DIAGNOSIS OF AUTISM STPECTRUM DISORDER BASED ON BRAIN NETWORK CLUSTERING

A Thesis Submitted to the

College of Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon

By

Lingkai Tang

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5A9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

ABSTRACT

Developments in magnetic resonance imaging (MRI) provide new non-invasive approach—functional MRI (fMRI)—to study functions of brain. With the help of fMRI, I can build functional brain networks (FBN) to model correlations of brain activities between cortical regions. Studies focused on brain diseases, including autism spectrum disorder (ASD), have been conducted based on analyzing alterations in FBNs of patients. New biomarkers are identified, and new theories and assumptions are proposed on pathology of brain diseases.

Considering that traditional clinical ASD diagnosis instruments, which greatly rely on interviews and observations, can yield large variance, recent studies start to combine machine learning methods and FBN to perform auto-classification of ASD. Such studies have achieved relatively good accuracy. However, in most of these studies, features they use are extracted from the whole brain networks thus the dimension of the features can be high. High-dimensional features may yield overfitting issues and increase computational complexity. Therefore, I need a feature selection strategy that effectively reduces feature dimensions while keeping a good classification performance.

In this study, I present a new feature selection strategy that extracting features from functional modules but not the whole brain networks. I will show that my strategy not only reduces feature dimensions, but also improve performances of auto-classifications of ASD. The whole study can be separated into 4 stages: building FBNs, identification of functional modules, statistical analysis of modular alterations and, finally, training classifiers with modular features for auto-classification of ASD. I firstly demonstrate the whole procedure to build FBNs from fMRI images. To identify functional module, I propose a new network clustering algorithm based on joint non-negative matrix factorization. Different from traditional brain network clustering algorithms that mostly perform on an average network of all subjects, I design my algorithm to factorize multiple brain networks simultaneously because the clustering results should be valid not only on the average network but also on each individual network. I show the modules I find are more valid in both views. Then I statistically analyze the alterations in functional modules between ASD and typically developed (TD) group to determine from which modules I extract features from. Several indices based on graph theory are calculated to measure modular

properties. I find significant alterations in two modules. With features from these two modules, I train several widely-used classifiers and validate the classifiers on a real-world dataset. The performances of classifiers trained by modular features are better than those with whole-brain features, which demonstrates the effectiveness of my feature selection strategy.

Keywords: autism spectrum disorder, functional brain network, auto-classification, network clustering

ACKNOWLEDGEMENTS

I would like to convey my most sincere gratitude to my supervisor, Professor Fang-Xiang Wu, for his everlasting patience and all those inspiring discussions. Without help from Dr. Wu, I will not be able to finish my research and this thesis. Also, I should thank him for leading me into this fascinating area and encouraging me to learn about image processing, network analysis, machine learning, etc.

I would like to thank other committee members: Professor Travis Wiens and Professor Daniel Chen, for their suggestions and comments on my research and thesis.

My appreciation also goes to the group members, who are Lin Wu, Yichao Shen, Ping Luo, Fei Wang, Ali Jamali Beyrami, Sakib Mostafa, et al. I should also thank all my other friends here. Thank you all for your support and advice.

Finally, my special gratitude belongs to my parents. Without their support, I will never have the chance to study in this great university and meet all these great people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABIDE	Autism Brain Imaging Data Exchange
AD	Alzheimer's Disease
ADHD	Attention Deficit Hyperactivity Disorder
ADI-R	Autism Diagnostic Interview, Revised
ADOS	Autism Diagnostic Observation Schedule
ANCOVA	Analysis Of Covariance
ANN	Artificial Neural Network
ANOVA	Analysis Of Variance
ARI	Adjusted Rand Index
ASD	Autism Spectrum Disorder
AUC	Area Under Curve
BOLD	Blood Oxygen Level Dependent
CSF	Cerebrospinal Fluid
CT	Computed Tomography
DMN	Default Mode Network
DWI	Diffusion Weighted Imaging
FBN	Functional Brain network
fMRI	Functional Magnetic Resonance Imaging
FPN	Frontal-Parietal Network
FPR	False Positive Rate
FWHM	Full Width at Half Maximum
GM	Grey Matter
JSNMF	Joint Symmetrical Non-negative Matrix Factorization
kNN	K Nearest Neighbors
LDA	Linear Discriminant Analysis
LOOCV	Leave-One-Out Cross Validation
LR	Logistic Regression
LRLR	Lasso-Regularized Linear Regression

MCI	Mild Cognitive Impairment
MRI	Magnetic Resonance Imaging
MSC	Multi-view Spectral Clustering
NMF	Non-negative Matrix Factorization
NMI	Normalized Mutual Information
PCC	Pearson Correlation Coefficient
PET	Positron Emission Tomography
PSO	Particle Swarm Optimization
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
rs-fMRI	Resting State Functional Magnetic Resonance Imaging
SC	Spectral Clustering
SNMF	Symmetrical Non-negative Matrix Factorization
SVM	Support Vector Machine
TD	Typically Developed
TPR	True Positive Rate
WM	White Matter

CHAPTER 1
INTRODUCTION

1.1 Background

1.1.1 Magnetic Resonance Imaging

As the center of nervous system, brain is the most complex organ in a human body, controlling other parts of body with neuronal signals or hormones. Over the past decades, advances in brain imaging techniques, such as magnetic resonance imaging (MRI), computed tomography (CT) or positron emission tomography (PET), provide various non-invasive approaches to study brain anatomies and functions or diagnose brain diseases *in vivo*. Among all the imaging techniques, MRI stands out in brain studies, providing more details in soft tissues which are major components of brains [1]. Also compared with CT or PET, MRI does less harm to bodies due to exclusion of radiological materials.

The physics of MRI can be generalized as calculating the time constant of hydrogen atoms in the scanned part of body returning from an excited state to an equilibrium state [2]. The atoms are put into a static magnetic field and are excited or magnetized by another oscillating magnetic field applied to it. After the oscillating magnetic field is removed, the exited atoms can recover to an equilibrium state and during this process, they emit radio frequency signals deteriorating exponentially with time that can be detected by a receiving coil. Different time constants of different tissues yield different magnitudes of MR signals which furtherly lead to the contrast in MRI images. Adjusting configuration of MRI scanning, I can detect boundary between two types of tissues. Distinguished by the direction of the radio frequency signal I measure from, there are T1- and T2-weighted images, where T1 measures the signal in the same direction as the static magnetic field and T2 transverse to the static magnetic field. T1-and T2- weighted images provide complementary information to each other. T1-weighted signals are stronger at

fatty tissues, or white matters (WM) in brains, while T1-weighted signals are stronger at more water content, or grey matters (GM) [3]. T1-and T2- weighted MRI images have been widely used in clinical diagnosis of brain diseases, such as Alzheimer's disease, brain tumor, infectious disease, or traumatic brain injuries [4, 5, 6]. Researchers also attempt to link psychological disorders with abnormalities in MRI images [7].

