

## **CCS: Connectome Computation System**

## **Abstract**

Many neuroscientists prefer to explore the mysteries of brain by Magnetic Resonance Imaging (MRI) now because of its non-invasive and high spatial and temporal resolution. With the wide application of this technology in scientific research institutes and clinical institutions all over the world, a variety of large sample and multi-center open databases have been gradually established. The launch of connectome projects in many countries also provides chances for the exploration of brain functional connectome that based on large-scale neuroimaging. Meanwhile, the ability to process tons of data has been challenged. With the advent of the “big data” era of human brain connectome, a simpler and more efficient computing platform is needed, which can process imaging data and provide parallel computing, so as to shorten the data processing time. Connectome Computation System (CCS), integrates the functions of three major MRI analysis software (AFNI, FSL and FreeSurfer), forms a configurable, reliable, and extendable MRI data processing platform based on the current research progress of MRI methodology.

Although there are many data mining, analysis and visualization (MAV) tools available now, such as SPM, FSL, AFNI and FreeSurfer, widely accepted standards for developing MAV tools and related algorithms are still lacking. Besides, these tools do not provide automatic and parallel computing scripts. In recent years, some software packages have made efforts to solve these problems, trying to provide users with the simplest operation mode for the analysis of MRI data. However, they are running serially and cannot meet the needs of multi-center data processing. CCS, based on parallel computing, transfers the original computing on supercomputer to the small workstation with GPU, which greatly improves the computing efficiency. CCS can preprocess the structural MRI (sMRI) data, resting state functional MRI (rfMRI) data, diffusion MRI (dMRI) data and calculate the individual functional map. In addition, compared with other tools, CCS has four functional modules: quality control module, test-retest reliability and reproducibility evaluation module, connectome-wide association module and visualization module. The calculation of these four modules can be completed at the terminal. At the same time, for the convenience of users, a CCS graphical interface was developed by Python (pyccs), which can generate the command-line program for the operation of each step, keep the running log, and record the inspection, debugging and data backup in detail.

## Contents

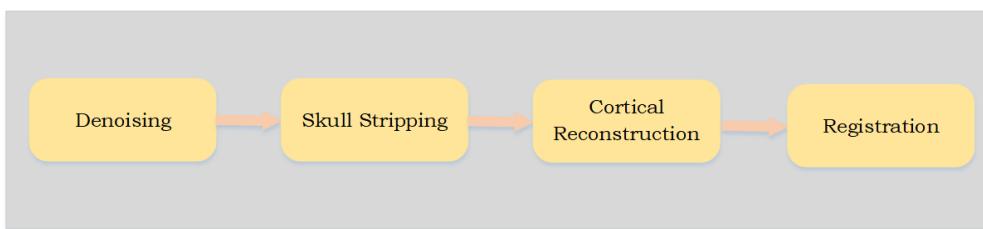
|  |    |
|--|----|
| Abstract.....  | 1  |
| Chapter 1 Preprocessing of Structural Magnetic Resonance Imaging .....               | 1  |
| 1.1 Denoising .....  | 1  |
| 1.2 Skull Stripping .....  | 2  |
| 1.3 Cortical Reconstruction.....   | 3  |
| 1.4 Registration.....  | 4  |
| Chapter 2 Preprocessing of Resting-state Functional Magnetic Resonance Imaging ..... | 5  |
| 2.1 Basic Processing .....   | 5  |
| 2.2 Registration.....  | 6  |
| 2.3 Segmentation.....  | 7  |
| 2.4 Nuisance Covariates Regression .....   | 7  |
| Chapter 3 Preprocessing of Diffusion Tensor Imaging .....                            | 9  |
| 3.1 Basic Processing .....   | 9  |
| 3.2 Registration.....  | 10 |
| 3.3 Segmentation.....  | 11 |
| Chapter 4 Individual Structural Map.....   | 12 |
| 4.1 Morphology .....   | 12 |
| 4.2 Connectivity of Fiber Tracts .....   | 13 |
| Chapter 5 Individual Functional Map (3D) .....                                       | 14 |
| 5.1 Amplitude of Low-Frequency Fluctuation.....                                      | 14 |
| 5.2 Regional Homogeneity.....  | 15 |
| 5.3 Seed-based Functional Connectivity.....  | 15 |
| 5.4 Interhemispheric Functional Connectivity .....                                   | 16 |
| 5.5 Independent Component Analysis.....  | 16 |
| 5.6 Network Centrality.....  | 17 |

|   |           |
|---|-----------|
| <b>Chapter 6 Individual Functional Map (2D) .....</b>                         | <b>18</b> |
| 6.1 Amplitude of Low-Frequency Fluctuation.....                               | 19        |
| 6.2 Regional Homogeneity.....   | 19        |
| 6.3 Seed-based Functional Connectivity.....                                   | 20        |
| 6.4 Interhemispheric Functional Connectivity.....                             | 20        |
| 6.5 Independent Component Analysis.....                                       | 20        |
| 6.6 Network Centrality.....   | 20        |
| <b>Chapter 7 Quality Control .....</b>  | <b>21</b> |
| 7.1 Denoising .....   | 21        |
| 7.2 Cortical Reconstruction.....  | 22        |
| 7.3 Registration.....   | 22        |
| 7.4 Segmentation.....   | 22        |
| 7.5 Head Motion .....   | 22        |
| <b>Chapter 8 Test-Retest Reliability.....</b>                                 | <b>25</b> |
| 8.1 Test-Retest Reliability .....   | 26        |
| 8.2 Reproducibility .....   | 26        |
| <b>Chapter 9 Connectome-Wide Association Module.....</b>                      | <b>27</b> |
| 9.1 Linear Model and Statistical Analysis.....                                | 27        |
| 9.2 Connectome-wide Association Mining and Unsupervised Machine Learning..... | 27        |
| 9.3 Supervised Machine Learning Model.....                                    | 28        |
| 9.4 Multimodal Image Integration Analysis .....                               | 29        |
| <b>Chapter 10 Visualization .....</b>   | <b>31</b> |
| 10.1 Basic Elements .....   | 31        |
| 10.2 Toolbox .....  | 31        |
| <b>Supplementary Material    CCS Sample Scripts.....</b>                      | <b>33</b> |
| A. CCS Installation and Usage .....   | 34        |
| B. Calculation of Connectome Map .....  | 37        |

# **Chapter 1 Preprocessing of Structural Magnetic Resonance Imaging**

CCS optimizes the preprocessing of structural Magnetic Resonance Imaging (sMRI) by applying various structure image processing software through script.

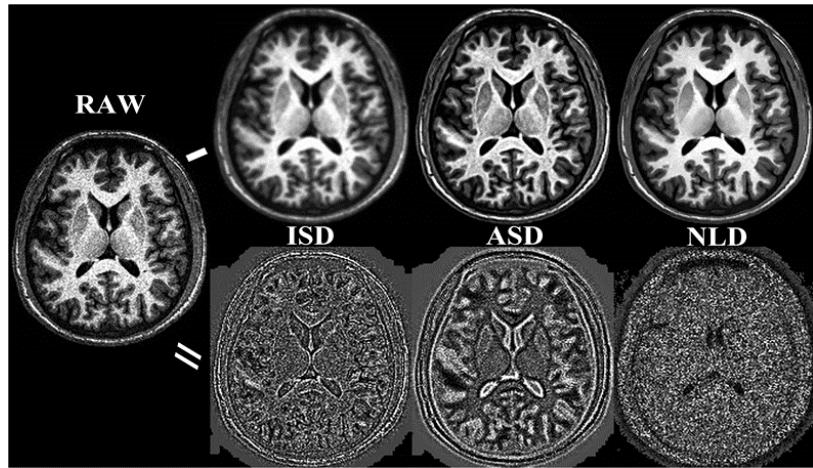
T1 weighted image has been processed by denoising based on non-local means diffusion algorithm, optimized skull stripping, cortical surface reconstruction, cortical parcellation based on morphometry, and linear and nonlinear registration of the cortical surface. These procedures lay a foundation for subsequent integration of structural and functional data and statistical analysis at individual and group level. Figure 1.1 shows the logical relationship and execution sequence between these processing steps.



**Figure 1.1 Flow chart of sMRI preprocessing**

## **1.1 Denoising**

Generally, noises are inevitable in MRI data (Figure 2, RAW). How to remove these noises has always been a hotspot in medical image processing. There are three main algorithms for image denoising: 1) Gaussian smoothing (isotropic diffusion equation, ISD), 2) anisotropic diffusion equations (ASD), 3) non-local means diffusion (NLD). ISD is the most widely used, but it has obvious disadvantages. As showed in figure 1.2, ISD largely reduces the sharpness of GM-WM tissue boundaries while suppressing the noise, depicted as the overall boundary pattern in the difference image. By contrast, the ASD does good job in preserving the boundaries. However, it still distorts textures which is clearly presented in the ASD-RAW difference images. Moreover, the ASD can also introduce flow-like artifacts along tissue boundaries. The NLD produces rather uniform noise distributions, achieving promising removal of noise.

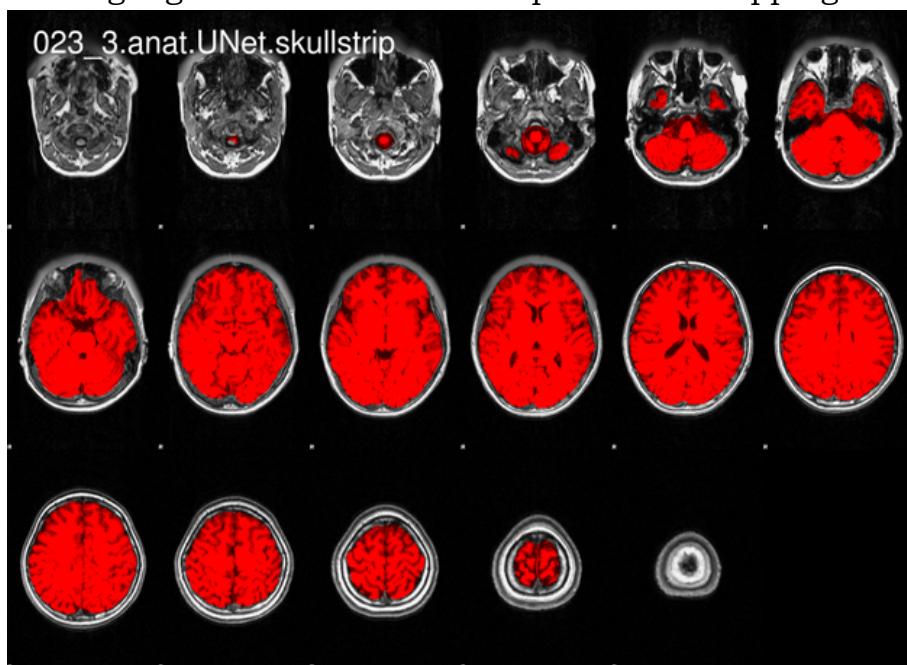


**Figure 1.2 Comparison of common sMRI denoising algorithms**

## 1.2 Skull Stripping

Skull is wrapped in the outer layer of brain tissue, plays a protective role. As the basis of subsequent image processing, stripping the skull from brain tissue is the most important first step. CCS combines the existing automatic skull stripping algorithm to create alternative stripping images to provide decision-making basis for manual intervention.

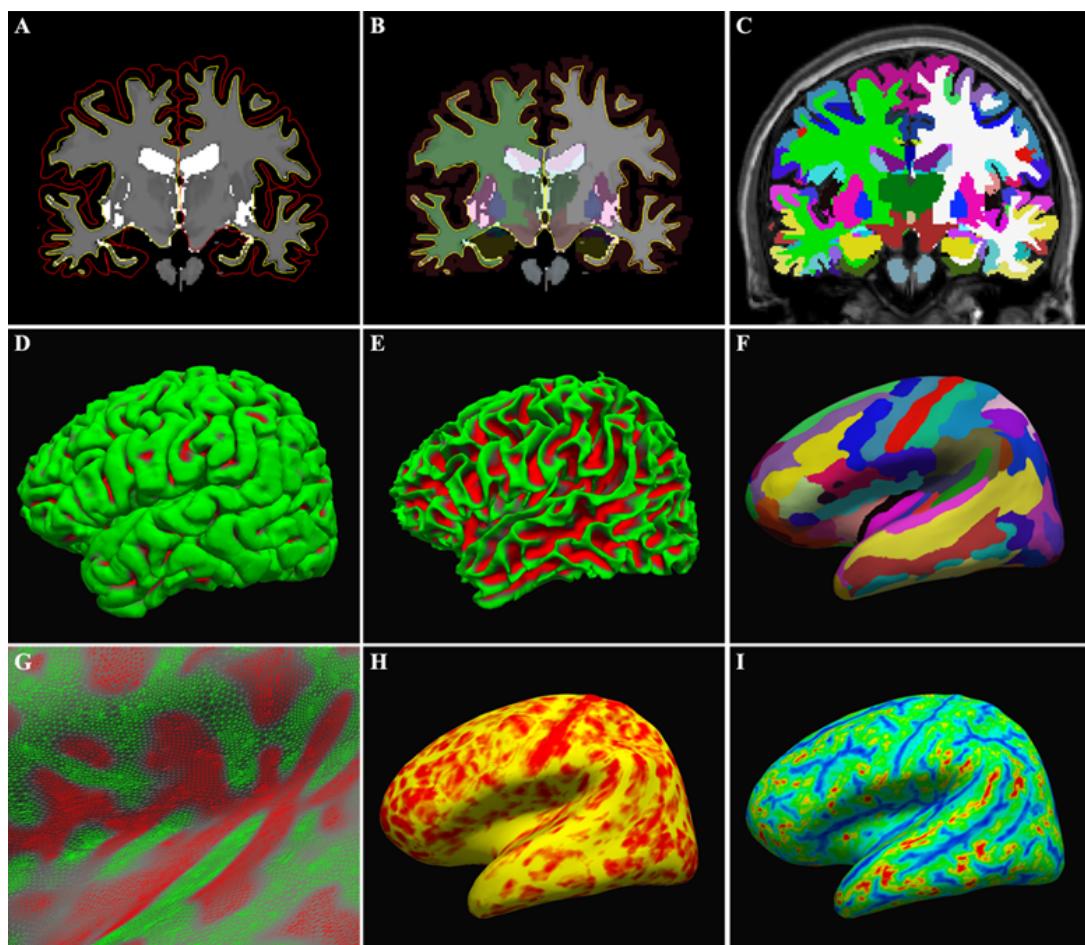
The core script in CCS for skull stripping is “[ccs\\_anat\\_01\\_pre\\_freesurfer.sh](#)”, the skull stripping method based on the transfer learning method to train the skull stripping model (deepbet). A well-trained model by CCS team is saved in the models folder. This model has been proved performed good of skull stripping in children and adults. Figure 1.3 highlights the brain after deepbet skull stripping in red .



