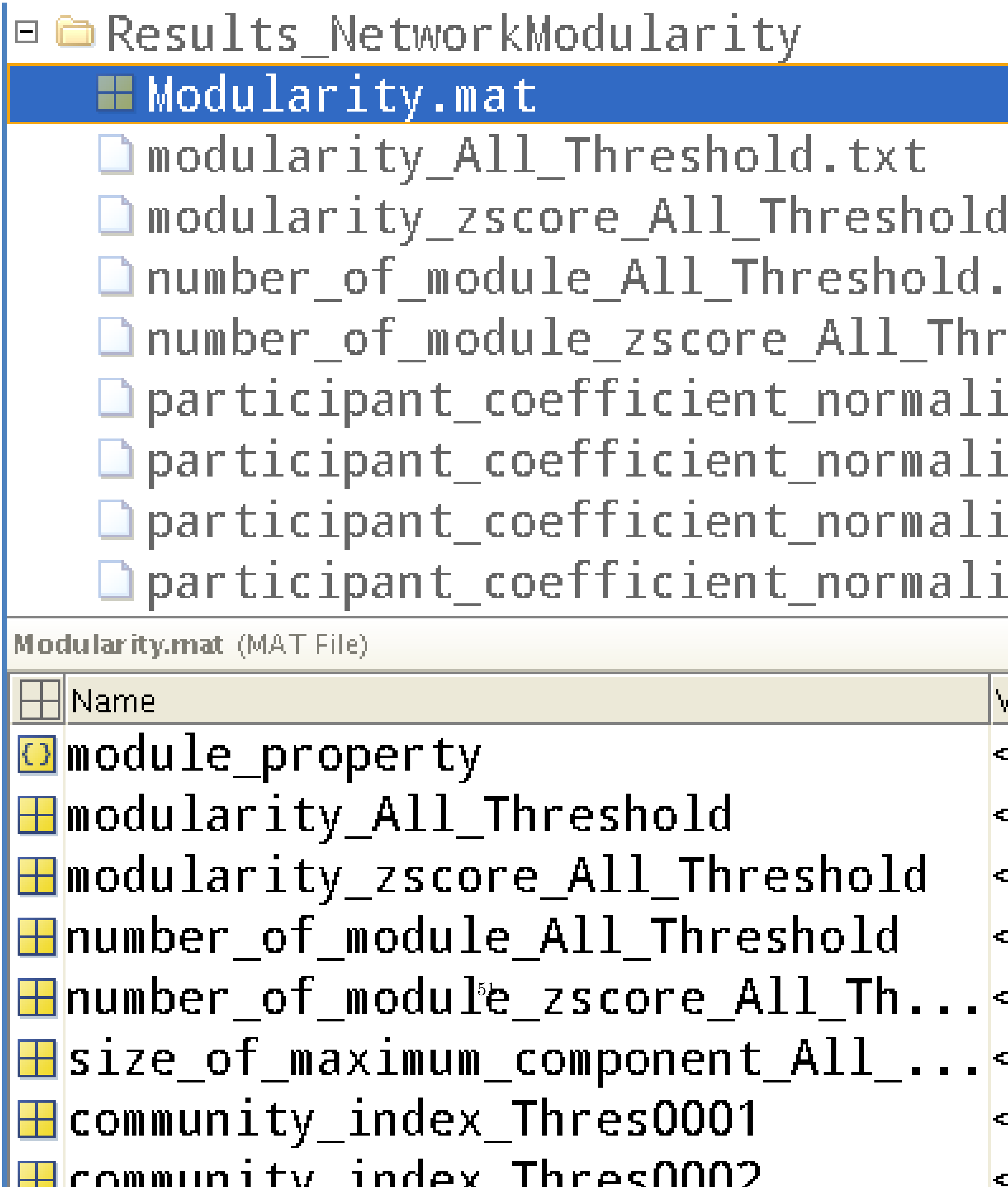
  $$modularity\_zscore = \frac{modularity - mean(modrand)}{std(modrand)}$$

  modrand is a R×1 array, R is the number of randomized network. It is the modularity of randomized network.

10.2.5 Network - Assortativity

You can load ”Assortativity.mat” or TEXT file which have the same names of metric to get the following metrics.