Other MRI techniques are developed to study brains by setting different MRI pulse sequences, such as functional magnetic resonance imaging (fMRI) and diffusion weighted imaging (DWI). Measuring the blood-oxygen-level dependent (BOLD) signals, fMRI provides information on level of brain functional activities. When the neurons return to the original state from an activated state, they pump ions across the neuronal cell membranes. Since neurons themselves do not preserve energy, more glucose and oxygen must be transported immediately by blood for the ion pumps to consume. Such a process leads to a temporary increase in the concentration of oxygenated hemoglobin in the activated area. Oxygenated and deoxygenated hemoglobin react differently to the oscillating magnetic field from the MRI scanner. Deoxygenated hemoglobin loses magnetization faster, which yields contrast between activated and deactivated areas. In practice, I usually obtain a series of fMRI images along time to record fluctuation of functional activity during a period. Depending on the state of subjects during scanning, fMRI could be catalogued into resting state fMRI (rs-fMRI) and task related fMRI, where task related fMRI requires subjects to perform specific tasks while rs-fMRI, on the contrary, asks them to do nothing. While fMRI focuses on function of brain, DWI depicts the neuronal structure. DWI measures the rate of diffusion of water molecules in different directions. This relies on the fact that brain white matter has a fibrous structure similar to some anisotropic crystals and water flows more rapidly along the direction of fibers. Fractional anisotropy, which measures the degree of a diffusion process towards each direction, can be calculated to show that, at each location which direction neuronal fibers are more likely to expand to, indicating which brain areas are neuronally connected.

Currently, DWI and fMRI have been adopted in many researches. Studies based on fMRI images have identified several brain regions coupled with critical cognitive functions, including vision, moving, emotion, memory, etc. [8]. Also, abnormalities in fMRI images are mapped to

pathology and symptom of brain diseases, such as Alzheimer's disease (AD) or brain tumor [8], as well as neurodevelopmental diseases such as autism spectrum disorder (ASD) [9]. Compared with academical communities, fMRI is less accepted in clinical practices, but still has potential to provide complementary information on testing efficiency of treatments or degree of recovery of brain from diseases [10, 11]. DWI has also been implemented in several studies revealing subtle alterations in various diseases that relate to white matter abnormalities, e.g., AD is referred as neuronal disconnection [12, 13] and ASD is characterized as impairment of white matter neural circuitry yielding damaged high-order brain functions [14].

1.1.2 Brain Network Analysis and Clustering

Having largely promoted my understanding on cytoarchitecture and functionality of brain, conventional methods focus on the brain images themselves, while recent studies find an alternative path to study brains from a network perspective. Since brains are complex systems functioning based on billions of connected neurons, naturally, brains can be viewed as networks. Compared with images, brain networks contain extra information about interaction among regions. With the help from graph theory and network analysis methods, studies on brain network analysis have revealed various insights on organization of functional and structural connectivity of brain cortices, including identification of critical hubs and modules, or study of global properties [15, 16, 17]. The results of such studies are mostly in a good agreement with results from anatomical experiments or studies based on images, meanwhile, help bring out new theories and explanations.

Generally, there are two types of brain network—structural brain network and functional brain network (FBN). Structural network captures the neuronal structures, i.e., which brain regions are connected densely to each other by neurons, while functional network shows the functional organization, i.e., how brain regions correlated or cooperated with each other when performing certain cognitive functions. Thus, structural networks are built upon DWI images, usually by drawing tractography, and functional networks are derived from fMRI images, by calculating correlation values between BOLD signals. Networks built from task related fMRI can help link

brain activities to specific tasks. However, rs-fMRI is still widely-used in brain network analysis because even though the brain is in resting state, regions that normally interact with each other frequently still show strong correlation between them [18]. Therefore, FBNs built upon rs-fMRI provide intrinsic functional organization that is not biased by cognitive tasks.

Numerous methods of network analysis have been implemented for brain networks, among which network clustering methods are proved to be efficient in identifying modules. Clustering, or cluster analysis, can be defined as a process of partitioning a set of objects into several groups where objects in the same group show more similarity or are closely related in certain ways. Such a group is usually called a cluster. For network clustering, I consider each vertex as an object and cluster the network based on connectivity among vertices. Vertices in the same cluster are supposed to be highly connected with each other and perform similar functions or serve common goals, and furtherly affect the functionality of the whole network. A special case of a cluster is called a clique where vertices in the cluster are connected to each other.

Previous studies on brain network analysis have shown that brain networks have certain small-world properties, such as high clustering coefficient and low characteristic path length [19, 20]. Small-world networks tend to have more cliques or nearly cliques, or in another words, vertices tend to form clusters. This matches my understanding that brain is a modular system where different brain parts have separated cognitive functions or anatomical organizations. In brain network clustering, especially for functional networks, a cluster is usually called a module, indicating the similarity of function inside the cluster. Thus, it is feasible to identify brain modules based on clustering methods. Implementations of clustering methods have successfully identified modules from different aspects of view. Study in [17] maps modules to cognitive functions while study in [21] links clusters with 4 types of behaviors.

1.1.3 Brain Disease Related Studies Based on Brain Networks

Previous studies also applied brain network analysis to brain diseases, including ASD [22, 23, 24, 14, 25], AD [26, 27, 28, 29] or schizophrenia [30, 31, 32]. ASD is a neurodevelopmental disorder characterized by repetitive social behavior, restricted interests and mental inflexibility

[33]. It is estimated that about 1% of global population is suffering from ASD [34] and the percentage has the trend to increase in the past two to three decades [35, 36]. Increased public awareness on ASD has appealed to numerous studies focused on different perspectives. From brain images, various biomarkers shared widely among subjects with ASD have been identified, such as regional and global overgrowth of brain volumes in early childhood [37] or hyper- and hypo-activation of cortical regions [38, 39, 40, 41, 42].

As mentioned before, brain network analysis has yielded new results in organization of brain connectivity, so naturally, it can be promising to use it to study ASD. Advanced imaging techniques, such as DWI and fMRI, also provide valid data sources to build brain networks. Recent studies on brain network analysis on networks of ASD group and typically developed (TD) group have found various alterations possibly caused by ASD. Hyper- and hypo-connectivity among certain regions are widely reported in several investigations [43, 44, 45], indicating possible causes of ASD symptoms. Developmental changes in brain networks are also studied, using data collected from subjects with different ages [46, 47]. There are also studies about the differences appeared in subcategories of ASD, such as high-functioning ASD and Asperger syndrome, and diseases closely related to ASD, e.g., attention deficit hyperactivity disorder (ADHD) [48, 49, 50].