**Figure 1.3 Automatic skull stripping results of deepbet**

### 1.3 Cortical Reconstruction

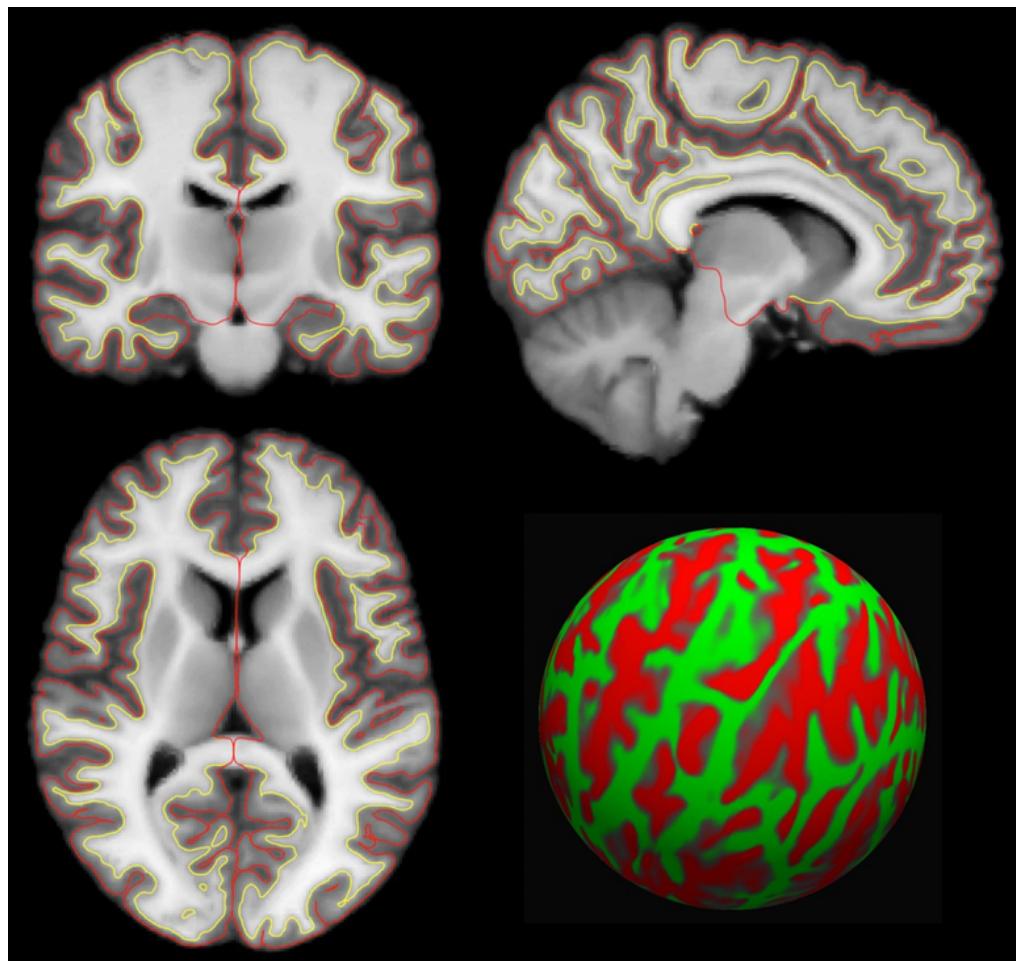
The core script in CCS for cortical reconstruction is “[ccs\\_anat\\_02\\_freesurfer.sh](#)”. The cortical reconstruction includes several steps. Firstly, the white matter segmentation is completed, and on this basis, the white matter surface (yellow curve in Figure 1.4 A) and pial surface (red curve in Figure 1.4 A) are constructed. The white matter surface is located under the cortex, and the difference between individuals is larger than that of pial surface, and the reconstruction quality is higher, so it will be used to align the sulcus and gyrus in the subsequent cortical reconstruction. Secondly, the subcortical and cortical (Figure 1.4B) areas are segmented respectively according to the sulcus distribution and gray characteristics. Finally, the 3D reconstruction of pial surface (Figure 1.4D) and white matter surface (Figure 1.4E) are completed by using triangular finite element method (green is the gyrus and red is the sulcus). Figure 1.4F shows the segmented image after blowing the surface of the pial flat. On the reconstructed cortical surface, each node of the triangle grid (Figure 1.4G) can be assigned with corresponding structural or functional attribute metrics. Figure 1.4H and Figure 1.4I show the morphological map of the cortical thickness and surface area.



**Figure 1.4 Each step of cortical reconstruction**

## 1.4 Registration

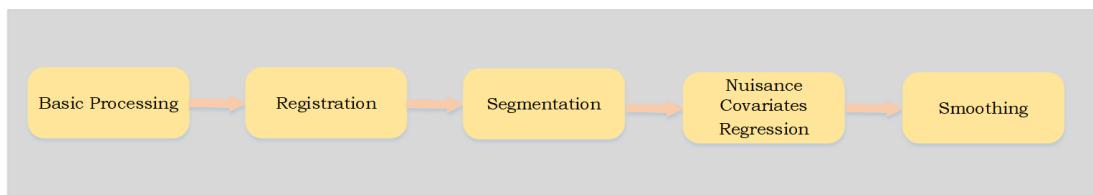
The purpose of registration is to match the anatomical positions of different individuals, so it is also called spatial standardization. The core scripts in CCS for registration is “[ccs\\_anat\\_03\\_postfs.sh](#)”. The former script is mainly used for registration based on 2D cortical surface, and 3D brain image, and the images are registered to the same standard space (Figure 1.5).



**Figure 1.5 MNI 152 standard space and its cortex properties**

# Chapter 2 Preprocessing of Resting-state Functional Magnetic Resonance Imaging

The resting functional Magnetic Resonance Imaging (rfMRI) is based on blood oxygen-dependent level (BOLD) signal, which could be affected by many confounding factors. Therefore, it is necessary to preprocess the raw image in order to correct the magnetic field instability, remove non-brain information, motion influence and image grayscale deviation between individuals that are produced in the process of image acquisition. It includes image registration, segmentation, nuisance covariates regression, and smoothing (Figure 2.1). Similar to preprocessing of sMRI, these steps are essential preparations for subsequent rfMRI analysis.



**Figure 2.1 Flow chart of rfMRI preprocessing**

## 2.1 Basic Processing

The core script in CCS for basic processing is “[ccs\\_01\\_funcpreproc.sh](#)”. The basic processing of rfMRI includes removing unstable points, despiking, slice timing, head motion correction, generating whole brain mask, and spatial-temporal standardization.

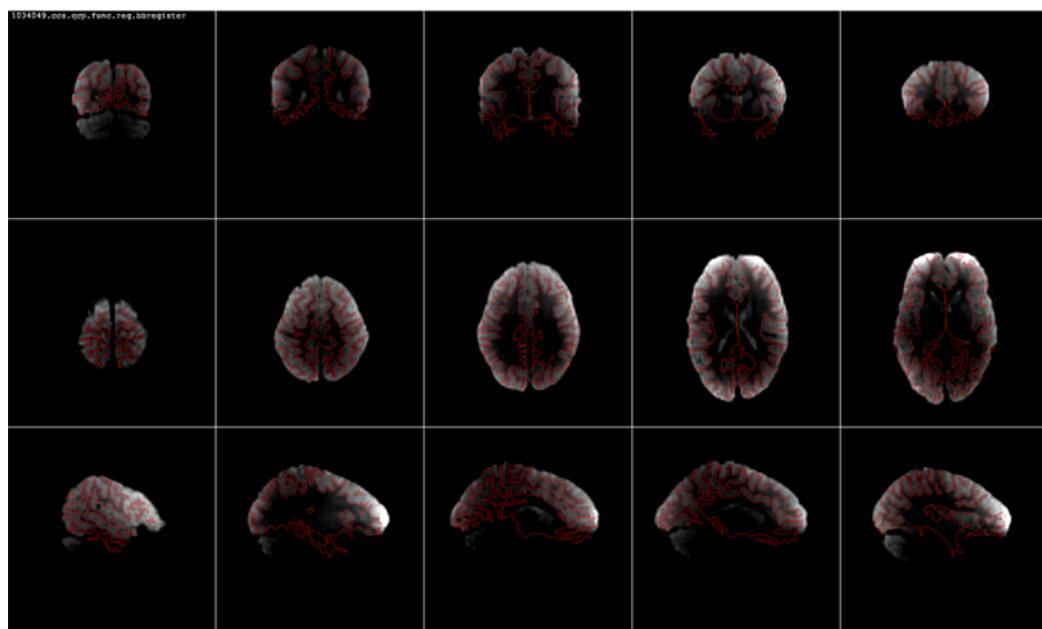
- At the beginning of scanning, the magnetization is unstable. To ensure the steady state of longitudinal magnetization, the first few time points of the image will be removed, which is generally for 10 seconds. ([AFNI: 3dcalc](#))
- In the 4D individual time series, there may be a variety of unexpected factors leading to outliers, which will affect the image analysis and need to be corrected. ([AFNI: 3dDespike](#))
- In order to align the time of each slice to the same time point, slice timing is needed. ([AFNI: 3dTshift](#))
- Head motion is inevitable in image acquisition, so it is necessary to correct the head motion based on Fourier transform. ([AFNI: 3dvolreg](#))
- With the help of the results of sMRI preprocessing that removed the non-brain tissue, non-brain voxels in rfMRI are deleted and then the whole brain mask of rfMRI could be created for the subsequent processing. ([AFNI: 3dAutomask](#))

- To enhance the comparability of signals among different individuals, spatial-temporal standardization is necessary. ([FSL: fsmaths](#))

## 2.2 Registration

The core script in CCS for rfMRI registration is “[ccs\\_02\\_funcbbregister.sh](#)”. CCS involves the calculation and analysis based on 2D cortical surface and 3D volume space, both require the registration of brain images of different individuals to the standard space for analysis. The same as the steps in the previous sMRI registration, the rfMRI image needs to be registered to the standard template. Registration of rfMRI includes three steps:

(1) The individual's rfMRI image are registered with the sMRI image. This step requires only 6-parameter rigid body transformation, so it is also called image alignment. To make full use of the information of high-resolution sMRI image, based on the WM-GM surface information of structural image, the boundary-based registration (BBR) developed by FreeSurfer team is adopted to estimate the 6-parameter rigid body registration matrix from the functional space to the structural space ([FreeSurfer: bbregister](#)) (Figure 2.2, red curve).



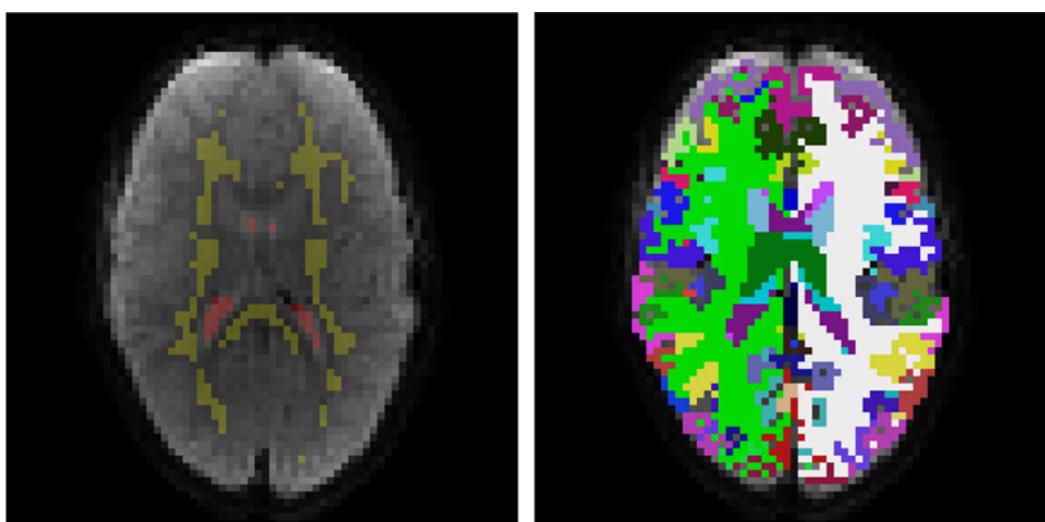
**Figure 2.2 Structural-Functional image alignment based on WM-GM boundary**

(2) Using affine linear registration ([FSL: flirt](#)) and spline based nonlinear registration ([FSL: flirt](#)) to estimate the transformation matrix of individual structure template to standard template. This step has finished in the preprocessing of sMRI.

(3) Calculating the transformation matrix of the rfMRI image registration to the standard space by combining the above two transformation matrices.

### 2.3 Segmentation

The core script in CCS for rfMRI segmentation is “[ccs\\_03\\_funcsegment.sh](#)”. The resolution and tissue contrast of rfMRI are relatively low, so the image segmentation mainly uses the results of the sMRI segmentation. Using the 6-parameter rigid body registration matrix created in 2.2(1) to transform the individual cerebrospinal fluid and white matter mask (Figure 2.3, left) and gray matter parcellation map (Figure 2.3, right) into the individual functional space.



**Figure 2.3 Functional image segmentation example**

### 2.4 Nuisance Covariates Regression

The core script in CCS for rfMRI nuisance covariates regression are “[ccs\\_04\\_funcnuisance.sh](#)” and “[ccs\\_04\\_funcAROMA.sh](#)”. In rfMRI data, a lot of noise will affect the signal, such as unstable magnetic field signal of scanner, rhythmic physiological activities such as breathing and heartbeat, and head motion during scanning. In order to minimize the influence of these noise, CCS calculates corresponding variates and removes them from the data at the individual level. Specifically, ICA-AROMA is be used to get rid of nuisance. In addition, CCS also takes the mean signal (or principal component signal) of white matter and cerebrospinal fluid as independent variables into the regression equation to remove the influence of these non-brain tissue signals. Whether to remove the global signal is still controversial, so both the data that with global signal and removed the global signal will be saved in CCS for users.