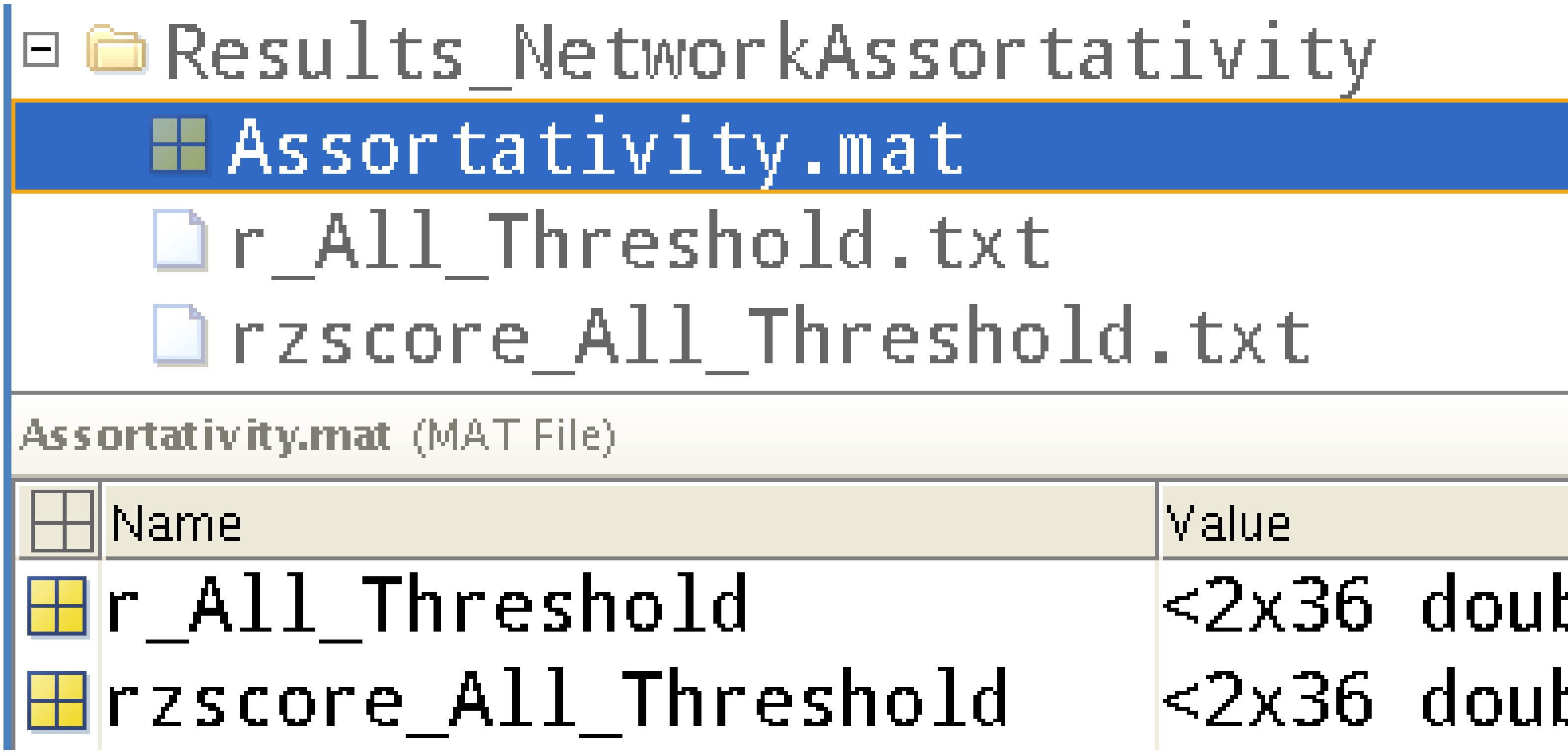
- **r**: Assortativity of network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **rzscore**: The z-score of assortativity of network, M×N array, M is the number of subjects, N is the number of threshold sequences.



```
⊟ Results_NetworkModularity
    Modularity.mat
    modularity_All_Threshold.txt
    modularity_zscore_All_Threshold
    number_of_module_All_Threshold.
    number_of_module_zscore_All_Thr
    participant_coefficient_normali
    participant_coefficient_normali
    participant_coefficient_normali
    participant_coefficient_normali
```

Modularity.mat (MAT File)