Machine learning based methods, including deep learning methods, are also implemented in many recent studies in analyzing brain networks of ASD subjects. Machine learning means building a mathematical model to make decisions with a given dataset while the model is not programmed explicitly for the task, or the model learns to make correct decisions by itself progressively [51]. Generally, there are unsupervised machine learning referred as data mining and supervised machine learning referred as predictive analytics. In unsupervised machine learning, the machine is not told the labels of sample, while supervised machine learning is on the contrary. Unsupervised machine learning is widely used for cluster analysis, which divide samples into groups, while supervised machine learning is mostly developed for predicting unknown labels. Studies using machine learning on brain diseases are mostly focused on identifying new biomarkers [52, 53] and auto-classification of different groups of subjects [54, 55, 56], i.e., ASD versus TD. Clinical ASD diagnosing approaches, require much

communication and observation of subjects, thus the results could be easily biased by clinicians' personal judgements and expectations [57]. So auto-classification can provide complementary approaches for clinical diagnosis.

1.2 Problem Statement

As mentioned above, small-worldness of brain networks guarantees accuracy of network clustering methods to find modules, and existing methods have successfully identified some. However, clustering methods previously implemented may be flawed as previous studies mostly calculate an average network as the input to the clustering algorithms. Generally, an average network is obtained by taking average over all individual networks which are built upon data collected from single subjects. However, by taking average, information contained by individual networks could be lost. Another point is that resulted modules are valid on the average network but may be less valid on each individual network. Although the individual networks are not the same, I believe the modular organization of brain networks of different subjects should not differ much. On the other hand, if I cluster on single individual networks, the clustering results could differ much from each other. Therefore, I should develop a clustering algorithm that identifies valid modules for each subject.

Previous studies on auto-classification of ASD mainly use features extracted from the whole brain to train the classifiers. Considering the sizes of brain networks are usually large, the dimensions of feature vectors could also be relatively high. In machine learning, high dimensional feature vectors could lead to overfitting issues and increased computational complexity. Therefore, in this study, to reduce feature dimension, I propose a new feature selection strategy that only use features from a module identified by a clustering algorithm. To determine from which module I obtain features, I also test the significance level of modular differences between ASD and TD subjects. Considering previous studies are mostly focused on global or nodal alterations, investigating alterations in modules can also build a clearer link between functions of brain and ASD, since global alterations may fail to connect ASD

symptoms to cerebral regions. Also, since vertices in modules are highly correlated, alterations in one vertex may have an impact on others or the whole module, either.

1.3 Objectives and Basic Ideas

Generally, the goal of this research is to develop a new strategy to selection features from a single module to reduce feature dimension or disease diagnosis.

To achieve this objective, firstly, I propose a new network clustering algorithm designed for brain networks. The algorithm is based on non-negative matrix factorization (NMF), but different from ordinary NMF, my algorithm factorizes multiple matrices simultaneously. I evaluate the clustering results with some common indices and demonstrate my method outperforms other competing ones.

Secondly, to demonstrate there are actual differences between modules of ASD and TD networks, and to choose the module to extract features from, I calculate some modular indices and use t-test to demonstrate the significance level of modular difference. I find that one module has significant differences than others, so I extract modular features from this module.

Thirdly, I train some classifiers which were also previously used in literature, with modular and whole-brain features, separately. I show that classifiers trained with modular features have better performances.

1.4 Thesis Organization

In Chapter 1, I introduce the background of this study, the problem statement, the major objective and my basic idea for achieving it and the overview of this thesis. In Chapter 2, I introduce the dataset I use and the whole process to build brain networks from MRI images. In Chapter 3, I introduce a new network clustering algorithm for identifying brain modules and I show the clustering results. In Chapter 4, I do some statistical analysis to find significant

modular alterations. In Chapter 5, I use modular features to do classification of ASD with several commonly-used classifiers. In Chapter 6, I conclude this study.

CHAPTER 2
BRAIN IMAGE DATASETS AND BUILDING BRAIN NETWORKS

In this chapter, I introduce the database I collect from and more details of the dataset I use for both clustering and classification. Then I introduce the whole procedure to build brain networks from MRI images, including the preprocessing of images, defining vertices by brain cortex partitioning, and using information in images to derive edges. As introduced before, there are structural networks and functional networks. However, in this study, I only focus on FBNs. Therefore, in this chapter, I only introduce the procedure to build FBNs.

2.1 Database and Dataset

The data used in this research can be found in Autism Brain Imaging Data Exchange (ABIDE, http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html) [58]. Up to date, the database has included data of over 1500 subjects, uploaded by multiple institutions, following the same data aggregation strategy. In this database, data of each subject usually contain both structural images (T1-weighted, [59]) and functional images (fMRI), as well as phenotypic information such as test scores of ASD assessing and diagnosing instruments. Other information is also included, like age of subjects, IQ, sex, medication situation, etc.

The MRI image data I use is acquired in UCLA with the configuration of scanning introduced in [24]. The dataset totally contains 79 subjects, including 37 from TD subjects and 42 from ASD subjects. Ages of the subjects ranges from 8.36 to 18.18 with a mean of 13.20 and standard deviation of 2.43. For each subject, I build an FBN. The detailed procedure is introduced in the following sections, including preprocessing of MRI images, and defining vertices and edges. I should notice that all FBNs are built from rs-fMRI images, but the network building pipeline can also be implemented to task-related fMRI data. The FBNs can be also found in UCLA

multimodal connectivity database [60].

2.2 Preprocessing of fMRI Images

Image data are preprocessed using two toolboxes, AFNI [61] and FSL [62], installed in Linux environment (Ubuntu 18.04). The two toolboxes provide various functions for image processing and statistical analysis of several types of MRI images. In the preprocessing stage, I use both T1-weighted images and rs-fMRI images, but the T1-weighted images are only for image registration and brain segmentation, which means they are only used in this section. As I know, fMRI data contains a series of images along time. A single image is also called a volume or a brick in fMRI data processing. Images of each subject are processed independently.

The whole preprocessing scheme is inspired by a pipeline incorporated in AFNI but with my own modifications. In functional BOLD images and T1-weighted images, brain-only images are extracted from skulls and other surrounding noise areas using AFNI (3dAutomask and 3dSkullStrip, with default settings). Functional volumes on time series are motion corrected using FSL MCFLIRT tool [63]. Volumes are registered to a mean volume with a normalized correlation cost function and are resampled with sinc interpolation. In general, image registration means aligning multiple images by rotating, translating, scaling and skewing them to achieve maximal overlapping. In this case, overlapping is measured by normalized correlation. Sinc interpolation is to fill the empty space between voxels to increase the resolution. 6 parameters corresponding to 6 degrees of freedom for rigid body movement (translation and rotation along x, y, z axes) are calculated for each volume from MCFLIRT. If the root mean square displacement over all voxels between two consecutive volume is over 2.5 mm, I consider this subject as an outlier and leave it out for any future processing and analysis. Images are applied a Gaussian kernel with full width at half maximum (FWHM) of 5.0 mm for spatial smoothing. 9 covariates are regressed out of the images including 6 parameters of rigid body movement and average time series of WM, Cerebrospinal fluid (CSF) and the whole brain. I consider the 9 covariates as the noise factors and use linear regression to calculate the residual