## **2.5 Smoothing**

The core scripts in CCS for rfMRI smoothing are “[ccs\\_05\\_funcpreproc\\_nofilt.sh](#)” and “[ccs\\_05\\_funcpreproc\\_cortex.sh](#)”. As the final step of preprocessing, smoothing at spatial and temporal level is needed in order to enhance the SNR. Time domain smoothing includes low-frequency bandpass (0.01-0.1 Hz) filtering and linear trend removal. In spatial domain, Gaussian smoothing is used. In the subsequent brain connectome analysis, some functional map index (such as low-frequency fluctuation amplitude) do not need bandpass filtering. At the same time, with the progress of spatio-temporal sampling technology, high-frequency signals also show its possible neuroscience significance. Therefore, CCS also keeps the results without filtering. In the spatial smoothing, the 3D volume based and 2D surface based rfMRI data are Gaussian smoothed respectively.

# Chapter 3 Preprocessing of Diffusion Tensor Imaging

Diffusion Tensor Imaging (DTI) is a non-invasive magnetic resonance imaging technique, which can be used to measure the characteristics of white matter fibers. This technology measures the weighted T2 images of human brain by changing the direction of gradient magnetic field, and then constructs the diffusion tensor of each position in the brain based on these images, so as to further construct the white matter fiber of brain. The algorithm mainly includes: basic processing, image registration and segmentation (Figure 3.1).

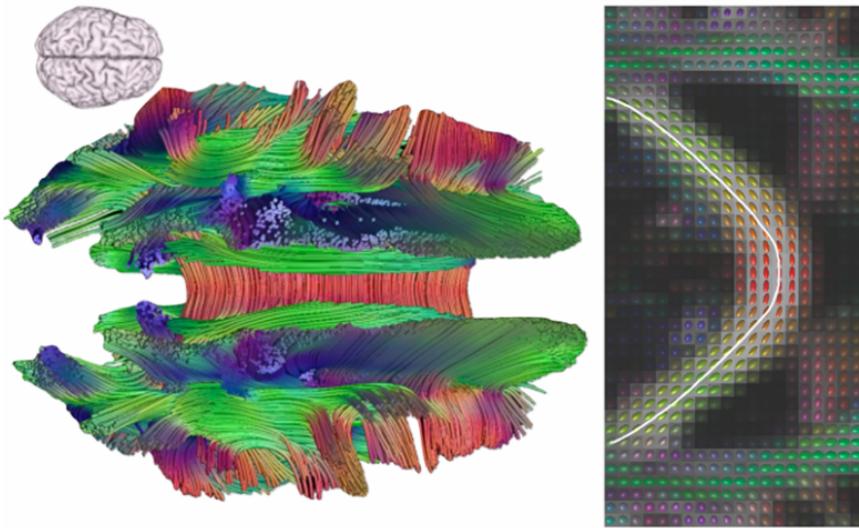


**Figure 3.1 Flow chart of DTI preprocessing**

## 3.1 Basic Processing

CCS mainly employs FDT toolkit from FSL (<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/fdt>) and DTK software (<http://www.trackvis.org>). The core script in CCS for DTI basic processing is “`ccs_01_dtipreproc.sh`”, including the following steps:

- 1) Eddy currents in gradient coil cause the expansion and shear of DTI. Distortion correction can correct these deformations and slight head motion by affine registration to a reference B0 image template.
- 2) Extracting B0 image and diffusion weighted image respectively.
- 3) Generating B0 brain image for subsequent analysis. Then it is registered with high-resolution T1 image to create mask.
- 4) Diffusion tensor fitting, that is, fitting a diffusion tensor model on each voxel. CCS provides two fitting algorithms: a. `FDT`, `dtifit`, b. `DTK`, `dti_recon`. Tensor fitting will create various white matter fiber parameter images, such as FA and MD map. Based on these results, the whole brain fibers could be reconstructed according to certain rules (Figure 3.2).

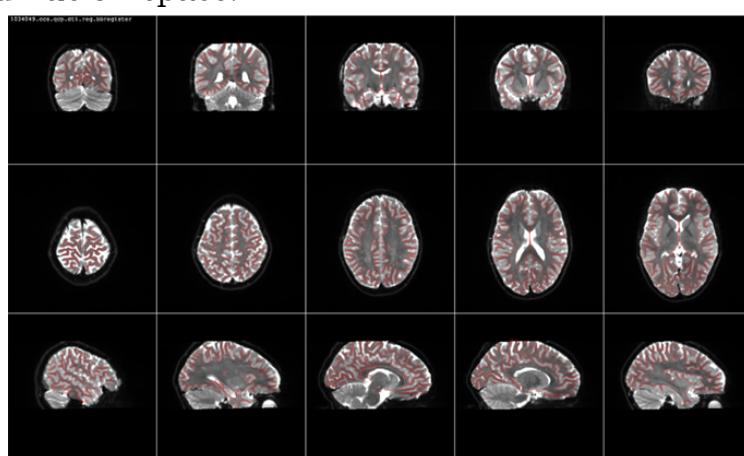


**Figure 3.2 DTI reconstruction of whole brain fiber**

### 3.2 Registration

The core script in CCS for DTI registration is “[ccs\\_02\\_dtibregister.sh](#)”. Fiber tracking and reconstruction need to be analyzed in different individual DTI space, so the information in standard space needs to be transformed into individual space (Figure 3.3). This includes three steps:

- (1) The B0 image of an individual is registered to the individual structural image. Similar to the functional image, BBR from FreeSurfer is adopted to estimate the 6-parameter rigid body registration matrix.
- (2) Estimating the transformation matrix for registration of individual space structural image to standard space. Here we can use the transformation information from the rfMRI registration.
- (3) Combining the above two matrices, we can calculate the transformation matrix that transforms the standard space into the individual diffusion space.



**Figure 3.3 Structural-Diffusion image alignment based on WM-GM boundary**

### 3.3 Segmentation

The core script in CCS for DTI segmentation is “[ccs\\_03\\_dtisegment.sh](#)”. The spatial resolution of DTI is low, so the diffusion image segmentation needs support from the results of the sMRI segmentation. Using the 6-parameter rigid body registration matrix created in 3.2, transform the individual gray matter parcellation map into the individual diffusion space (Figure 3.4). Based on this map, we can construct the white matter fiber structure connectome.

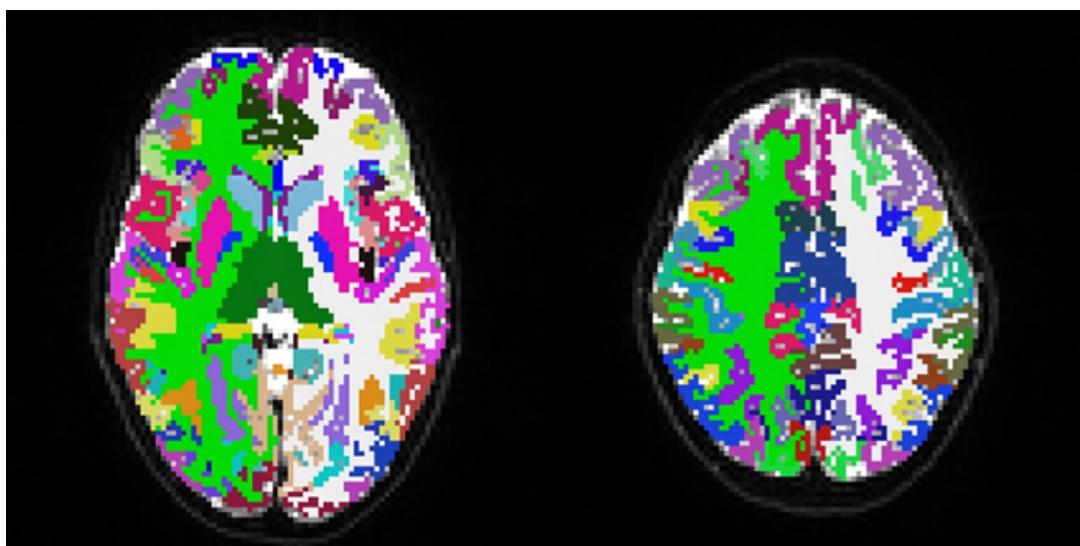
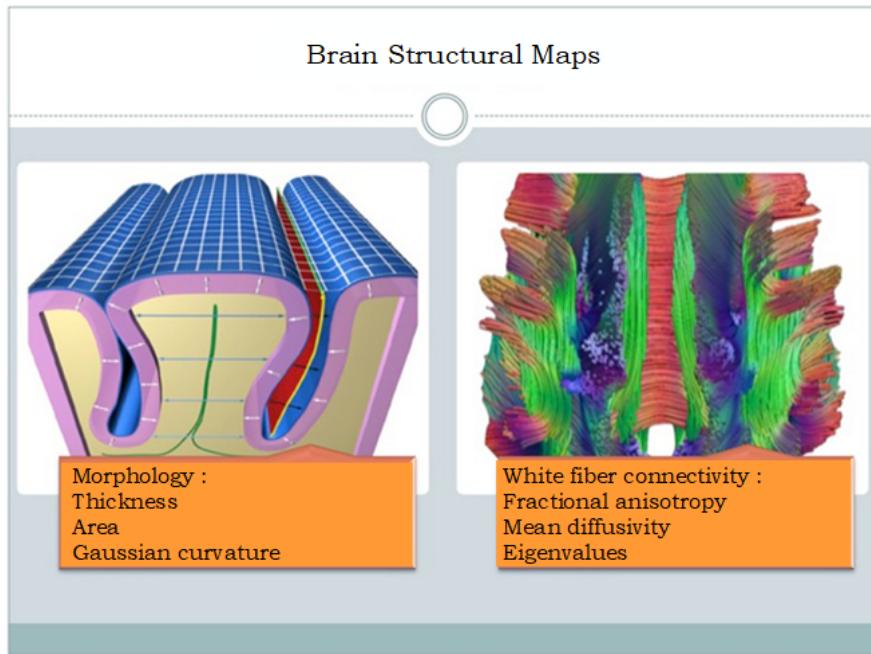


Figure 3.4 DTI segmentation example

## Chapter 4 Individual Structural Map

CCS provides different structural maps of individuals for subsequent statistical analysis at the individual or group level. Structural maps include morphology and white matter fiber connectivity, which correspond to sMRI T1 image and DTI processing results (Figure 4.1).

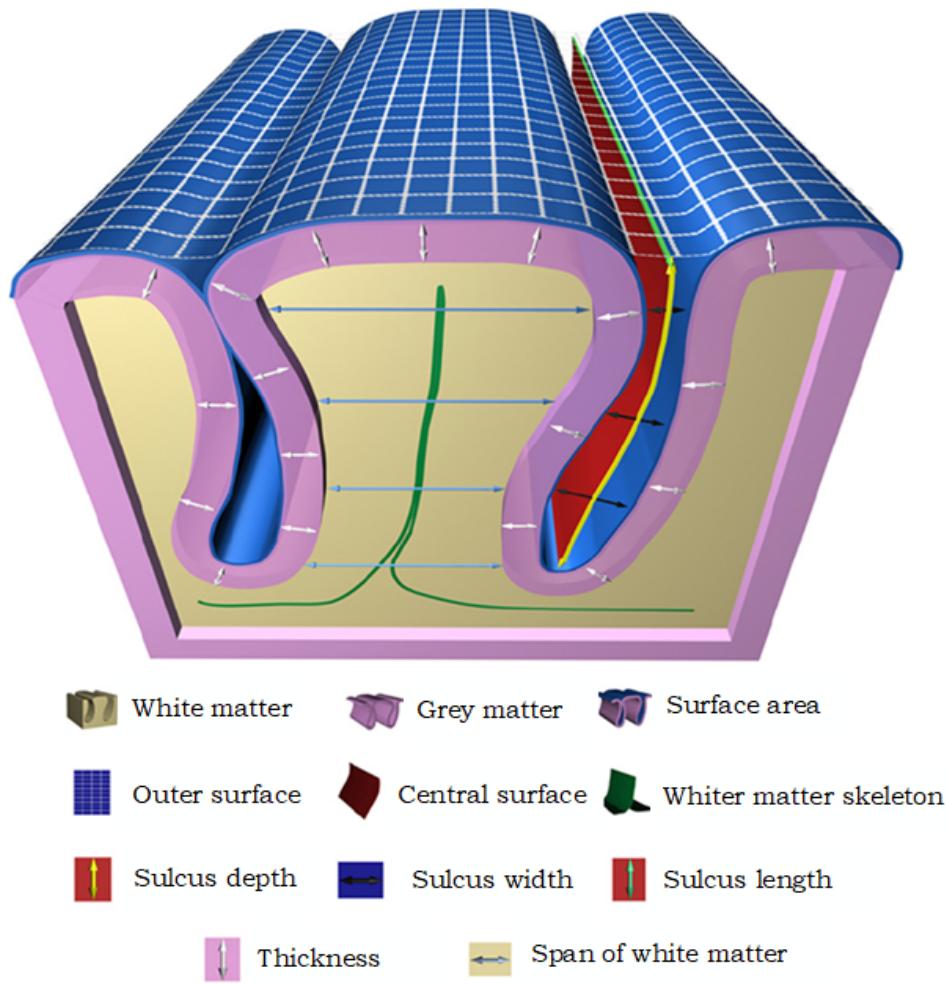


**Figure 4.1 Individual brain structural maps**

### 4.1 Morphology

CCS creates the morphological map of each individual after the cortical reconstruction. The most commonly used morphological indexes include cortical thickness, surface area, mean curvature, sulcus depth and volume. All the data will be saved in surf folder of individual directory. Figure 4.2 shows the meaning of each index.

- Thickness: the cortex has two surfaces, the gray-white matter interface (inner surface) and the gray-meningeal interface (outer surface), and cortical thickness is the distance between the two surfaces.
- Mean /Gaussian curvature: the curvature of curved surface. The reciprocal of the radius of the inscribed sphere in two regular directions is its principal curvature, the mean curvature is the average of the two curvatures, and the Gaussian curvature is the product of the two principal curvatures.
- Sulcus depth: point multiplication of displacement which produced during cortical expansion and regular unit vector, reflects the large-scale geometric information of cortical.
- Cortical volume: the product of thickness and surface area.



**Figure 4.2 The morphology of cerebral cortex**

#### 4.2 Connectivity of Fiber Tracts

CCS creates the DTI map of each individual after the DTI preprocessing, including fractional anisotropy (FA), mean diffusivity (MD), eigenvector (L1), and eigenvalue.