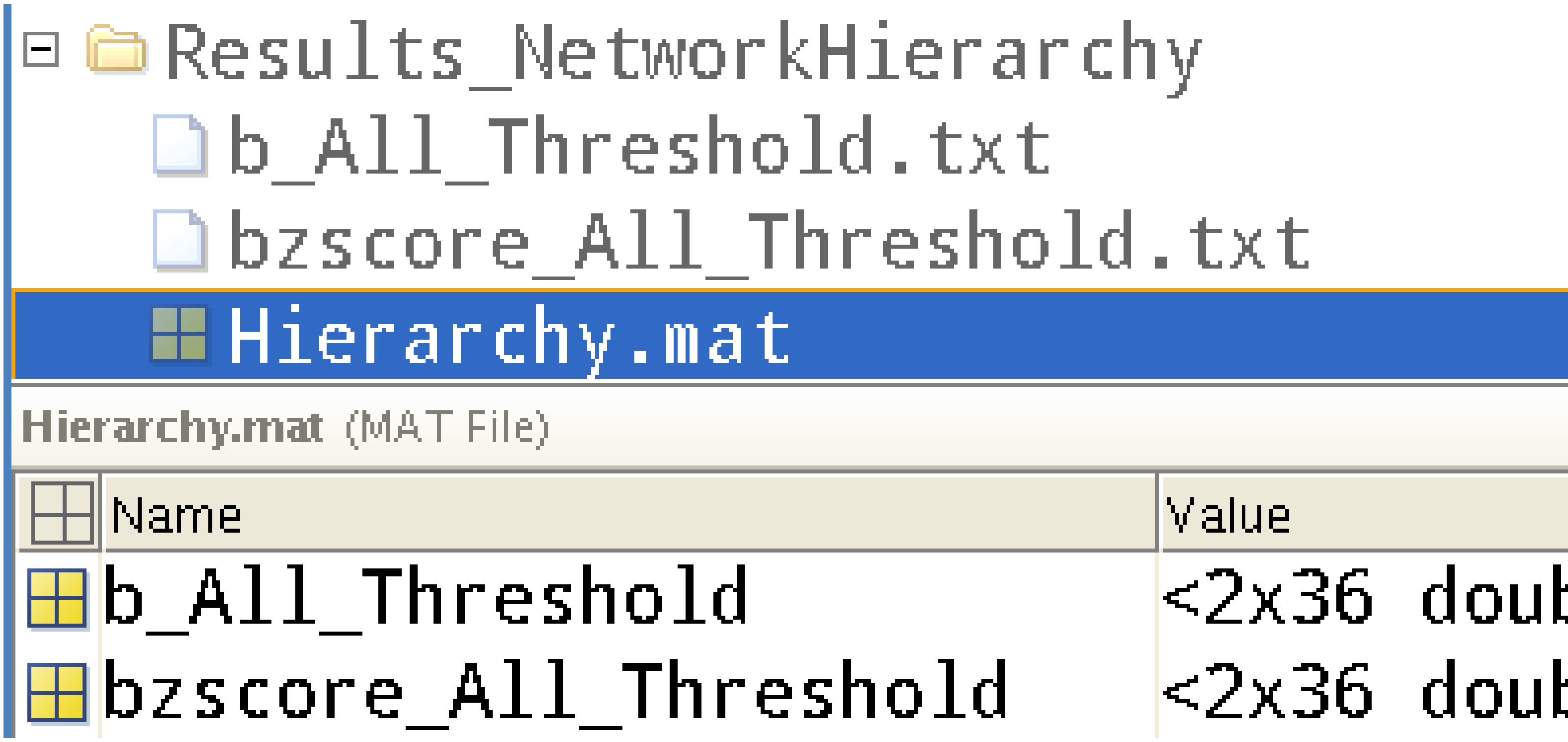
Name	V
module_property	<
modularity_All_Threshold	<
modularity_zscore_All_Threshold	<
number_of_module_All_Threshold	<
number_of_module_zscore_All_Th...	<
size_of_maximum_component_All_...	<
community_index_Thres0001	<
community_index_Thres0002	<



```
Results_NetworkAssortativity
    Assortativity.mat
    r_All_Threshold.txt
    rzscore_All_Threshold.txt
```

Assortativity.mat (MAT File)

Name	Value
r_All_Threshold	<2x36 doub
rzscore_All_Threshold	<2x36 doub



Results_NetworkHierarchy
b_All_Threshold.txt
bzscore_All_Threshold.txt
Hierarchy.mat