as the denoised volumes. WM and CSF signals are regressed out because most of neuronal activities occur in GM, since GM contains more neuronal cell bodies and less axons, compared with WM. Thus, WM and CSF signals mostly reflect non-neural fluctuations such as scanner instability, which may lead to unnecessarily increased inter-regional functional correlation strength [64, 65]. Average whole brain signal is also considered to contain noises such as respiratory and cardiac noise or psychological noises [66]. I also regress out head motion parameters because it may cause spurious signal and affect further network analysis [67]. To separate WM and CSF signal, I first segment WM and CSF tissues with FSL FAST [68] on T1-weighted images, then calculate their average time series with AFNI 3dmaskave. The regression is fulfilled with AFNI 3dDeconvolve and 3dREMLfit. A bandpass filter of 0.01-0.1 Hz is applied to the images to isolate fluctuation caused by respiration and heartbeat. All functional volumes are registered to the MNI 152 standard space, which is an average of 152 T1-weighted images and is integrated in FSL, using a structural image as a medium, i.e., functional volumes are firstly registered to structural images, then structural images are registered to MNI 152 standard space and two transformations are concatenated. The two-step registration is conducted in FSL FLIRT, with affine transformation (12 degrees of freedom) and mutual information cost function for both steps. In motion correction, I only use 6 parameters since the sizes of the brains in each volume are the same, though 12 parameters also describing skewing other than rotation and translation.

This pipeline includes steps that are shared in many researches, but the actual implementation of each step, like the parameter settings, or the order of the steps, may vary in other studies. Especially for the noise regression step, many other regressors can be added, such as the derivatives or the principle components of the regressors I choose. Additionally, although the use of global average signal is commonly-used, but there are several studies showing that regressing out global signal may increase negative connectivity [69, 70]. Therefore, the pipeline should be adjusted for different situations.

2.3 Building Brain Networks

Networks contain two elementary objects: vertices and edges connecting vertices. Thus, building brain networks from brain images contains two steps: defining vertices and generating connections among vertices.

2.3.1 Defining Vertices

Vertices in brain networks represent brain regions, or regions of interest (ROI). I partition the whole brain cortex into small ROIs and each ROI corresponds to a vertex in the networks and networks of all subjects share the same set of ROIs.

In previous studies, anatomical atlases are implemented to define ROIs [16, 15, 71]. An anatomical atlas is a partitioning of cortex based on anatomical features such as sulci and gyri. Commonly used anatomical atlases includes Desikan Atlas, Destrieux Atlas, AAL, etc. [72, 73, 74]. Such atlases have been proved efficient in building structural brain networks but maybe defective for functional networks. The regions from anatomical atlases are relatively large so each of them may cover several sub-regions with different functions, from where the signals collected are a mixture corresponding to several functions and distort the networks. In some other cases, voxelwise ROIs are defined to reach a high resolution [17]. Here, each ROI is a voxel in brain images, so the sizes of resulted brain networks could be extremely large, meanwhile, carrying lots of information irrelevant to global analysis [16] (e.g. A network built with voxelwise ROI in [17] contains over 44,000 vertices.).

The brain networks used in this research are built on ROIs introduced in [17]. Different from anatomical or voxelwise atlas mentioned above, ROIs here are defined based on brain functional information, or on fMRI images. The basis of defining ROIs here is to find regions obviously activated or distinguished from neighbors in rs-fMRI or task-related fMRI. For task-related fMRI, the subjects are asked to perform a series of tasks including verb generation, button pushing etc. 322 ROIs are identified to be activated through several tasks. For rs- fMRI, the images show obvious transition of fMRI signal correlation which forms boundaries that

partition the cortex into regions. 254 ROIs are identified from this method. Combining ROIs from the two methods and removing overlapped ones, 264 ROIs are finally generated and modeled into vertices. The 264 ROIs can be represented by spheres with 5 mm radius and centered at coordinates in MNI 152 standard space.

2.3.2 Defining Edges

Identifying connections among ROIs is the second step to build brain networks. Functional and structural brain networks can have identical ROIs but how the connections are defined distinguishes them. Functional connections are supposed to quantify functional correlation or similarity between cortical regions, while structural connections focus on describing neuronal structure bedding inside cerebral cortex.

Functional connections, as mostly rely on fMRI data, are built upon the fact that small cortical regions are not acting alone. In fMRI scanning, I can obtain a signal in time series for each ROI I defined in the last step. Then between every pair of ROIs, Pearson correlation coefficient (PCC) can be calculated to quantify the correlations. For two time series T_1 and T_2 with the same dimension, PCC can be defined as follows:

$$PCC(X,Y) = \frac{cov(T_1, T_2)}{\sigma_{T_1}\sigma_{T_2}} \tag{2.1}$$

where cov represents the covariance and σ is the standard deviation. PCC ranges from -1 to 1 where 1 means signals from two ROIs are perfectly positive correlated with each other while -1 means perfectly negative correlated. The value of PCC between two ROIs is also the weight of edge between two corresponding vertices in the functional network. The networks are represented by 264×264 connectivity matrices, where each row or column corresponds to a vertex and each element is the edge weight or functional correlation between two vertices. In Figure 2.1, I present a flowchart demonstrating the whole process for building FBNs, including preprocessing of fMRI images.