## Chapter 5 Individual Functional Map (3D)

At present, the rfMRI signals are analyzed mainly from the perspectives of functional differentiation and integration (Figure 5.1). Functional differentiation explores the temporal dynamic properties of large-scale human brain processing units, such as the fluctuation intensity and complexity of resting low-frequency signals, which is the property of a single time series. Functional integration refers to the relationship between brain processing units, such as examining the intensity of functional synchronization in local brain regions, examining the functional connectivity between two brain regions or between multiple brain regions, the functional connectivity between the left and right hemispheres, and the connectivity strength of the whole brain large-scale functional connectivity group, etc. This chapter introduces several voxel-based function maps in 3D space.

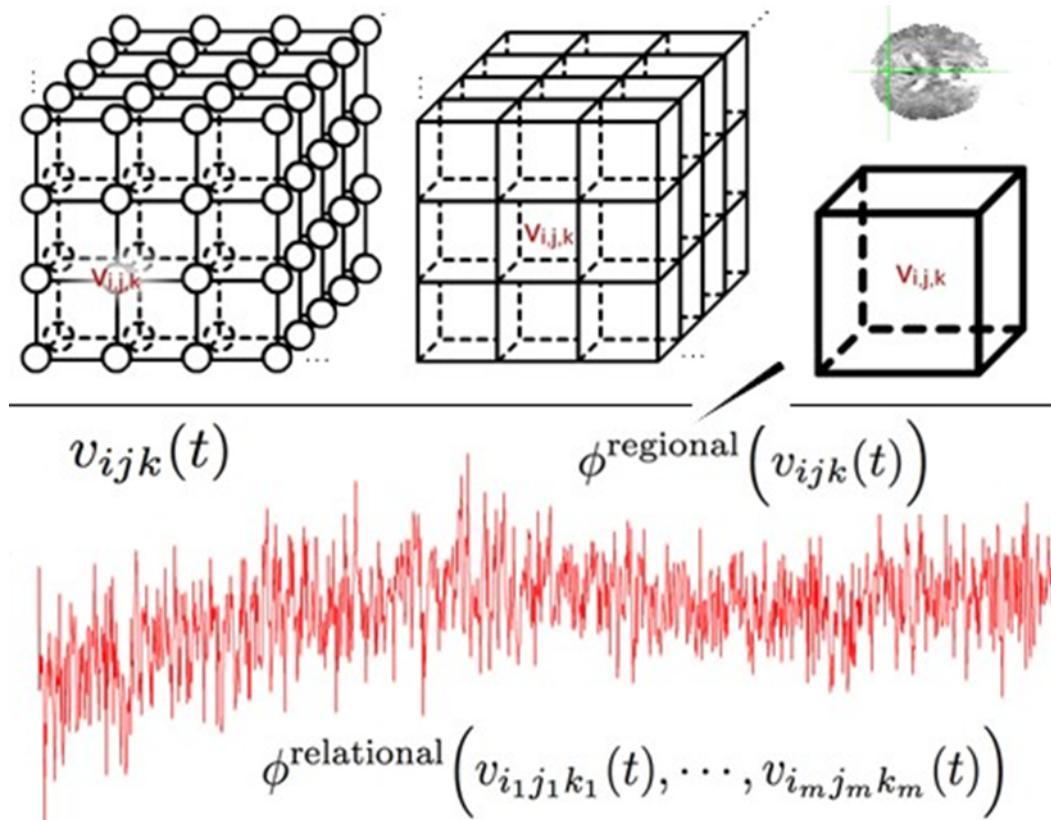


Figure 5.1 The calculation of individual functional map (3D)

### 5.1 Amplitude of Low-Frequency Fluctuation

Amplitude of low-frequency fluctuation (ALFF) assumes that resting state BOLD signal has its physiological significance in the low frequency range. The average amplitude of all frequency points in a low frequency band (0.01-0.1 Hz) is used to describe the intensity of a voxel's spontaneous activity, which reflects the level of each voxel's

spontaneous activity in the resting state from the perspective of energy. Fractional ALFF (fALFF) refers to the ratio of power spectrum of low-frequency range to that of the entire frequency range. FALFF is an improved algorithm based on ALFF. Previous studies have proved that fALFF can effectively inhibit the non-specific signal components in the cistern areas compared with ALFF, reduce the interference of physiological noise, and improve the sensitivity and specificity of detecting brain spontaneous activity. These two functional maps exhibit good test-retest reliability. With the increasing attention to the low-frequency fluctuation amplitudes in different frequency bands, they have been widely used in the research of various diseases.

The core script in CCS for the calculation of ALFF is “[ccs\\_06\\_singlesubjectALFF.sh](#)”. It can be used to calculate the ALFF and fALFF of low-frequency band (0.01-0.1 Hz) and slow4 band.

## 5.2 Regional Homogeneity

Regional homogeneity (ReHo) was proposed by Zang et al. They assumed that the BOLD signal of neighboring voxels are similar under certain conditions. In this index, Kendall coefficient is used to measure the consistency of time series between one voxel and several neighboring voxels in 3D space (6, 18 or 26 voxels). The range of ReHo is from 0 to 1, the higher the value, the better synchronization of these time series. It has been proved to have a high test-retest reliability.

The core script in CCS for the calculation of ReHo is “[ccs\\_06\\_singlesubjectReHo.sh](#)”. It can be used to calculate the functional consistency at voxel level in 3D space.

## 5.3 Seed-based Functional Connectivity

Functional connectivity is defined as temporal correlations between two brain regions. Seed-based functional connectivity (sFC) is a simple and widely used method to calculate the resting state functional connectivity, the basic steps of the method are as follows: (1) Determine the specific brain area as the seed region according to the study purpose, to extract the time series of voxels in the region and calculate the average time series of it. (2) Calculate the correlation coefficient between the mean time series of seed and the whole brain then transform it to the Fisher-z value as the connectivity strength. There are several ways to determine seed: (1) Activation map of related tasks. (2) Prior anatomy knowledge. (3) Standard brain map. The core script in CCS for the calculation of sFC is “[ccs\\_06\\_singlesubjectSFC.sh](#)”. Users can calculate sFC by providing their own seed mask.

## **5.4 Interhemispheric Functional Connectivity**

Voxel-mirrored homotopic connectivity (VMHC) refers to the functional connectivity between hemispheres on a symmetrical brain template. In other words, it quantifies the resting state functional connectivity between each voxel in one hemisphere and the mirror voxel in the opposite hemisphere. Before calculating the VMHC, we first create a standard symmetrical brain template, then register preprocessed functional image to the symmetrical standard template, thus obtaining the mirror symmetry points of each voxel. Finally, calculate the Pearson correlation coefficient of time series on each pair of mirror symmetry voxels and converted into fisher-z value.

The core script in CCS for the calculation of VMHC is

[`"ccs\_06\_singlesubjectVMHC.sh"`](#). The script

[`"ccs\_06\_singlesubjectVMHC-SFC.sh"`](#) can be used to calculate the functional connectivity map between whole brain and the significant region of VMHC.

## **5.5 Independent Component Analysis**

Independent component analysis (ICA) is another common method for rfMRI analysis. CCS provides two special ICA methods. (1)

DualRegression method. In this method, the pre-specified spatial component map and multiple linear regression model are used to construct the corresponding functional connectivity network map of individuals, which shows high test-retest reliability and repeatability. (2) ICA that based on the individual component map mining algorithm (gRAICAR). This algorithm decomposes the data into multiple components without making assumptions about the pattern of brain activity, and each component represents a different network of brain activity. This decomposition is carried out several times, during which the replicability level of each independent component is calculated. Then, these components are sorted and averaged to obtain more stable results according to replicability. Compared with the commonly used methods, this algorithm has obvious advantages: (1) It does not need to assume the working mode of the brain, which is beneficial to the discovery of new brain functional networks. (2) ICA is a multivariate statistical method, which can detect multiple components at the same time, and each component can represent different brain activity characteristics. (3) The influence of noise can be minimized by the reliable components. (4) Sorting the components, which reflects the relative strength of different components to a certain extent, thus providing new information. (5) Getting more reliable results.

The core script in CCS for the calculation of dual regression ICA is [`"ccs\_06\_singlesubjectICA.sh"`](#). Now, gRAICAR is based on MATLAB,

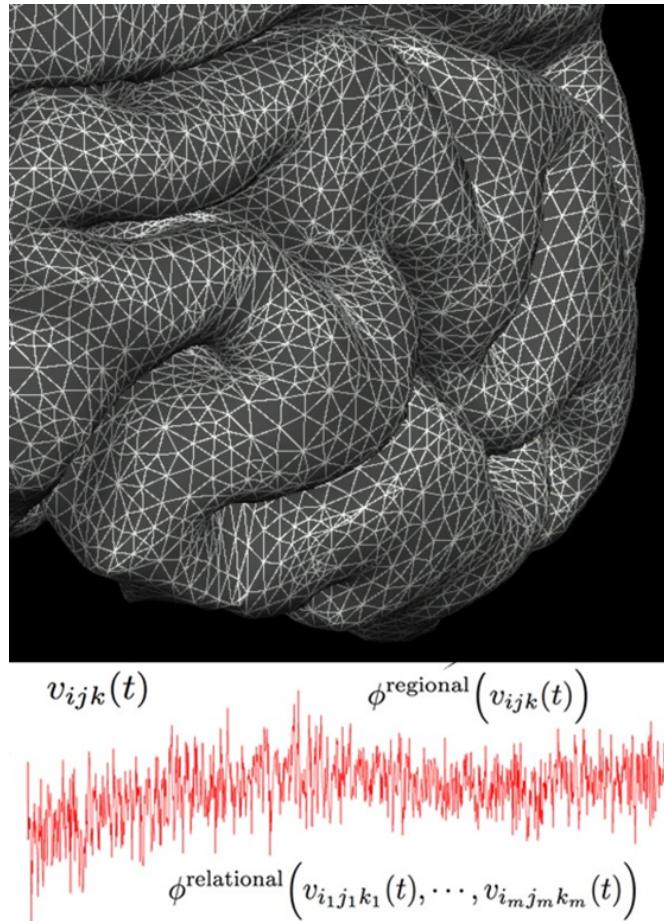
and the script of this algorithm will be provided in the future CCS version.

## **5.6 Network Centrality**

Centrality is an important indicator of network analysis, which is used to measure the importance of “elements” in the network. The “elements” here refer to nodes, edges, communities and the whole network. CCS provides several methods to calculate network centrality, measures the connectivity strength within the whole brain functional connectome from different perspectives. For example, degree centrality (DC) reflects the number and intensity of direct connectivity with the node. Eigenvector centrality (EC) reflects the strength of connectivity with the nodes from the perspective of global features. Subgraph centrality (SC) measures the connectivity strength between node and subgraph of the network. Betweenness centrality (BC) reflects the importance of the nodes in the connectivity of the whole brain network. The calculation of centrality can be divided into binary centrality and weighted centrality. The adjacency matrix of binary centrality use 1/0 to indicate whether there are connections in the brain, and a threshold should be set as the standard to determine whether connections existed. The adjacency matrix of weighted centrality is represented by the strength of functional connectivity. The core script for the calculation of voxel-based centrality in CCS is “[ccs\\_06\\_singlesubjectVNCM.m](#)”.

## Chapter 6 Individual Functional Map (2D)

CCS extends the functional map mentioned in chapter 5 to the cortical surface(Figure 6.1), thus to give full play to the 2D characteristics of functional tissues of the human brain cortex, minimize the partial volume effect and avoid the mixing of different tissues' functional signals (such as GM-WM signal and GW- CSF signal). At the same time, the cortical surface coordinate system can also provide the basis for multimodal image integration analysis.

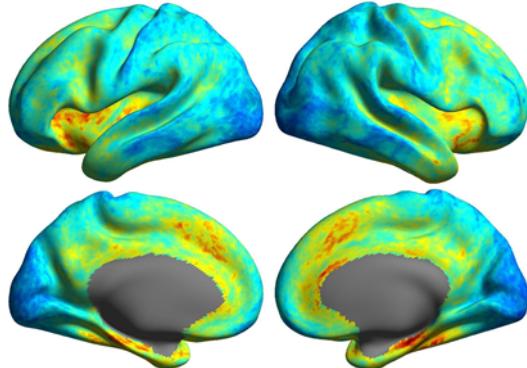


**Figure 6.1 The calculation of individual functional map (2D)**

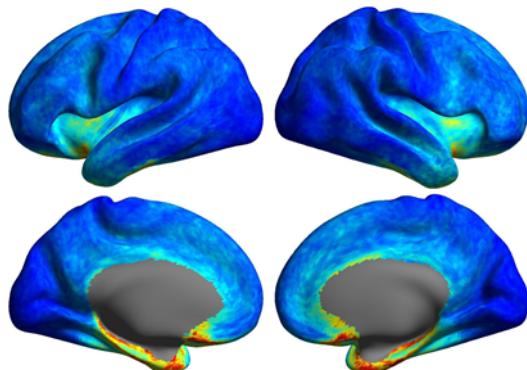
One of the key steps in the calculation of cortical surface functional map is how to project the 3D time series to the 2D cortical surface. In CCS, the preprocessed time series (non-smooth) in individual space are first projected onto the standard cortical grid with the resolution of 1 mm (fsaverage), and then down-sampled to the standard grid with the resolution of 4 mm (fsaverage5). All the following function maps are calculated based on this standard grid, and the calculation principle is consistent with the 3D function map. All 2D functional map are calculated based on MATLAB now, and the OCTAVE version will be developed in the future.

## 6.1 Amplitude of Low-Frequency Fluctuation

The core script for the calculation of 2D ALFF in CCS is “[ccs\\_06\\_singlesubject2dALFF.m](#)”. This script can be used to calculate the ALFF and fALFF of each vertex.



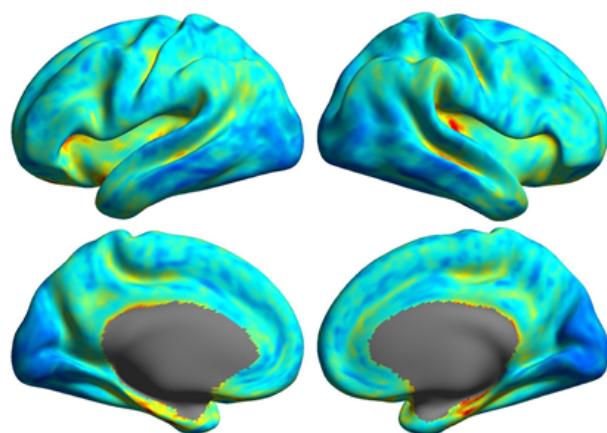
**Figure 6.2 Functional map of 2D ALFF**



**Figure 6.3 Functional map of 2D fALFF**

## 6.2 Regional Homogeneity

The core script for the calculation of 2D ReHo in CCS is “[ccs\\_06\\_singlesubject2dReHo.m](#)”. This script can be used to calculate the Kendall’s Concordance Coefficient of time series of each vertex and that of its neighboring 6 or 19 vertexes.