Hierarchy.mat (MAT File)

Name	Value
b_All_Threshold	<2x36 doub
bzscore_All_Threshold	<2x36 doub

The formula is following:

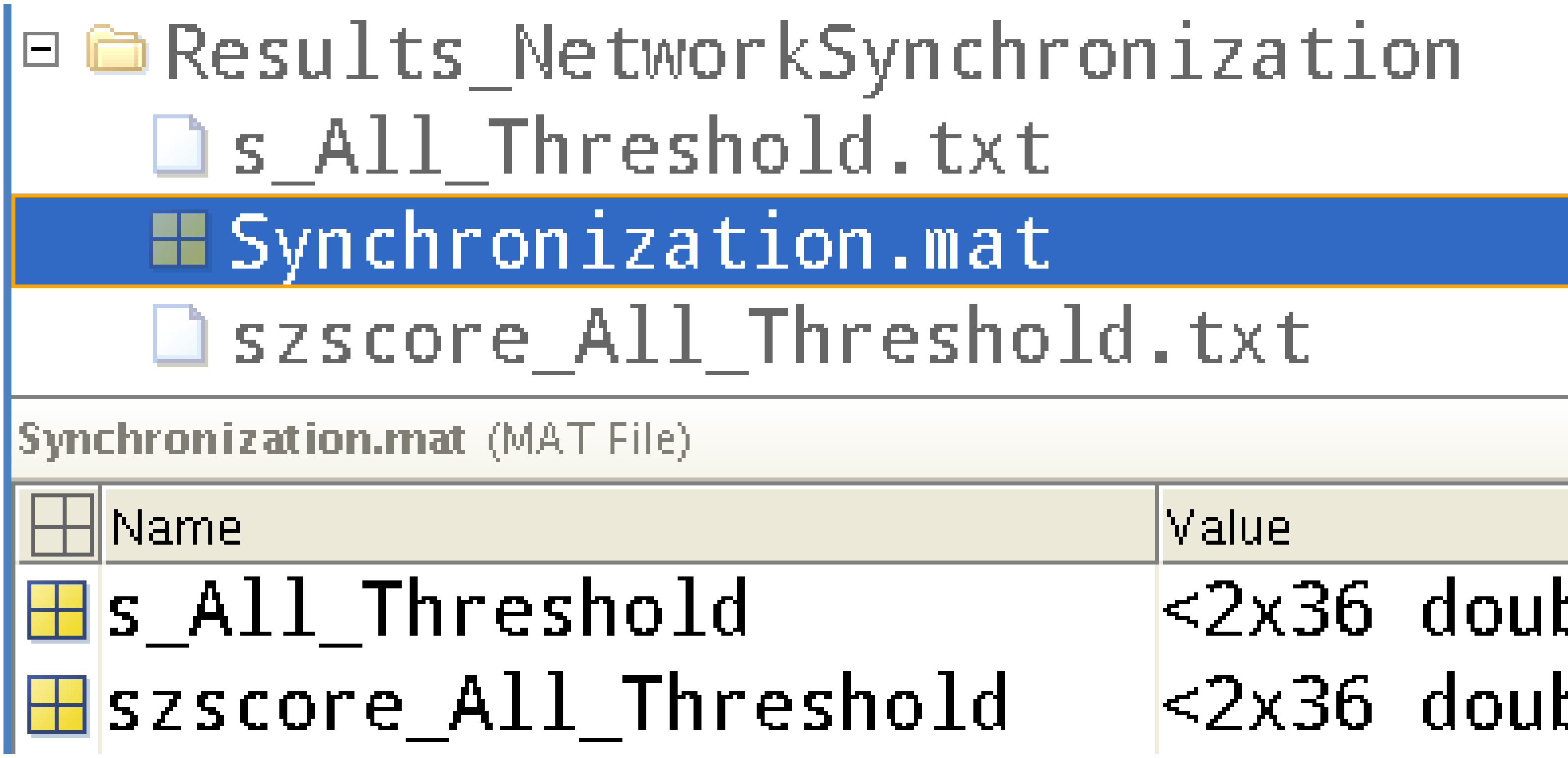
$$rzscore = \frac{r - mean(rrand)}{std(rrand)}$$

rrand is a R×1 array, R is the number of randomized network. It is the assortativity of randomized network.