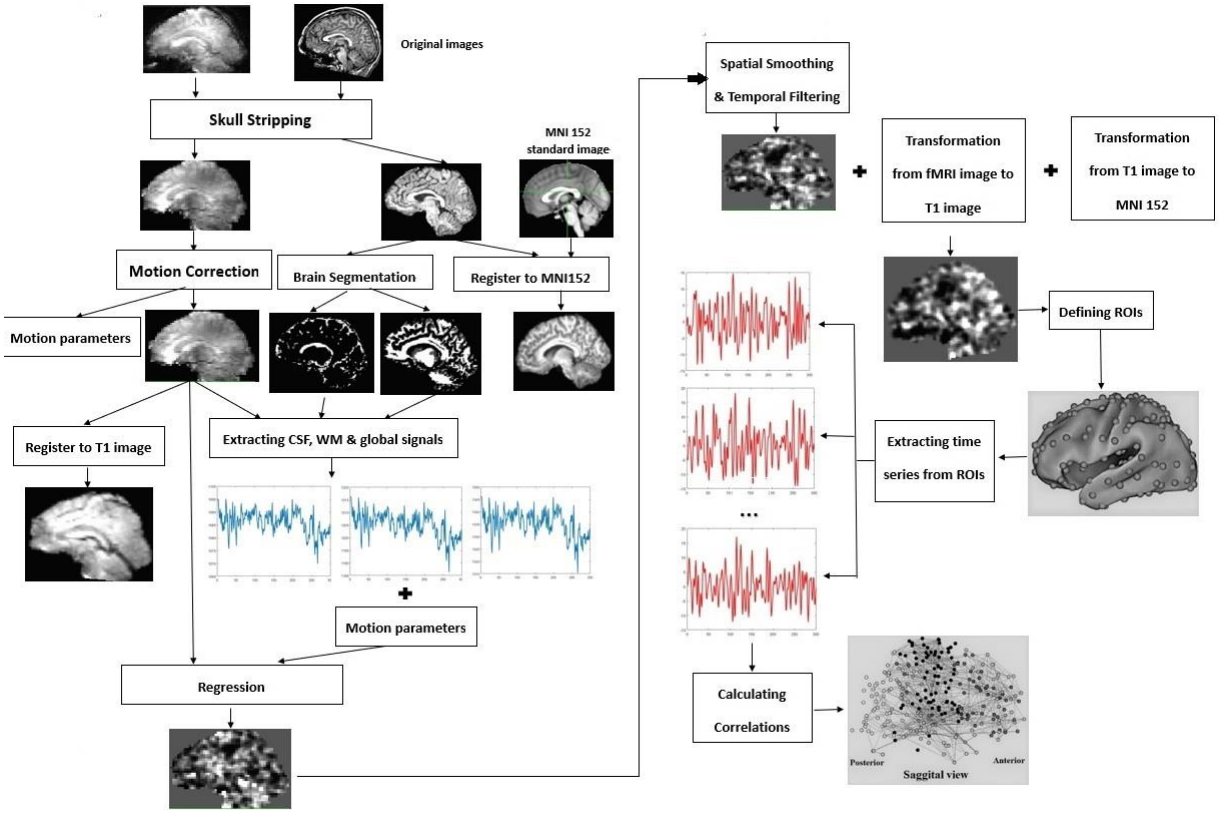

Figure 2.1 A flowchart showing the whole process for building FBNs. ROIs shown are defined in [17].

CHAPTER 3
JOINT SYMMETRICAL NON-NEGATIVE MATRIX FACTORIZATION FOR BRAIN NETWORK CLUSTERING

3.1 Introduction

My newly developed clustering algorithm, named joint symmetrical non-negative matrix factorization (JSNMF), is an NMF-based method able to cluster several networks with identical vertex set simultaneously.

NMF is an unsupervised machine learning method capable of extracting lower dimensional features [75]. Introduced in [76], original NMF method aims to solve an optimization problem

$$\min_{W,H} \|A - UZ\|_F^2 , \tag{3.1}$$

where $\|\cdot\|_F$ is Frobenius norm, A is the original feature matrix with each row being a feature vector of a object and each row of U is a feature vector with lower dimension than those in A. In this way, entries in A can be viewed as linear combinations of row vector in U using coefficients given in corresponding column of Z. Based on the feature matrix U, I can furtherly do the clustering using methods such as k-means. Following that, a variety of NMF methods has been developed and implemented successfully on many datasets [77]. For network clustering, [78] introduced an NMF-based algorithm to identify overlapping clusters and implemented it for social networks, and [79] presented another one doing the same job but also focusing on identification of hubs and outliers. In addition, [80] shows that NMF is efficient in finding protein complexes in protein-protein interaction networks.

Usually, the optimization of equation (3.1) cannot be solved analytically. Therefore, in most of cases, the updating rules are derived to solve them numerically. I should also notice that depends on the initialization of U and Z, a global minimum of (3.1) may not be reachable, but a local

one usually suffices.

Traditional NMF methods factorizes one matrix at a time, however, real-world dataset may contain multiple views, or attributes, of a set of objects, where each view delivers information from a different perspective. Traditional NMF methods may not deal with such situations well, since they can only learn from one view at a time and the views usually complement each other. Therefore, it would be more reasonable if I find a way to integrate information from all the views to cluster the objects. Recently, several multi-view clustering algorithms are developed [81, 82, 83, 84]. One subtype of multi-view clustering methods is to build a joint loss function that combines every view, then solve for a consensus indicating clustering information. [85] and [83] introduce two spectral-clustering-based methods and [81, 82, 84] proposed another one based on NMF.

In this study, since individual networks are obtained from single subjects, I assume each of them to be a view, representing a different organization of connectivity of the objects which are cortical ROIs. Previously developed multi-view methods are designed for general clustering cases where there are no constraints for input feature matrices, while brain networks are represented by symmetrical matrices. When factorizing symmetrical matrices, the problem can be formulated as follows

$$\min_{H} \|A - HH^T\|_F^2 \tag{3.2}$$

or

$$\min_{H,S} \|A - HSH^T\|_F^2 \tag{3.3}$$

where A is the adjacency matrix of a network and H is the feature matrix. According to [86], such factorization is equivalent to kernel k-means and spectral clustering.

In this chapter, I present an NMF-based multi-view network clustering method called JSNMF to find modules in functional brain networks. I will validate my clustering results and compare with other competing methods.

3.2 Methods

3.2.1 JSNMF

Similar to other NMF methods, JSNMF solves an optimization problem but with minimizing a different objective function

$$O_{JSNMF} = \sum_{v=1}^{n} \left\| A^{(v)} - HS^{(v)}H^T \right\|_F^2 + \alpha \sum_{i=1}^{N}\sum_{j=1}^{K} \left| H_{ij} \right|$$

$$s.t. H \geq 0, S^{(v)} \geq 0 \text{ for } v = 1, \dots, n \tag{3.4}$$

where $A^{(v)} \in R_{\geq 0}^{N \times N}, R_{\geq 0} = \{x \in R | x \geq 0\}$ denotes an single view in a dataset $A = \{A^{(1)}, \dots, A^{(n)}\}$, representing an individual brain network in my case, $H \in R_{\geq 0}^{N \times K}$ is the consensus I obtain from all the views as the feature matrix and cluster indicator and $S^{(v)} \in R_{\geq 0}^{K \times K}$, and α is a positive regularization factor to make sure H is sparse. K is the reduced dimension and also the number of clusters that I set in priori. Different from methods in [81, 82, 84], JSNMF is designed specifically for factorization of symmetrical matrix, and I use the form in (3.3) but not (3.2) because $S^{(v)}$ provides extra degrees for freedom so that I can get closer estimation of $A^{(v)}$.

Rewriting the Frobenius norm in Equation (3.4) into the trace form, I have

$$O_{JSNMF} = tr\left(\sum_{v=1}^{n} \left({A^{(v)}}^2 - 2A^{(v)}HS^{(v)}H^T + \left(HS^{(v)}H^T \right)^2 \right) \right) + \alpha \sum_{i=1}^{N}\sum_{j=1}^{K} \left| H_{ij} \right| , \tag{3.5}$$

where $tr(\cdot)$ denotes the trace of a matrix.