**Figure 6.3 Functional map of 2D ReHo**

### 6.3 Seed-based Functional Connectivity

The core script for the calculation of 2D sFC in CCS is “[ccs\\_06\\_singlesubject2dSFC.m](#)”. This script can be used to calculate the time series correlation coefficient between each vertex and all other vertexes.

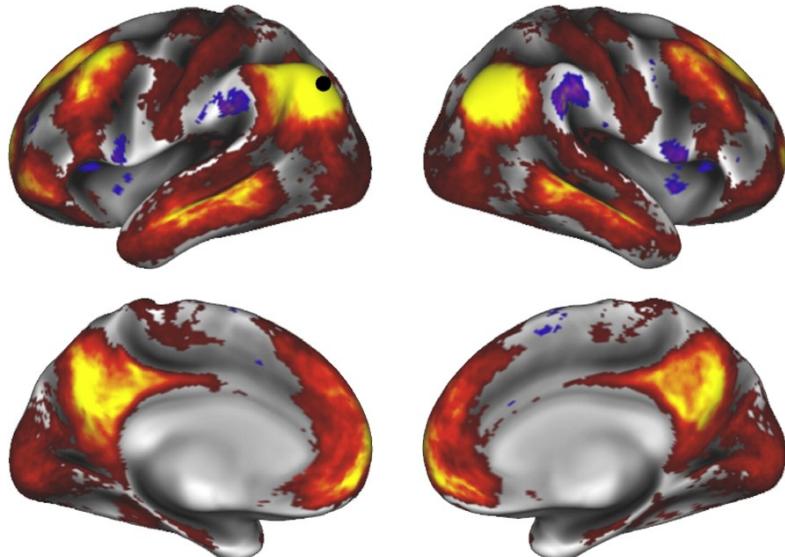


Figure 6.4 Functional map of 2D sFC

### 6.4 Interhemispheric Functional Connectivity

The calculation of 2D VMHC in CCS is not available yet, and it will be developed the next version.

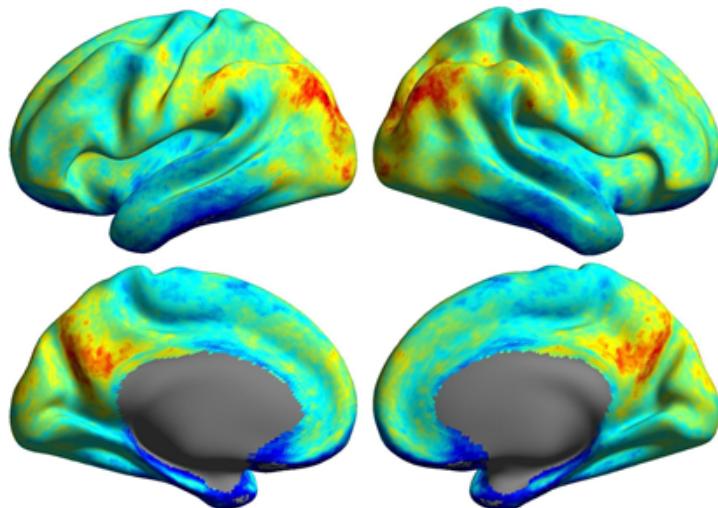


Figure 6.5 Functional map of 2D VMHC

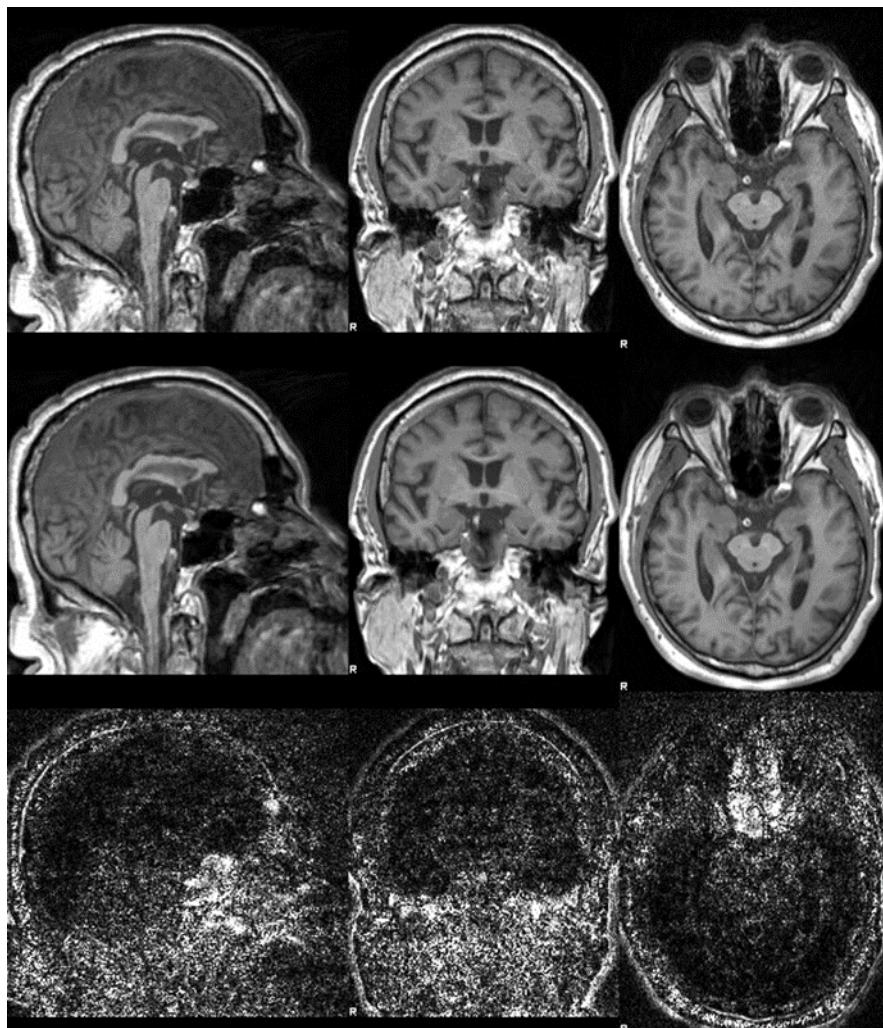
### 6.5 Independent Component Analysis

The calculation of 2D ICA in CCS is not available yet, and it will be developed the next version.

### 6.6 Network Centrality

## Chapter 7 Quality Control

In the research of MRI, the quality and individual difference of the images determine the effect of batch processing. Quality Control (QC) is one of the most important steps, which directly relates to the number of subjects in group analysis and corresponding statistical results. In the past, different QC strategies and standards were adopted, and CCS provides a very detailed QC module, which mainly includes skull stripping, cortical reconstruction, registration, segmentation, and head motion.



**Figure 7.1 QC of sMRI denoising**

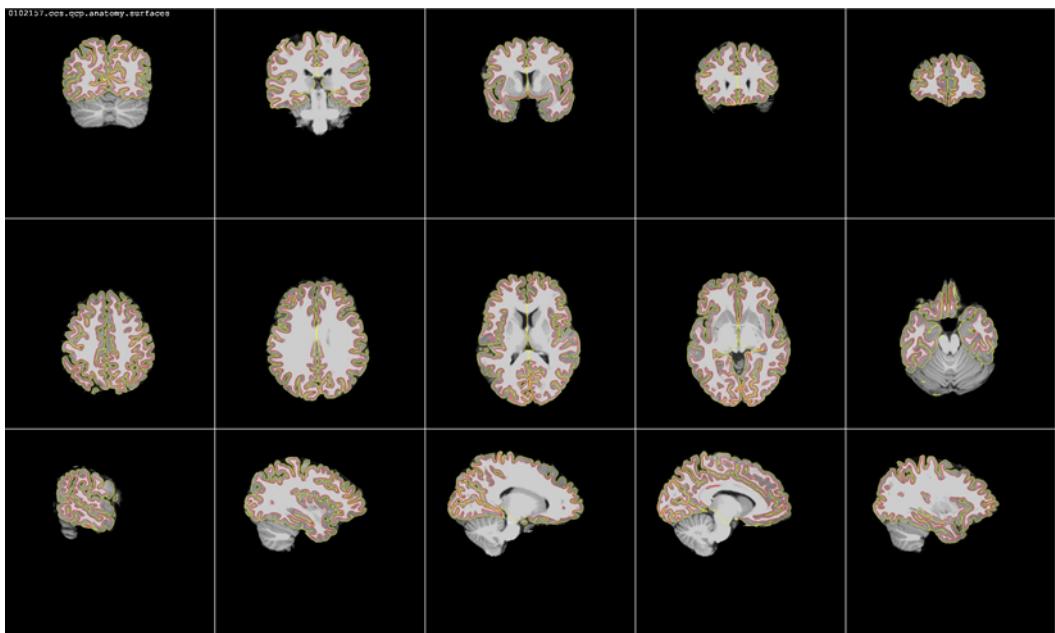
### 7.1 Denoising

All images have more or less nuisance noise. If the noise looks obvious, the data will be rejected directly for the analysis. Noise to be considered includes whether the head motion causes obvious artifacts during the scanning, whether there are organic changes in the image, etc. This can be determined by directly observing the original image.

CCS first denoises images based on non-local means, and creates pictures for visual inspection (Figure 7.1). Figure 7.1 shows the original image in the first row, the image after denoising in the second row, and the noise image in the last row. This figure also helps users to check motion artifacts and other image problems.

## 7.2 Cortical Reconstruction

Script “[ccs\\_01\\_anatcheck\\_surf.sh](#)” can be used for the creation of GM-WM boundary and pial curve (Figure 7.2). Script “[ccs\\_01\\_anatcheck\\_render.sh](#)” can be used for the creation of 3D GM-WM interface (Figure 7.3).



**Figure 7.2 The reconstruction of GM-WM boundary and pial curve**

## 7.3 Registration

Script “[ccs\\_02\\_funcbbregister.sh](#)” can be used for the checking of BBR registration (rfMRI to sMRI). Script “[ccs\\_02\\_dticheck\\_bbregister.sh](#)” can be used for the checking of BBR registration (DTI to sMRI). For 3D nonlinear registration, the script “[ccs\\_01\\_anatcheck\\_fnirt.sh](#)” can be used to create the FNIRT spatial normalization results of sMRI (Figure 7.4) and “[ccs\\_02\\_funccheck\\_fnirt.sh](#)” can be used to create the FNIRT spatial normalization results of rfMRI (Figure 7.5).

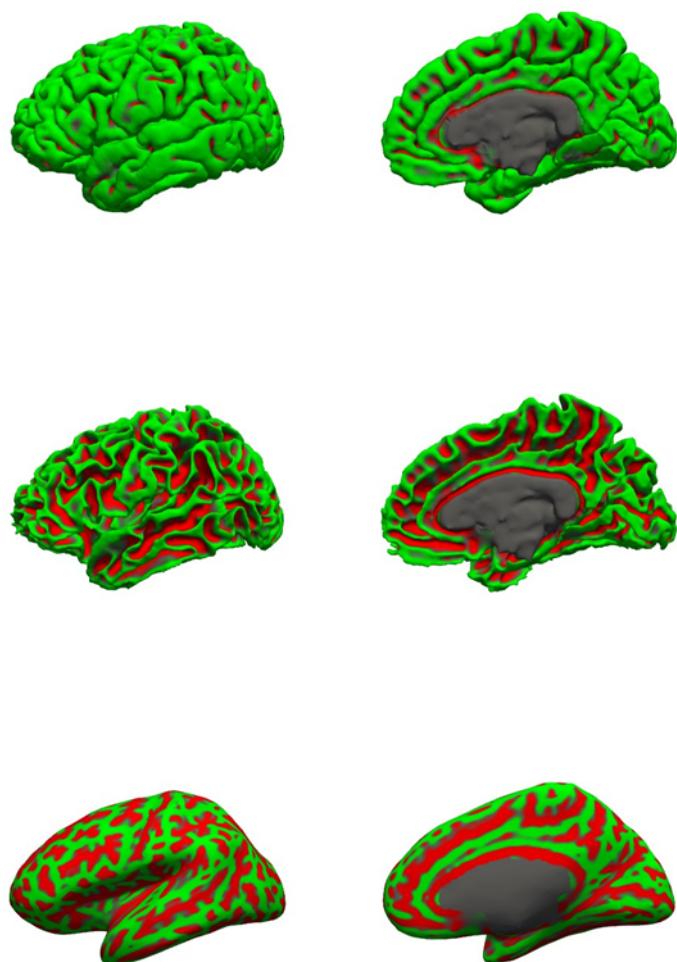
## 7.4 Segmentation

Script “[ccs\\_01\\_anatcheck\\_vol.sh](#)” can be used for the checking of sMRI segmentation (Figure 7.6).

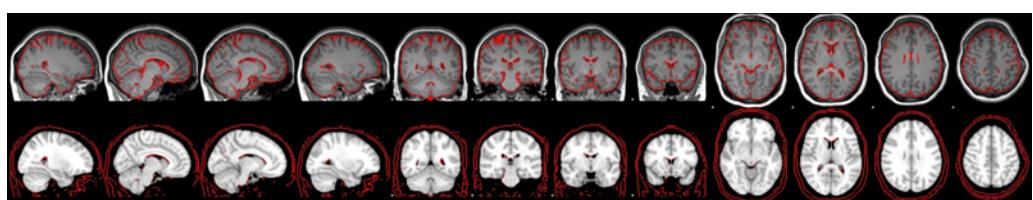
## 7.5 Head Motion

Head motion is a common confounding factor in MRI. The sMRI image that exhibit obvious motion artifact can be excluded in the first step of QC. This section mainly discusses the QC of the head motion of the rfMRI. CCS can not only preprocess and postprocess the sMRI and rfMRI image, but also creat the QC documents for registration error, Jacobian matrix and head motion. Using FD theory proposed by Power et al., script “[ccs\\_06\\_singlesubjectQCP.m](#)” calculates different motion values of functional image and creates the real-time motion monitoring picture (Figure 7.7) for subsequent QC.