10.2.6 Network - Hierarchy

You can load "Hierarchy.mat" or TEXT file which have the same names of metric to get the following metrics.

- **b**: Hierarchy of network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **bzscore**: The z-score of hierarchy of network, M×N array, M is the number of subjects, N is the number of threshold sequences.



```
Results_NetworkSynchronization
    s_All_Threshold.txt
    Synchronization.mat
    szscore_All_Threshold.txt
```

Synchronization.mat (MAT File)

Name	Value
s_All_Threshold	<2x36 doub
szscore_All_Threshold	<2x36 doub

The formula is following:

$$bzscore = \frac{b - mean(brand)}{std(brand)}$$

brand is a R×1 array, R is the number of randomized network. It is the hierarchy of randomized network.

10.2.7 Network - Synchronization

You can load "Synchronization.mat" or TEXT file which have the same names of metric to get the following metrics.

- **s**: Synchronization of network, M×N array, M is the number of subjects, N is the number of threshold sequences.

- **szscore**: The z-score of synchronization of network, M×N array, M is the number of subjects, N is the number of threshold sequences.

The formula is following:

$$szscore = \frac{s - mean(srand)}{std(srand)}$$

srand is a R×1 array, R is the number of randomized network. It is the synchronization of randomized network.

10.2.8 Node - Degree

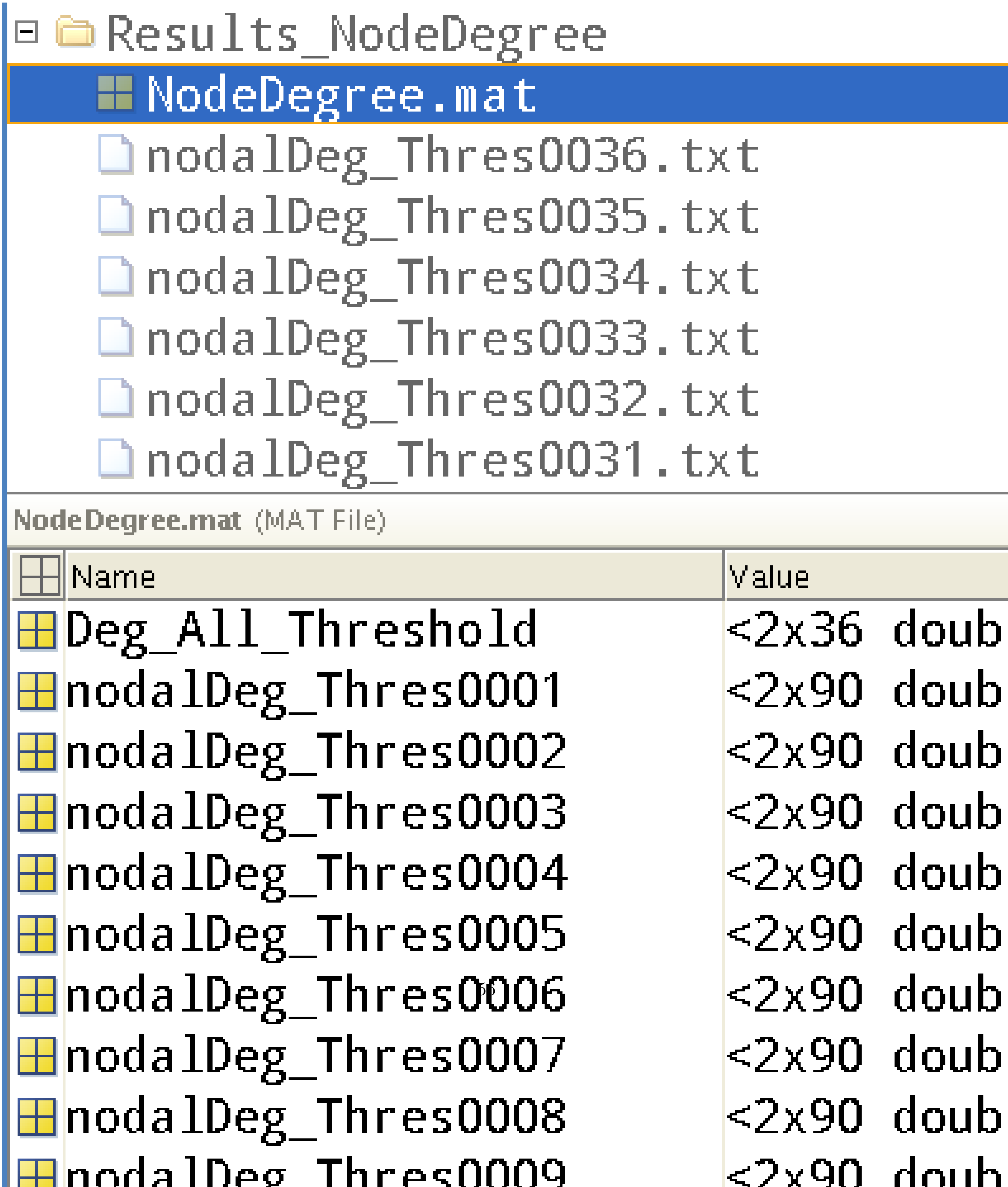
You can load "NodeDegree.mat" or TEXT file which have the same names of metric to get the following metrics.