To minimize (3.5), I can employ the Lagrange multiplier and turn () into a new optimization problem without any constraints

$$O_L = tr\left(\sum_{v=1}^{n} \left({A^{(v)}}^2 - 2A^{(v)}HS^{(v)}H^T + \left(HS^{(v)}H^T \right)^2 \right) \right) + \alpha \sum_{i=1}^{N}\sum_{j=1}^{K} \left| H_{ij} \right| + tr(H^T\Lambda) , \tag{3.6}$$

where $\Lambda \in R^{N \times K}$ is Lagrange multipliers for H. To find minima of O_L, I can solve $\frac{\partial O_L}{\partial S^{(v)}} = 0$

and $\frac{\partial O_L}{\partial H} = 0$ for $S^{(v)}$ and H, respectively. Here,

$$\frac{\partial O_L}{\partial S^{(v)}} = -2H^T A^{(v)} H + 2H^T H S^{(v)} H^T H \tag{3.7}$$

$$\frac{\partial O_L}{\partial H} = \sum_{v=1}^{n} \left(-4A^{(v)} H S^{(v)} + 4HS^{(v)} H^T H S^{(v)}\right) + \Lambda + \alpha E \tag{3.8}$$

where $E = 1^{N \times K}$.

$\frac{\partial O_L}{\partial S^{(v)}} = 0$ can be directly solved and I have

$$S^{(v)} = (H^T H)^{-1} H^T A^{(v)} H (H^T H)^{-1}. \tag{3.9}$$

To find the solution of $\frac{\partial O}{\partial H} = 0$, I could use an updating rule

$$H_{ij} \leftarrow H_{ij} \left(\frac{4\left(\sum_{v=1}^{n} A^{(v)} H S^{(v)}\right)_{ij}}{\left(\alpha + 4\sum_{v=1}^{n} (HS^{(v)} H^T H S^{(v)})\right)_{ij}} \right)^{\frac{1}{4}}. \tag{3.10}$$

Proof of convergence of the updating rule is as follows.

For a function $F(H)$, function $Z(H, H')$ is an auxiliary function of $F(H)$ if $Z(H, H) = F(H)$ and $Z(H, H') \geq F(H)$. According to [75], $F(H)$ is non-increasing with the updating rule

$$H = arg \min_{H} Z(H, H') \tag{3.11}$$

because

$$F(H_0) = Z(H_0, H_0) \geq Z(H_1, H_0) \geq F(H_1) \ldots \tag{3.12}$$

if $H_1 = arg \min_{H} Z(H, H_0)$ and so on. In this case, given $O_{JSNMF}(H)$, its auxiliary function of can be defined as

$$Z(H,H')$$

$$= \sum_{v=1}^{n}\left(A^{(v)^2} - 2\sum_{ijkl} H'_{ij}S^{(v)}_{jk}H'_{lk}A^{(v)}_{li}\left(1+\log\frac{H_{ij}H_{lk}}{H'_{ij}H'_{lk}}\right) + \sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H^4_{ij}}{H'^3_{ij}}\right)$$
$$+\alpha\sum_{ij}\frac{H'_{ij}\left(H^4_{ij}+{H'_{ij}}^4\right)+2{H'_{ij}}^5}{4{H'_{ij}}^4}. \quad (3.13)$$

To prove $Z(H,H')$ is an auxiliary function of O_{JSNMF}, I can obviously see that $Z(H,H) = O_{JSNMF}(H)$. Thus, the following steps prove $Z(H,H') \geq F(H)$.

I separate $Z(H,H')$ into 3 terms, each corresponding to a term in $O_{JSNMF}(H)$. Since $A^{(v)^2}$ does not include H, I do not consider it. Firstly, since $\forall\, z>0, z \geq 1+\log z$, if I consider $z = \frac{H_{ij}H_{lk}}{H'_{ij}H'_{lk}}$, then

$$\sum_{ijkl} H'_{ij}S^{(v)}_{jk}H'_{lk}A^{(v)}_{li}\left(1+\log\frac{H_{ij}H_{lk}}{H'_{ij}H'_{lk}}\right) \leq \sum_{ijkl} H_{ij}S^{(v)}_{jk}H_{lk}A^{(v)}_{li} = tr\left(A^{(v)}HS^{(v)}H^T\right). \quad (3.14)$$

Secondly, following [87], if I consider $H_{ij} = \mu_{ij}H'_{ij}$, I have

$$\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H^4_{ij}}{H'^3_{ij}} = \sum_{ijkpqr} H'_{ik}S^{(v)}_{kp}H'_{qp}H'_{qr}S^{(v)}_{rj}H'_{ij}\mu^4_{ij}\ . \quad (3.15)$$

Switching indices, I have another three equations

$$\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H^4_{ij}}{H'^3_{ij}} = \sum_{ijkpqr} H'_{qp}S^{(v)}_{pk}H'_{ik}H'_{ij}S^{(v)}_{jr}H'_{qr}\mu^4_{qr} \quad (3.16)$$

$$\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H^4_{ij}}{H'^3_{ij}} = \sum_{ijkpqr} H'_{qr}S^{(v)}_{rj}H'_{ij}H'_{ik}S^{(v)}_{kp}H'_{qp}\mu^4_{qp} \quad (3.17)$$

$$\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H^4_{ij}}{H'^3_{ij}} = \sum_{ijkpqr} H'_{ij}S^{(v)}_{jr}H'_{qr}H'_{qp}S^{(v)}_{pk}H'_{ik}\mu^4_{ik}\ . \quad (3.18)$$

According to Equation (3.9), $S^{(v)}$ is always symmetrical. Thus, adding Equations (3.15) to (3.18) together, I get

$$\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H_{ij}^{4}}{H'^{3}_{ij}}=\sum_{ijkpqr}H'_{ik}S_{kp}^{(v)}H'_{qp}H'_{qr}S_{rj}^{(v)}H'_{ij}\frac{\mu_{ij}^{4}+\mu_{qr}^{4}+\mu_{qp}^{4}+\mu_{ik}^{4}}{4}\ . \quad (3.19)$$

Since $\forall a,b,c,d\geq 0, a^4+b^4+c^4+d^4\geq 4abcd$, I can easily prove

$$\begin{aligned}\sum_{ij}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H_{ij}^{4}}{H'^{3}_{ij}}&\geq\sum_{ijkpqr}H'_{ik}S_{kp}^{(v)}H'_{qp}H'_{qr}S_{rj}^{(v)}H'_{ij}\mu_{ij}\mu_{qr}\mu_{qp}\mu_{ik}\\&=\sum_{ijkpqr}H_{ik}S_{kp}^{(v)}H_{qp}H_{qr}S_{rj}^{(v)}H_{ij}=tr\left(\left(HS^{(v)}H^{T}\right)^{2}\right)\ .\end{aligned} \quad (3.20)$$