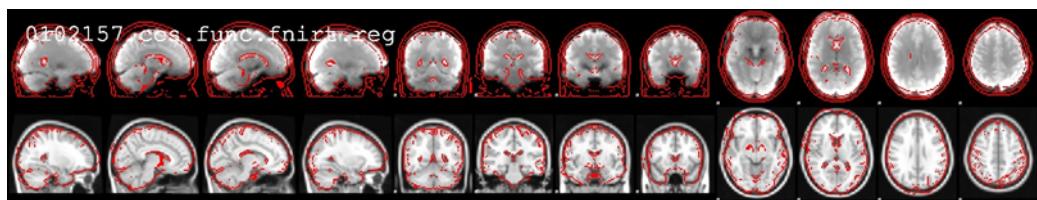
`0102157.ccs.qcp.anatomy.renders`



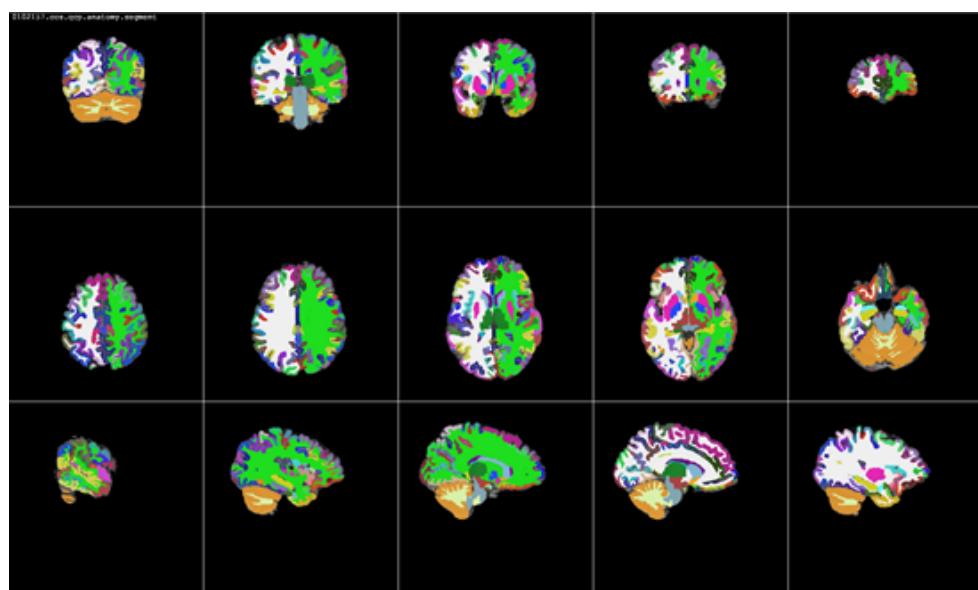
**Figure 7.3 3D GM-WM interface**



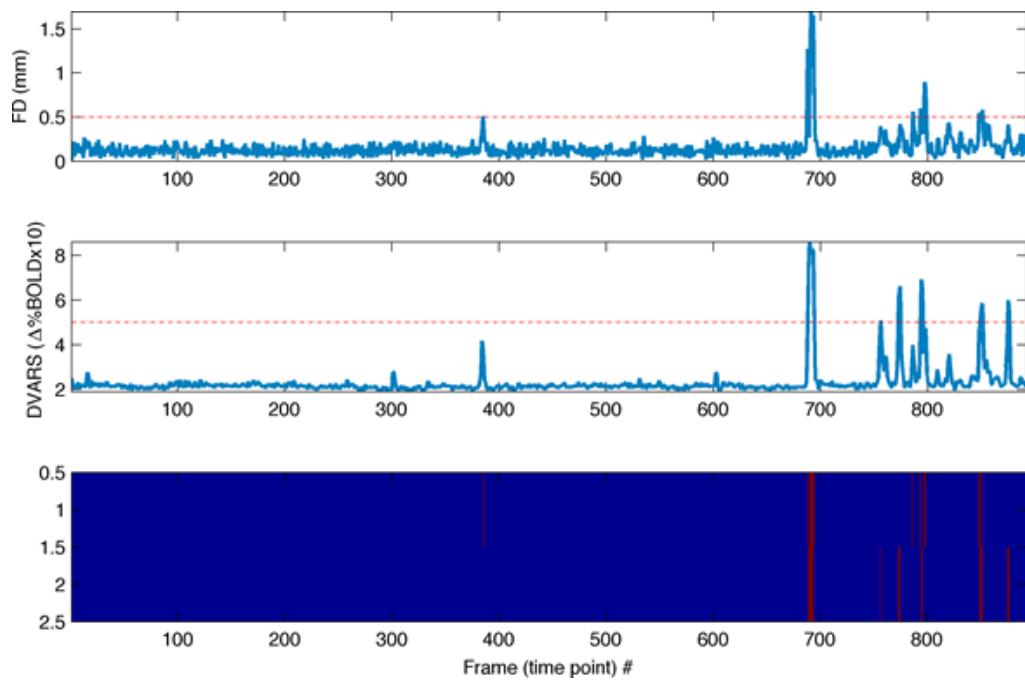
**Figure 7.4 QC of 3D sMRI nonlinear spatial normalization**



**Figure 7.5 QC of 3D rfMRI nonlinear spatial normalization**



**Figure 7.6 QC of sMRI Segmentation**



**Figure 7.7 QC of functional image motion**

## Chapter 8 Test-Retest Reliability

In the fields of society, behavior, physics, biology and medicine, there are various mixed factors affecting the actual measurement. It is important to establish and choose reliable and accurate measurement methods in practical application. Test-retest reliability is a statistical concept, which refers to the consistency of multiple measurements when an indicator is tested twice or more (retest). When it comes to clinical measurement (such as blood pressure), it refers to the change of intra-individual difference (the difference between two blood pressure measurements of the same person) relative to inter-individual difference (the difference between blood pressure measurements of different people). Various clinical diagnosis and the study of brain development need indicators and calculation methods with high test-retest reliability as small intra individual differences indicate the time stability of indicators and large inter individual differences indicate that it is easy to distinguish different individuals, which is beneficial to clinical auxiliary diagnosis. Previous studies have shown that rfMRI measurement and calculation methods are affected by various confounding factors (machine noise, thermal noise, fluctuation of non-brain neural activity, changes in external environment, lack of cognitive control, and standard of preprocessing). Therefore, the test-retest reliability of rfMRI calculation methods and indicators needs to be systematically studied in order to provide reference for practical application.

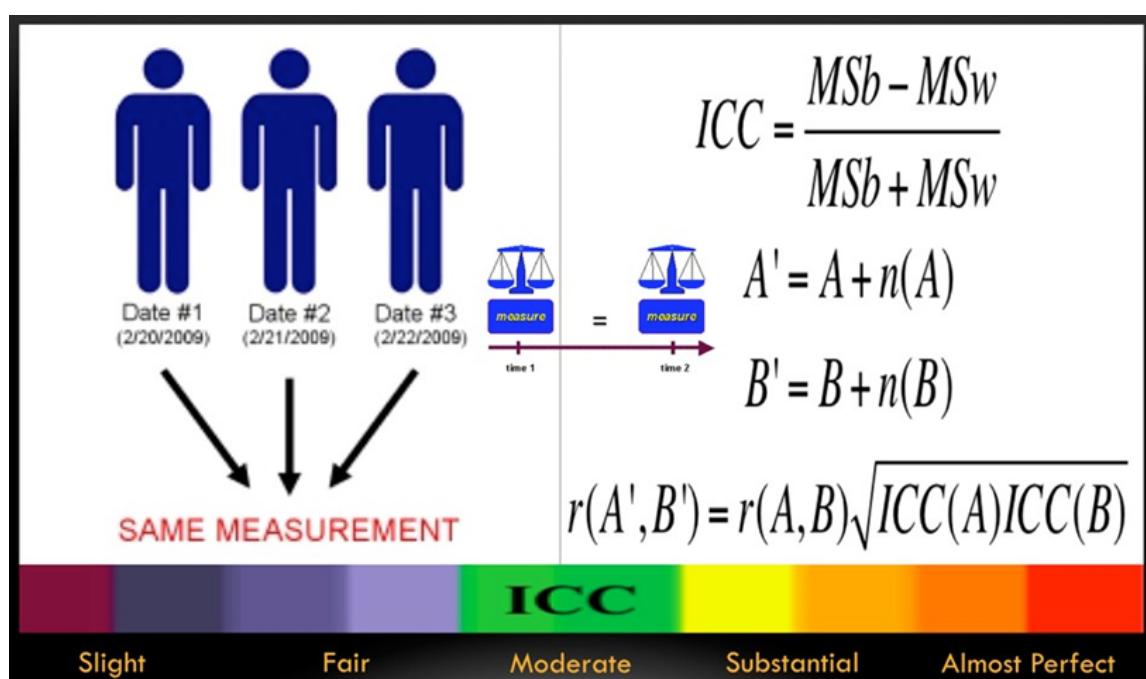


Figure 8.1 The concept and calculation formula of test-retest reliability

## **8.1 Test-Retest Reliability**

The intraclass correlation coefficient (ICC) can be used to calculate and quantify the test-retest reliability of different MRI measurements. In addition to a simple toolkit (IPN tools for Test-Retest Reliability Analysis:

<http://www.mathworks.com/matlabcentral/fileexchange/authors/21204>), CCS also includes a method that based on mixed linear model to avoid negative values.

## **8.2 Reproducibility**

The sample size in the functional neuroimaging research currently is mostly between 10 and 100, and such a sample size limited the study of the development of brain functional connectome and its relationship with behavioral performance. The sample size of a single site is obviously constrained by economic factors. It is very important to provide a data sharing platform with multi centers. In multi-center research, the reproducibility of research results is a great challenge. CCS provides the reproducibility of different indicators, and can be used to quantify the consistency and variation of different samples' results.

## **Chapter 9 Connectome-Wide Association Module**

The aim of CCS connectome-wide association module is to establish the association between the structural and functional connectome of brain, as well as the association between the connectome and the behavioral and clinical symptoms. The association model of CCS include classical linear statistical model, data mining model, machine learning model, and multimodal image integration model.

### **9.1 Linear Model and Statistical Analysis**

The linear regression model is used to establish the association between the connectome and behavior, clinical symptoms. The classical linear model is as follows:

$$Y = b_1X_1 + b_2X_2 + b_3X_3 + \dots + e$$

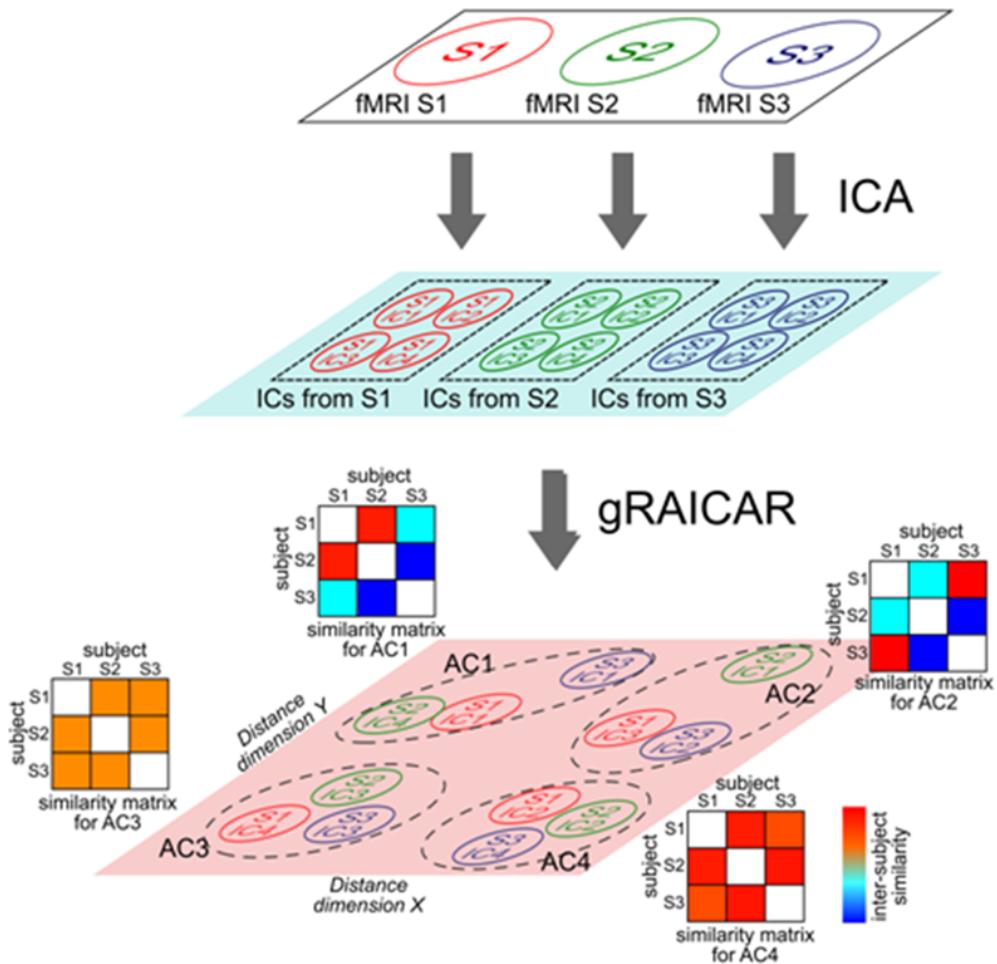
$Y$  is the column vector, which represents the connectome of each subject (such as the ReHo of a voxel or the centrality of an area).  $X_1$ ,  $X_2$  and  $X_3$  are the feature vectors of behavior or clinical symptoms. This model estimates the weights  $b_1$ ,  $b_2$  and  $b_3$  of  $X_1$ ,  $X_2$  and  $X_3$ , and tries to explain the connectome  $Y$  with the maximum variation of  $X_1$ ,  $X_2$  and  $X_3$  among the subjects. The contribution of  $X_1$ ,  $X_2$  and  $X_3$  and the differences among them are tested statistically. Linear model in CCS employs the FEAT from FSL and 3dRegAna from AFNI or the glmstats from FreeSurfer.

### **9.2 Connectome-wide Association Mining and Unsupervised Machine Learning**

Different from the classical linear statistical model mentioned above, the connectome-wide association mining model extracts brain functional network with obvious individual variation by mining neural image data, and then establishes a new association between the network and behavior or mind. The advantage of this model is that it does not depend on prior assumptions, and it provides assumptions for further testing.

This model adopts gRAICAR algorithm (<https://github.com/yangzhipsy/gRAICAR>). As shown in the Figure 9.1, gRAICAR classifies brain activity components (brain functional network) from multiple individuals into several categories (unsupervised learning) according to their similarity. The components in each category are as similar as possible, and it is required that they come from different individuals. Similarity matrix is used to describe the similarity among the representative components of each individual, which is called inter individual similarity matrix. These inter individual similarity matrices reflect the variation (or consistency) of different brain functional

networks among individuals. Therefore, gRAICAR can reflect inter individual variation of brain functional networks quantitatively.