- **Deg**: The degree of network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **nodalDeg_Thres***: The degree number of nodes, "*" is the label number of threhold sequences, M×N array, M is the number of subjects, N is the number of nodes.
- **aDeg**: The AUC (area under curve) of Deg, M×1 array, M is the number of subjects.
- **anodalDeg_All_Threshold**: The AUC of nodalDeg, M×N array, M is the number of subjects, N is the number of nodes.

10.2.9 Node - Efficiency

You can load "NodeEfficiency.mat" or TEXT file which have the same names of metric to get the following metrics.

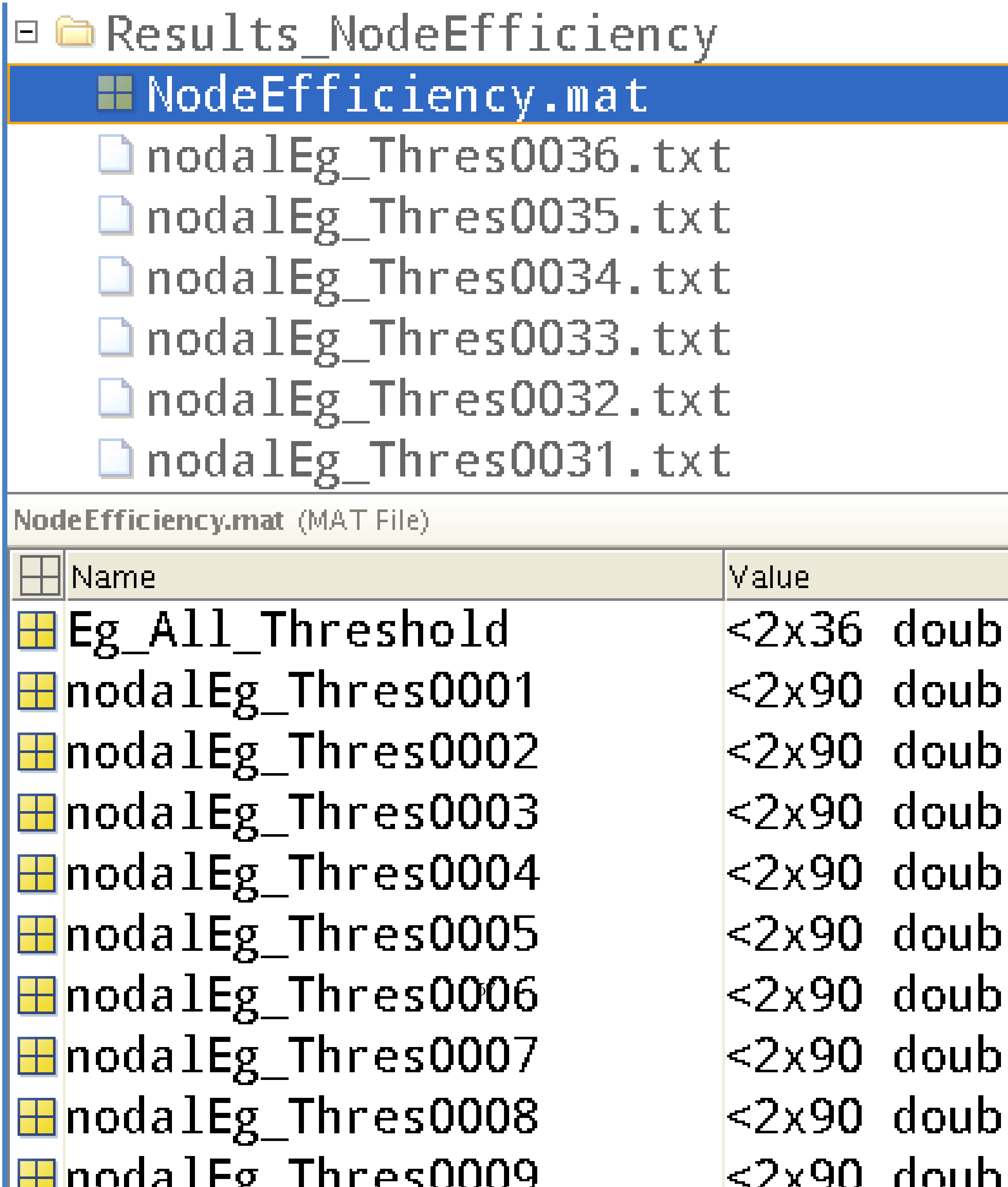
- **Eg**: The global efficiency of network, M×N array, M is the number of subjects, N is the number of threshold sequences.
- **nodalEg_Thres***: The global efficiency number of nodes, "*" is the label number of threshold sequences, M×N array, M is the number of subjects, N is the number of nodes.



```
⊟ Results_NodeDegree
      NodeDegree.mat
      nodalDeg_Thres0036.txt
      nodalDeg_Thres0035.txt
      nodalDeg_Thres0034.txt
      nodalDeg_Thres0033.txt
      nodalDeg_Thres0032.txt
      nodalDeg_Thres0031.txt
```