Thirdly, since $\forall\ a,b\geq 0, 2ab\leq a^2+b^2$, it can be proved that

$$\begin{aligned}\sum_{ij}\frac{H'_{ij}\left(H_{ij}^{4}+{H'_{ij}}^{4}\right)+2{H'_{ij}}^{5}}{5{H'_{ij}}^{4}}&=\sum_{ij}\frac{H'_{ij}\frac{\left(H_{ij}^{4}+{H'_{ij}}^{4}\right)}{{H'_{ij}}^{3}}+2{H'_{ij}}^{2}}{2H'_{ij}}\\&\geq\sum_{ij}\frac{H_{ij}^{2}+2{H'_{ij}}^{2}}{2H'_{ij}}\geq\sum_{ij}H_{ij}\ .\end{aligned} \quad (3.21)$$

Adding inequalities (3.14), (3.20) and (3.21) together, I can prove $Z(H,H')\geq O_{JSNMF}(H)$, thus $Z(H,H')$ is an auxiliary function of $O_{JSNMF}(H)$. $O_{JSNMF}(H)$ is non-increasing under the updating rule $H=arg\min_{H}Z(H,H')$. Since $O_{JSNMF}(H)$ has a lower bound of 0, it would converge eventually.

Now following [87], I should look for the solution to $arg\min_{H}Z(H,H')$ in the updating rule. To find minimum of $Z(H,H')$, I could take the first partial derivative of $Z(H,H')$ with respect to H

$$\frac{\partial Z(H,H')}{\partial H_{ij}}=\sum_{v=1}^{n}\left(-4\frac{H'_{ij}\left(A^{(v)}H'S^{(v)}\right)_{ij}}{H_{ij}}+4\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H_{ij}^{3}}{{H'_{ij}}^{3}}\right)+\alpha\frac{H_{ij}^{3}}{{H'_{ij}}^{3}}\ . \quad (3.22)$$

Then the Hessian matrix

$$\begin{aligned}\frac{\partial^{2}Z(H,H')}{\partial H_{ij}\partial H_{kl}}=\delta_{ik}\delta_{jl}\sum_{v=1}^{n}&\left(4\frac{H'_{ij}\left(A^{(v)}H'S^{(v)}\right)_{ij}}{H_{ij}^{2}}+12\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}\frac{H_{ij}^{2}}{H'^{3}_{ij}}\right)\\&+3\alpha\delta_{ik}\delta_{jl}\frac{H_{ij}^{2}}{{H'_{ij}}^{3}}\ ,\end{aligned} \quad (3.23)$$

where $\delta_{ab} = \begin{cases} 1, & a = b \\ 0, & otherwise \end{cases}$. The Hessian matrix is a diagonal matrix with all non-negative entries, thus it is positive semidefinite, which indicates that $Z(H, H')$ is a convex function. So in this case, $arg\min_H Z(H, H')$ can be found by solving $\frac{\partial Z(H,H')}{\partial H_{ij}} = 0$ and I get

$$H_{ij} = H'_{ij}\left(\frac{4\sum_{v=1}^{n}\left(A^{(v)}H'S^{(v)}\right)_{ij}}{\alpha + 4\sum_{v=1}^{n}\left(H'S^{(v)}H'^{T}H'S^{(v)}\right)_{ij}}\right)^{\frac{1}{4}}, \tag{3.24}$$

which is the same as the updating rule present in Equation (3.10).

Cluster indicator H can help determine which cluster a vertex belongs to. If H_{ij} is the maximum value in ith row of H, then vertex i will be assigned to cluster j. In my processing, each column of H is also normalized so that maximum values of columns are all one. This is to balance the numbers of vertices in each cluster in case sizes of certain clusters are too large or too small.

3.2.2 Evaluating Indices for Clustering Results

To measure the performance of my method in functional brain network clustering, I adopt 3 indices, modularity, conductance and coverage, to quantify the quality of the clustering results.

Generally, in previous studies, most indices developed to evaluate a clustering can be categorized into two kinds: external indices and internal indices. External indices use external information to decide whether a clustering is proper. Commonly used external indices, such as normalized mutual information (NMI) [88] or adjusted rand index (ARI) [89], compare the similarity between two clusterings, one of which is to be evaluated and the other one is a standard. Obviously, the more similar two clusterings are, the better a clustering method is. Such external indices, however, are not suitable in my case. Although there are several studies about functional module of brains [17, 21], there is no widely accepted standard for partitioning the whole-brain networks, especially in a functional way. In the following sections for evaluating my clustering results, I will introduce some identified functional modules as references to demonstrate that my clustering actually has biological meanings but here, to

evaluate the quality of clustering, I turn to some internal indices which require no external information. There are various internal indices developed to evaluate clustering, such as Silhouette index [90] or Dunn index [91], but these are designed for general clustering cases where the objects are points in Euclidean space. For network clustering, I have several indices specifically developed and implemented on my clustering results, which are modularity, conductance and coverage. In the following section, I will introduce the definitions of these 3 indices and what exactly each of them measures to quantify the clustering quality.

3.2.2.1 Coverage

Given a network with an $N \times N$ symmetrical adjacency matrix A, and a clustering with clusters $C_1, \dots, C_K$, where C_k is a vertex set containing all vertices in cluster k, coverage is defined as

$$Cov(C_1, \dots, C_K) = \sum_{k=1}^{K} W_{kk} \tag{3.25}$$

where W_{ij} is the fraction of edges end in cluster C_i and C_j over all edges. Coverage index measures the fraction of intra-cluster edges. Since I consider a good cluster should be highly connected internally and sparsely connected to the rest part of the network, intra-cluster edges should weight a large fraction over all edge weight. This index ranges from 0 to 1 and 1 is optimal. Coverage is a relatively simple method to evaluate a clustering, however, it has flawed in two aspects. Firstly, when all vertices of a network are clustered into a single cluster, coverage will be 1, but usually a clustering result like this is trivial. Secondly, it is incapable if I compare two clustering results with different numbers of clusters because naturally, this index decreases with increased number of clusters.

3.2.2.2 Modularity

To fix the problem coverage index has, I can define the modularity of the clustering as

$$Mod(C_1, \dots, C_K) = \sum_{k=1}^{K} \left(W_{kk} - \left(\sum_j W_{kj} \right)^2 \right) \tag{3.26}$$

Modularity measures the actual number of edges inside a cluster minus expected number of edges if they are randomly assigned to a network with same degree distribution. Here, if I

consider a network G' with the same node set as G and each node has the same degree while the edges are assigned randomly, then $\left(\sum_j W_{kj}\right)^2$ represents the expected edge weight between node i and j in G'. This index fixes the problems mentioned before of coverage index and gets rid of the possibility that intra-cluster edges exist by randomness. Modularity ranges from 0 to 1 and 1 is optimal.