**Figure 9.1 Schematic diagram of gRAICAR algorithm**

### 9.3 Supervised Machine Learning Model

The supervised machine learning model is helpful to establish large-scale or non-linear correlation between connectome and behavior. This method makes it possible to predict the behavior characteristics accurately. The supervised machine learning model in CCS is shown in Figure 9.2. This model establishes the relationship between the multiple connectome (such as the functional connectivity between multiple brain regions) and behavior measurement. In this model, “leave one out cross-validation” is carried out among the subjects,  $N - 1$  of the  $N$  subjects is used as the training set to estimate the mathematical model, and the remaining one is used as the test set to evaluate the generalization of this model. In the training set of each cross-validation, the subjects are further divided into internal training set and internal test set, so as to select effective features from many connectome features.

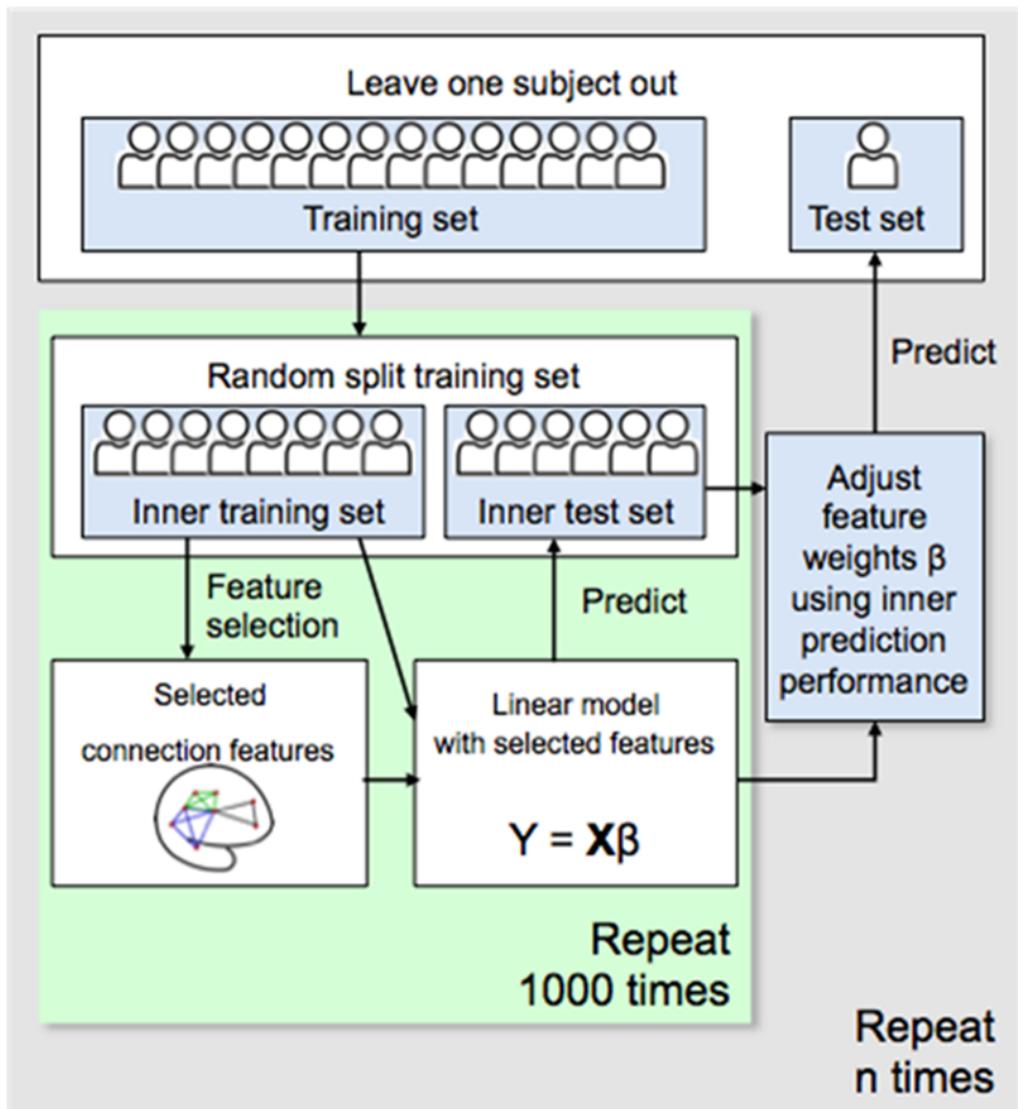
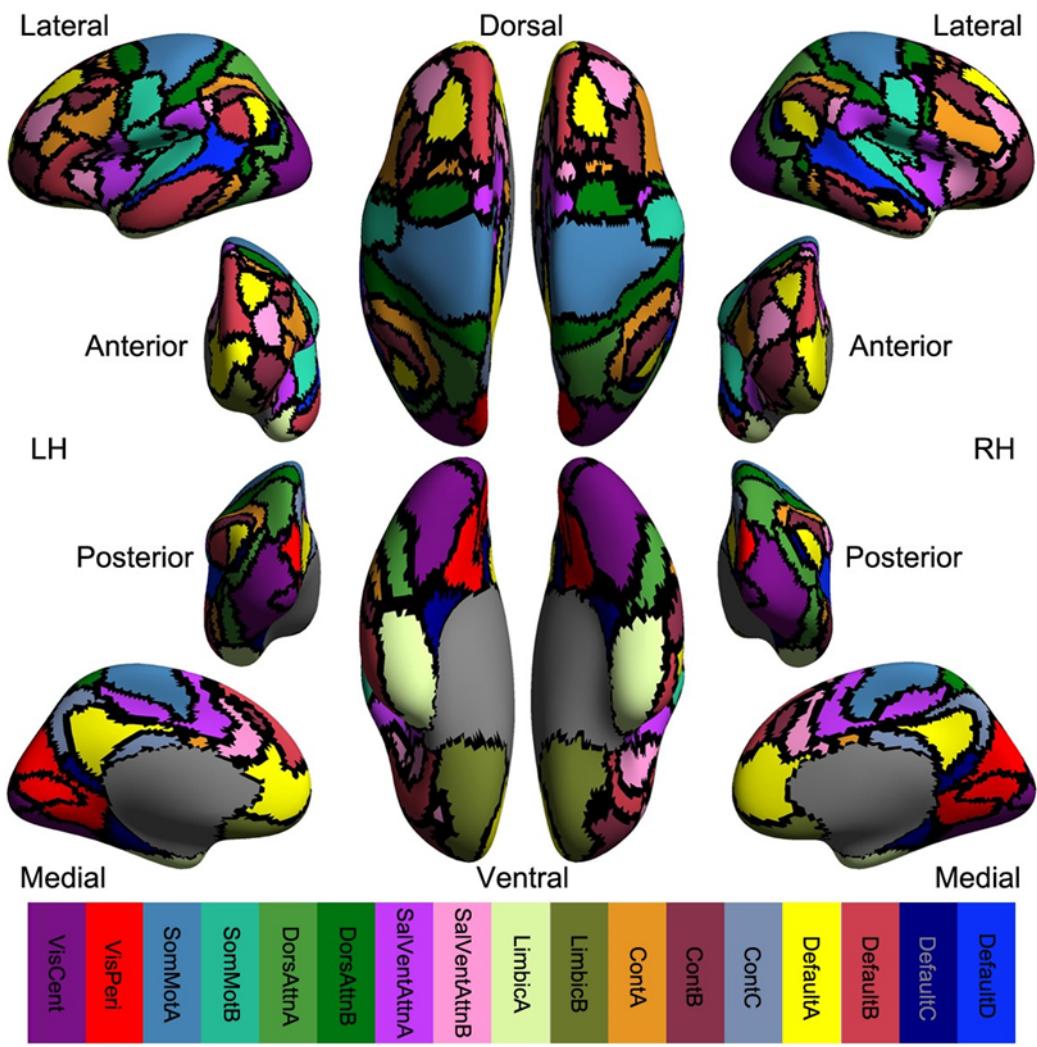


Figure 9.2 Supervised machine learning model

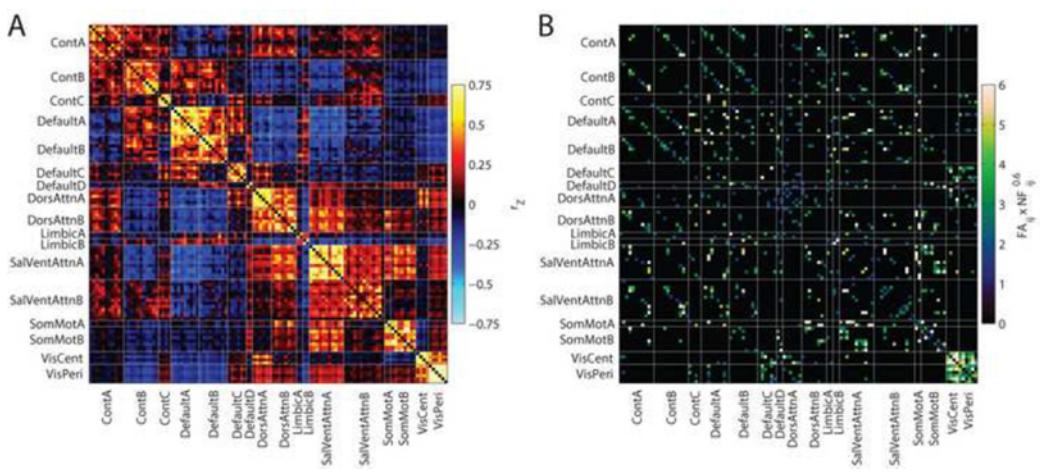
#### 9.4 Multimodal Image Integration Analysis

Based on the cortical coordinate system, CCS can naturally integrate the morphological attributes examined by sMRI and DTI images, and the BOLD signal measured by rfMRI. For example, based on resting rfMRI images of 1000 people, the cerebral cortex can be divided into 17 different functional connectivity networks(including 131 non overlapping brain regions) (Figure 9.4 ).

Such functional parcellation can be directly and concretely applied to practical problems. For example, Figure 9.5 is a connectivity matrix of structure (white matter fiber connectivity) and function (resting state functional connectivity) based on multimodal data throughout life span. It enables us to explore the internal relationship between structure and function throughout the life span and understand the mechanism of the brain.



**Figure 9.3 Functional connectivity network map of cerebral cortex**



**Figure 9.4 Connectivity matrix of structure and function**

# Chapter 10 Visualization

In the study of human brain connectome, MRI data involves the spatial-temporal dimension. The brain has its spatial physical features and temporal dynamic characteristics. With the development of imaging technology, how to mine and visualize these information efficiently and directly becomes very important.

## 10.1 Basic Elements

In the visualization of brain connectome, there are several basic elements that need to be considered: (1) brain model template, (2) color matching, and (3) text description. The visual template should be determined according to the geometric representation of the brain map. If the connectome is based on 3D images, the standard 3D template is usually used. If it is based on the results of 2D cortical grid, then the 2D standard cortical grid model should be used. The palette should be determined in a comprehensive way considering the distribution of the values of the map. If all the values are positive, choose the warm color system, otherwise, choose cold color system. Choose a double color system if both positive and negative values are available. The text description should be short and accurate, and the layout should give full consideration to the relative position and harmonious matching with color pictures.

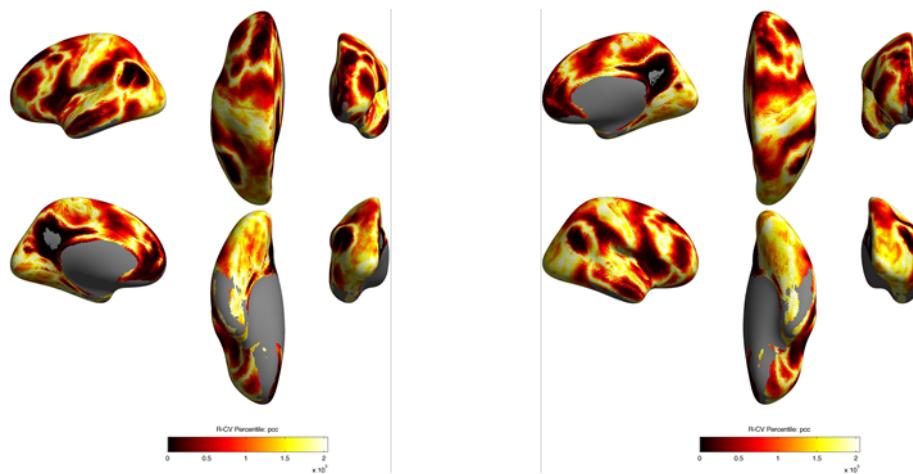


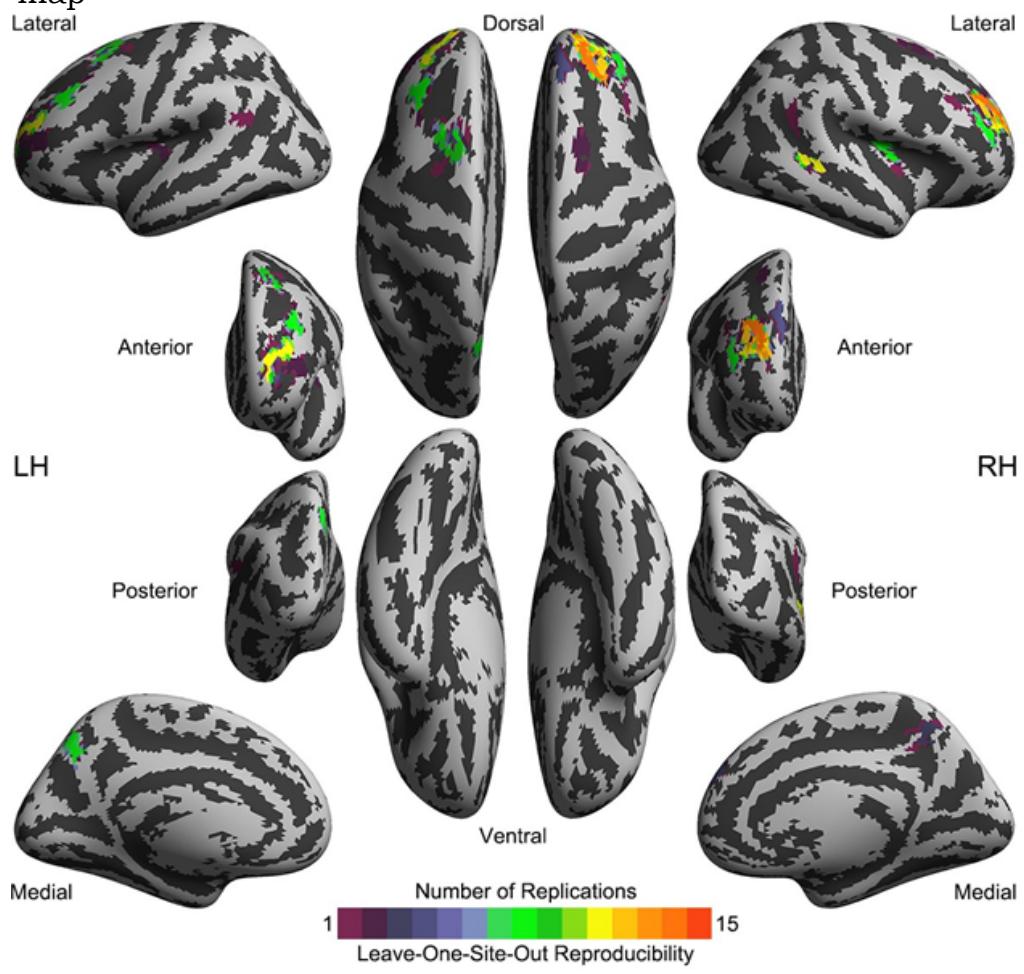
Figure 10.1 Hemisphere surface map- DMN functional connectivity distribution

## 10.2 Toolbox

CCS provides several toolbox functions for the visualization (vistool).