NodeDegree.mat (MAT File)

Name	Value
Deg_All_Threshold	<2x36 doub
nodalDeg_Thres0001	<2x90 doub
nodalDeg_Thres0002	<2x90 doub
nodalDeg_Thres0003	<2x90 doub
nodalDeg_Thres0004	<2x90 doub
nodalDeg_Thres0005	<2x90 doub
nodalDeg_Thres0006	<2x90 doub
nodalDeg_Thres0007	<2x90 doub
nodalDeg_Thres0008	<2x90 doub
nodalDeg_Thres0009	<2x90 doub



⊟ Results_NodeEfficiency

- NodeEfficiency.mat
- nodalEg_Thres0036.txt
- nodalEg_Thres0035.txt
- nodalEg_Thres0034.txt
- nodalEg_Thres0033.txt
- nodalEg_Thres0032.txt
- nodalEg_Thres0031.txt

NodeEfficiency.mat (MAT File)

Name	Value
Eg_All_Threshold	<2x36 doub
nodalEg_Thres0001	<2x90 doub
nodalEg_Thres0002	<2x90 doub
nodalEg_Thres0003	<2x90 doub
nodalEg_Thres0004	<2x90 doub
nodalEg_Thres0005	<2x90 doub
nodalEg_Thres0006	<2x90 doub
nodalEg_Thres0007	<2x90 doub
nodalEg_Thres0008	<2x90 doub
nodalEg_Thres0009	<2x90 doub

- **aEg**: The AUC (area under curve) of Eg, M×1 array, M is the number of subjects.

- **anodalEg_All_Threshold**: The AUC of nodalEg, M×N array, M is the number of subjects, N is the number of nodes.

10.2.10 Node - Betweenness

You can load "NodeBetweenness.mat" or TEXT file which have the same names of metric to get the following metrics.

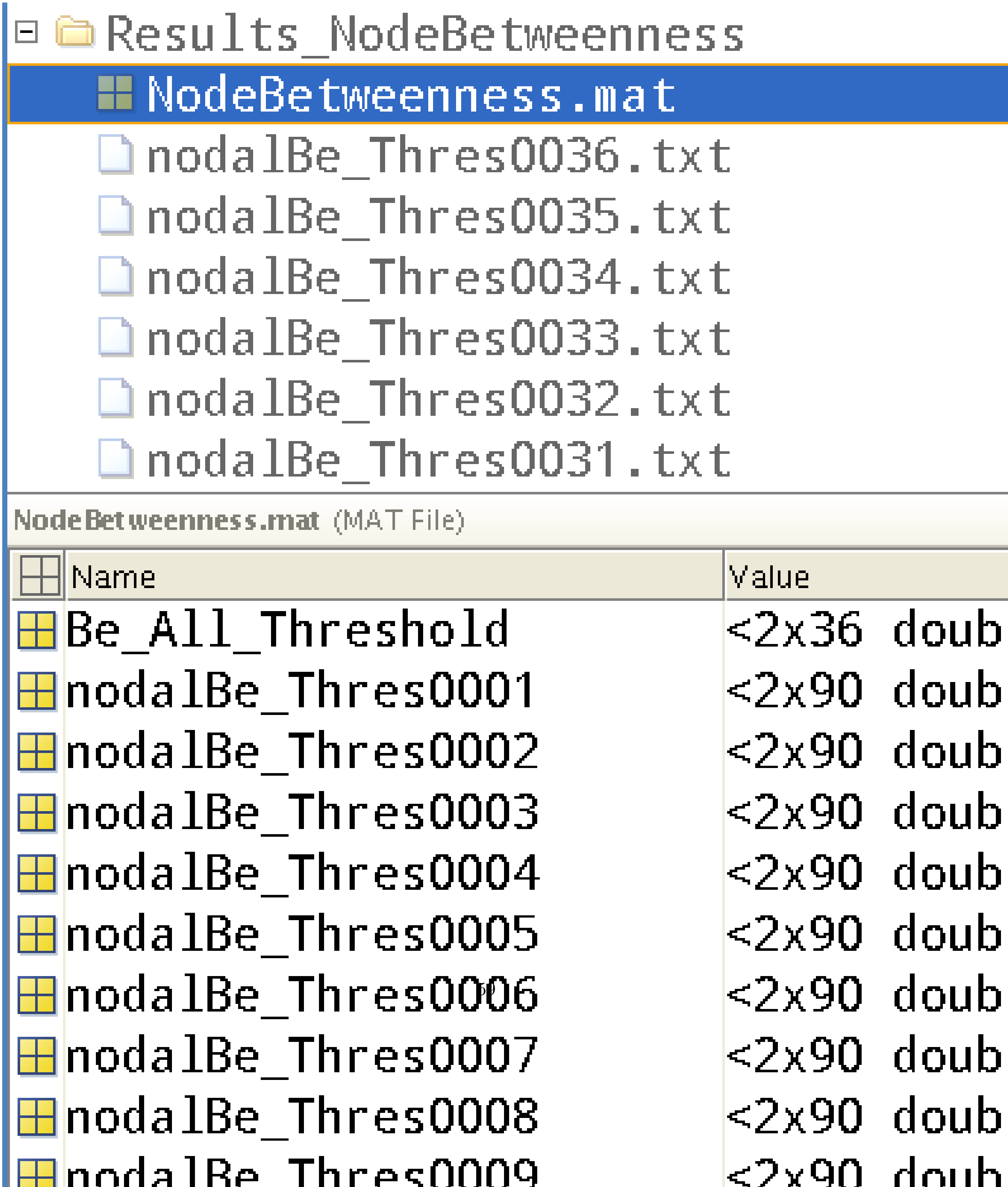
- **Be**: The betweenness of network, M×N array, M is the number of subjects, N is the number of threshold sequences.

- **nodalBe_Thres***: The betweenness number of nodes, "*" is the label number of threshold sequences, M×N array, M is the number of subjects, N is the number of nodes.

- **aBe**: The AUC of Be, M×1 array, M is the number of subjects.

- **anodalBe_All_Threshold**: The AUC of nodalBe, M×N array, M is the number of subjects, N is the number of nodes.

11 Network Comparison



Results_NodeBetweenness

NodeBetweenness.mat

nodalBe_Thres0036.txt

nodalBe_Thres0035.txt

nodalBe_Thres0034.txt

nodalBe_Thres0033.txt

nodalBe_Thres0032.txt

nodalBe_Thres0031.txt

NodeBetweenness.mat (MAT File)

Name	Value
Be_All_Threshold	<2x36 doub
nodalBe_Thres0001	<2x90 doub
nodalBe_Thres0002	<2x90 doub
nodalBe_Thres0003	<2x90 doub
nodalBe_Thres0004	<2x90 doub
nodalBe_Thres0005	<2x90 doub
nodalBe_Thres0006	<2x90 doub
nodalBe_Thres0007	<2x90 doub
nodalBe_Thres0008	<2x90 doub
nodalBe_Thres0009	<2x90 doub