- [`ccs\_SurfStatView.m`](#) : render cortical surface map
- [`ccs\_hemiSurfStatView.m`](#) : render hemispheric cortical surface map
- [`ccs\_mkcolormap.m`](#): create color map according to the picture
- [`ccs\_surf\_split.sh`](#): render the split cortical surface map
- [`ccs\_surf\_montage.sh`](#): render the cortical surface map then split

- [ccs\\_hemiFS\\_lh\\_split.sh](#): render the split left hemisphere surface map
- [ccs\\_hemiFS\\_rh\\_split.sh](#): render the split right hemisphere surface map



**Figure 10.2 Rendering cortical surface map**

## Supplementary Material CCS Sample Scripts

The main directory of CCS contains the scripts of each step for the preprocessing and postprocessing. The following table shows the script name and corresponding steps.

Table 1 The names of CCS scripts and the corresponding steps

| Script name                        | Step   |
|------------------------------------|--|
| <b>ccs_01_</b>                     | multimodal image preprocessing and quality check   |
| ccs_01_anatpreproc.sh              | sMRI preprocessing ( <b>updated</b> )  |
| ccs_01_anatsurfrecon.sh            | sMRI cortical reconstruction ( <b>updated</b> )  |
| ccs_01_anatcheck_surf.sh           | quality check of sMRI segmentation   |
| ccs_01_funcpreproc.sh              | rfMRI preprocessing  |
| ccs_01_dtipreproc.sh               | DTI preprocessing  |
| <b>ccs_02_</b>                     | multimodal image registration and quality check  |
| ccs_02_anatregister.sh             | sMRI registration  |
| ccs_02_anatregister_refine.sh      | sMRI registration(refine)  |
| ccs_02_anatcheck_fnirt.sh          | quality check of sMRI nonlinear registration   |
| ccs_02_funcbbregister.sh           | rfMRI registration (individual space)  |
| ccs_02_funcregister.sh             | rfMRI registration (standard space)  |
| ccs_02_funccheck_bbregister.sh     | quality check of rfMRI BBR   |
| ccs_02_funccheck_fnirt.sh          | quality check of rfMRI nonlinear registration  |
| ccs_02_dtibbregister.sh            | DTI BBR  |
| ccs_02_dticheck_bbregister.sh      | quality check of DTI BBR   |
| <b>ccs_03_</b>                     | multimodal image segmentation (the segmentation of sMRI has included in the sMRI preprocessing)      |
| ccs_03_funcsegment.sh              | rfMRI segmentation   |
| ccs_03_dtisegment.sh               | DTI segmentation   |
| <b>ccs_04_</b>                     | denoising  |
| ccs_04_funcnuisance.sh             | rfMRI noise regression (head motion, global signal, white matter signal, cerebrospinal fluid signal) |
| <b>ccs_05_</b>                     | final process of rfMRI   |
| ccs_05_funcpreproc_final.sh        | rfMRI filtering and smoothing  |
| ccs_05_funcpreproc_final_nofilt.sh | rfMRI smoothing  |
| <b>ccs_06_</b>                     | single subject post processing   |
| ccs_06_singlesubjectALFF.sh        | ALFF   |
| ccs_06_singlesubjectICA.sh         | ICA  |

|                                 |   |
|---------------------------------|---|
| ccs_06_singlesubjectReHo.sh     | ReHo  |
| ccs_06_singlesubjectSFC.sh      | SFC   |
| ccs_06_singlesubjectVMHC-SFC.sh | VMHC-SFC  |
| ccs_06_singlesubjectVMHC.sh     | VMHC  |
| <b>ccs_07_</b>                  | create group level files  |
| ccs_07_grp_4dmaps.sh            | connect the files of each subject and create group level analysis files |
| ccs_07_grp_boldmask.sh          | create group level mask   |
| ccs_07_grp_meanbold.sh          | calculate mean BOLD signal of group and create mask                     |
| ccs_07_grp_meanstruc.sh         | create mean sMRI image at group level                                   |
| ccs_07_grp_surfcluster.sh       | save the statistically significant cluster results on cortical surface  |

By editing CCS batch script, the above scripts can be employed to process the multimodal image data.

CCS has included the batch script of sMRI and fMRI, and the path is: [ccs/samplesscripts/](#). Users can edit script parameters according to their own needs.

Batch script is divided into two parts: parameters and start of script. The parameters that need to be edited by the user include CCS script path, data storage path, subject list, data name and whether to perform specific data analysis. In the script, users can skip the specific processing step by entering # to comment the statements.

## A. CCS Installation and Usage

### 1) Platform

CCS can be used on Linux/Unix and Mac systems. Also to run CCS you need to install:

- Matlab R2007a
- Python 3.0 or Above. Suggestion to install Anaconda  
<https://www.anaconda.com/products/individual#Downloads>

### 2) Requirements

CCS needs to install these tools in order to run this pipeline

- AFNI  
[https://afni.nimh.nih.gov/pub/dist/doc/html/doc/background\\_in\\_stall/main\\_toc.html](https://afni.nimh.nih.gov/pub/dist/doc/html/doc/background_in_stall/main_toc.html)
- FreeSurfer <https://surfer.nmr.mgh.harvard.edu/>
- FSL <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation>
- SPM12 <https://www.fil.ion.ucl.ac.uk/spm/>
- CAT12 <http://www.neuro.uni-jena.de/cat/index.html#DOWNLOAD>

- Docker and DeepBet <https://github.com/HumanBrainED/NHP-BrainExtraction>

### *3) Install CCS*

After downloading the CCS\_APP folder to the specified location, you need to configure the system environment variables and write the path of CCS\_APP into the environment. Usually, the environment variables are saved at bashrc or bash\_profile on Linux or Mac OX systems, so please first determine the name of the environment variable file used on your system.

```
```bash
echo "export CCS_APP=/dir-to-your-CCS_APP/CCS_APP/" >>
~/.bashrc
```

```

After adding the CCS\_APP path, you can check whether CCS\_APP has been successfully written to the environment variable.

```
```bash
echo $CCS_APP
```

```

If the screen shows the directory where your CCS\_APP is located, the CCS environment variable has been successfully configured.

### *4) Data Preprocessing*

Once the CCS and the corresponding isoftware have been successfully installed according to the above steps, the data can be pre-processed.

- **Data Organizing**

After obtaining the raw dicom data, it needs to be compressed and converted to BIDS format. The specific conversion method can be found on the BIDS format website: <https://bids.neuroimaging.io/>. There are several automated tools available to convert raw dicom data into BIDS format, such as

[dcm2bids](<https://unfmontreal.github.io/Dcm2Bids/>).

- **Transform into CCS**

After organizing the raw data into BIDS format, CCS will generate a folder for data processing based on the BIDS format, usually we name the folder as CCS and store it in the same folder as the raw BIDS data. This is done by running the ccs\_pre\_bidsccs.py command.

```
```bash
BIDS_DIR=/your_project_dir/BIDS
CCS_DIR=/your_project_dir/CCS
python $CCS_APP/ccs_pre_bids2ccs.py --BIDS_DIR $BIDS_DIR --
CCS_DIR $CCS_DIR
```

```

- **Structural image pre-processing**

Before starting the structural image pre-processing, the working path of CCS, the storage path of FreeSurfer and the subject number to be processed need to be defined first.

```
```bash
CCS_DIR=/your_project_dir/CCS
SUBJECTS_DIR=/your_project_dir/FreeSurfer
subject=/your_project_dir/CCS/001
````
```

First, structural image pre-processing is performed, including steps such as denoising and skull stripping.

```
```bash
$CCS_APP/ccs_anat_01_pre_freesurfer.sh $CCS_DIR
$SUBJECTS_DIR $subject
````
```

Once these steps have been completed, the skull stripping effect needs to be checked to determine whether to continue with the next step of data processing.

The second two steps of the CCS structural image pre-processing are the cortical reconstruction pipeline and the structural image alignment pipeline, as the following commands.

```
```bash
$CCS_APP/ccs_anat_02_freesurfer.sh $CCS_DIR $SUBJECTS_DIR
$subject
$CCS_APP/ccs_anat_03_postfs.sh $CCS_DIR $SUBJECTS_DIR
$subject
````
```

- Resting-state functional image pre-processing

The preprocessing of the functional image starts with modifying the template\_prepfuncpart.sh file in the CCS\_APP directory by filling in the parameters corresponding to the functional image:

- CCS\_DIR /your\_project\_dir/CCS
- SUBJECTS\_DIR /your\_project\_dir/FreeSurfer
- rest\_dir\_name (default:rest)
- rest\_name (default:rest)
- TR (default 2s)
- numDropping (Dropping first 10s of rest data. default:5 )
- sliceOrder (Tpattern: see helps from AFNI command 3dTshift, default: alt+z, if the sequence is multi-band:mbd)
- FWHM (default: 6)

After modifying the contents of the template, the following command can be run to start the pre-processing of the functional image.

```

```bash
CCS_DIR=/your_project_dir/CCS
subject=001
mkdir -p $CCS_DIR/$subject/scripts/
sed "s/CCSsubjectname/$subject/"
$CCS_APP/template_preproc_funcpart.sh >
$CCS_DIR/$subject/scripts/ccs_preproc_funcpart.sh
$CCS_DIR/$subject/scripts/ccs_preproc_funcpart.sh
```

```

## B. Calculation of Connectome Map

Batch script path for individual 3D (volume-based) connectome map: [ccs/samplesScripts/ccs\\_postproc\\_template.sh](#). The parameters need to be edited in the script are described as follows:

- `scripts_dir`: the storage path of CCS script in user's computer
- `analysisdirectory`: the storage path of data that need to be processed
- `subject_listanat_name`: path of subject list, the subject list contains the data name
- `anat_name`: name of sMRI data
- `rest_name`: name of rfMRI data
- `anat_dir_name`: name of sMRI directory
- `func_dir_name`: name of rfMRI directory
- `TR`: the scanning parameters of rfMRI, TR
- `do_anat_reg`: whether to carry out sMRI registration, YES = true, NO = false
- `do_anat_seg`: whether to carry out sMRI segmentation, YES = true, NO = false
- `fs_brain`: whether to use the brain tissue extracted by FreeSurfer for analysis, YES = true, NO = false
- `svd`: whether to decompose signal by singular method, YES = true, NO = false
- `gs_removal`: whether to remove the global signal, YES = true, NO = false

The step-by-step scripts called by the batch script are shown below:

- [ccs\\_06\\_singlesubjectALFF.sh](#): ALFF calculation
- [ccs\\_06\\_singlesubjectICA.sh](#): ICA calculation
- [ccs\\_06\\_singlesubjectReHo.sh](#): ReHo calculation
- [ccs\\_06\\_singlesubjectSFC.sh](#): SFC calculation
- [ccs\\_06\\_singlesubjectVMHC-SFC.sh](#): VMHC-SFC calculation
- [ccs\\_06\\_singlesubjectVMHC.sh](#): VMHC calculation

CCS also provides the MATLAB code of connectome map calculation (2D), the batch file is [ccs/ samplesScripts/runSurfaceCCS\\_](#)

`template.m`. The parameters need to be edited in this script are described as follows:

- `ccs_dir`: the path of CCS script in user's computer
- `ana_dir`: the storage path of data that need to be processed
- `ccs_matlab`: the path of matlab script
- `sub_list`: path of subject list, the subject list contains the data name
- `rest_name`: name of rfMRI data
- `grpmask_dir`: path of group mask
- `grptemplate_dir`: path of group template
- `fs_home`: installation path of FreeSurfer

Table 2 Names of MATLAB scripts and the corresponding steps

| Script name                                      | Step of Functionality                   |
|--|---|
| <code>ccs_06_</code>                             | postprocessing of rfMRI                 |
| <code>ccs_06_singlesubject2dALFF.m</code>        | 2D ALFF                                 |
| <code>ccs_06_singlesubject2dConnDensity.m</code> | 2D connectivity density                 |
| <code>ccs_06_singlesubject2dReHo.m</code>        | 2D ReHo                                 |
| <code>ccs_06_singlesubject2dSFC.m</code>         | 2D SFC                                  |
| <code>ccs_06_singlesubject2dVNCM.m</code>        | centrality of whole brain vertexes (2D) |
| <code>ccs_06_singlesubjectDMRIparcels.m</code>   | 165 parcellations of sMRI               |
| <code>ccs_06_singlesubjectParcelALFF.m</code>    | ALFF based on parcellation              |
| <code>ccs_06_singlesubjectParcelCCC.m</code>     | ICC based on parcellation               |
| <code>ccs_06_singlesubjectRFMRIparcels.m</code>  | 165 parcellations of rfMRI              |
| <code>ccs_06_singlesubjectVNCM.m</code>          | centrality of whole brain voxels (3D)   |
| <code>ccs_07_</code>                             | create group-level file                 |
| <code>ccs_07_grp_SurfMask.m</code>               | create surface mask at group level      |