3.2.2.3 Conductance

Conductance of a clustering $C_1, \dots, C_K$ on network G can be defined as

$$Con(C_1, \dots, C_K) = 1 - \frac{1}{K}\sum_{k=1}^{K} \frac{\sum_{i\in C_k, j\notin C_k} A_{ij}}{\min\left(\sum_{i\in C_k, j\in V} A_{ij}, \sum_{i\notin C_k, j\in V} A_{ij}\right)}, \tag{3.27}$$

where V denotes the set containing all vertices in the network. The second term in the conductance equation measures the average possibility of a one-step random walk entering or leaving a cluster. Given a network and a starting vertex, I randomly choose a neighbor of the starting vertex and move to it, then I again choose a neighbor of current vertex at random and move to it. Repeat the process and the sequence of vertices I have is a random walk on the network. A good cluster should be difficult for a random walk starting inside a cluster to move outside or starting outside a cluster to move inside. Therefore, the larger the conductance is, the better the cluster is. Conductance ranges from 0 to 1 and 1 is optimal. For cases with weighted edges, I assume a random walk is more likely to go through edges with higher weights.

3.2.3 Competing Methods

To demonstrate the performance of the JSNMF method that I developed, I compare it to some other competing methods also focusing on feature extraction and lowering dimension, including symmetrical non-negative matrix factorization (SNMF), spectral clustering (SC) and multi-view spectral clustering (MSC), which will be introduced in the following sections.

3.2.3.1 Symmetrical non-negative matrix factorization

I compare my method with a traditional SNMF method minimizing the objective function $O_{SNMF} = \|A - HH^T\|_F^2$, where A is the symmetrical adjacency matrix of a network and H

serves as the feature matrix and cluster indicator. Since this method can only be implemented on a single network, I use it to cluster the average network I have. Similar to JSNMF, SNMF also requires an updating rule to calculate H and make sure O_{SNMF} converges. Numerous methods have been introduced to do this [92], so here, I adopt a recently developed one [93] which guarantees convergence to a stationary points with smaller O_{SNMF} value. The updating rule is as follow.

$$H_{ij} \leftarrow \begin{cases} \sqrt{(A - HH^T)_{ii}}, D + (H^T H)_{jj} = 0 \\ \max\left(0, H'_{ij}\right), otherwise \end{cases} \tag{3.28}$$

where

$$H'_{ij} = H_{ij} - \frac{(HH^T H - AH)_{ij}}{2\left((H^T H)_{jj} + B\right)}$$

$$B = \max\left(0, -(A - HH^T)_{ii} + X_{ij}^2 + 2X_{ij}\left|\frac{(HH^T H - AH)_{ij}}{(H^T H)_{jj}}\right| + \frac{1}{2}\left|\frac{(HH^T H - AH)_{ij}}{(H^T H)_{jj}}\right|^2\right) \tag{3.29}$$

I use the same strategy on H as in JSNMF that I assign a vertex to a cluster where the entry value reaches the maximum in the corresponding column.

3.2.3.2 Spectral clustering

SC is another method aiming to extract feature matrices from network adjacency matrices with lower dimension. The general idea of SC is using eigenvectors of adjacency matrices. If I consider h_i to be the feature vector for vertex i in the network, then I can have an optimization problem minimizing $O_{SC} = \frac{1}{2}\sum_{ij} A_{ij}\left\|h_i - h_j\right\|^2$ s.t. $x_i^T x_j = \begin{cases} 1, & i = j \\ 0, & otherwise \end{cases}$. The function is built upon the assumption that if vertex i and j are close in the network, or in another word, they are connected by a highly weighted edge, then their feature vectors should not be far away in Euclidean space. It has been proved that $O_{SC} = tr(H^T LH)$, where $H = [h_1\ h_2\ ...\ h_n]^T$ is the cluster indicator and $L = D - A$ is the unnormalized Laplacian matrix where D is a diagonal matrix with degrees of nodes on its diagonal. Also, previous research has proved that if the feature vectors are desired to be p-dimensional, then H should be the p eigenvectors of L corresponding to the least p eigenvalues. On the feature matrix H, I can

use clustering algorithms such as k-means to identify modules. In this study, I employ an algorithm introduced in [94] that discretizes the feature matrix into a binary one. In the discretized matrix, each row only has one non-zero element indicating which cluster this vertex belongs to.

3.2.3.3 Multi-view spectral clustering

This method, introduced in [83], maximizes the objective function

$$\sum_{v=1}^{n_v} tr\left(\left(Q^{(v)}\right)^T L^{(v)} Q^{(v)}\right) + \sum_{v=1}^{n_v} \lambda_v tr\left(Q^{(v)}\left(Q^{(v)}\right)^T HH^T\right)$$
$$s.t.\left(Q^{(v)}\right)^T Q^{(v)} = I, H^T H = I, \tag{3.30}$$

where $L^{(v)}$ is the Laplacian matrix of view v. By solving a standard spectral problem with Laplacian $L^{(v)} + HH^T$ for $Q^{(v)}$ and another spectral problem with Laplacian $\sum_v \lambda_v Q^{(v)}\left(Q^{(v)}\right)^T$ for H, $Q^{(v)}$ and H can be iteratively updated and the problem could be solved.

3.3 Results and Discussions

Firstly, I implemented my method, JSNMF, on the dataset described in Chapter 2 with different settings of parameters and edge weight thresholds to find the best performance. For clustering purpose, I only use TD networks for better performances since ASD may alter the modular organization of FBNs [95]. According to Equation (3.4), there are two parameters, K as the number of clusters and α as the regularization factor. I try different combinations of K and α where K ranges from 2 to 10 and α ranges from 0.0001 to 1000. I use modularity to measure the performances of clusterings. Although I introduced three indices in the last section to evaluate clustering, here I choose modularity as a major one since it is the most widely used. In JSNMF and SNMF, the cluster indicator matrices are randomly initiated then updated iteratively until convergence, so it is likely that the results are different with different initiations. Therefore, for each setting of parameters, I run my programs for JSNMF and SNMF for 40

times, each, then applies results from each running to every individual brain network to calculate the indices. Although SNMF is implemented on the average network, I still measure its performance on individual networks since I believe the clusterings are supposed to be valid on each of them. Indices calculated on each running and each network are averaged to obtain a score on a setting of parameters.

As shown in Figure 3.1a, when I choose 0.35 as the edge weight threshold, the modularity reaches the largest when $K = 4$, i.e., it is most reasonable if I cluster my networks into 4 modules. I filter the edges because it will amplify the small-worldness of the networks. In the following I try several other thresholds under 0.35 because 0.35 is the maximal threshold to keep all networks connected. If I zoom in like shown in Figure 3.1b, I can see that when $\alpha = 1$, modularity is the highest among all the curves. This is true if I choose another edge weight threshold under 0.35. Therefore, I choose $K = 4$ and $\alpha = 1$ and under this setting of parameters, the clustering results are presented in Figure 3.2.

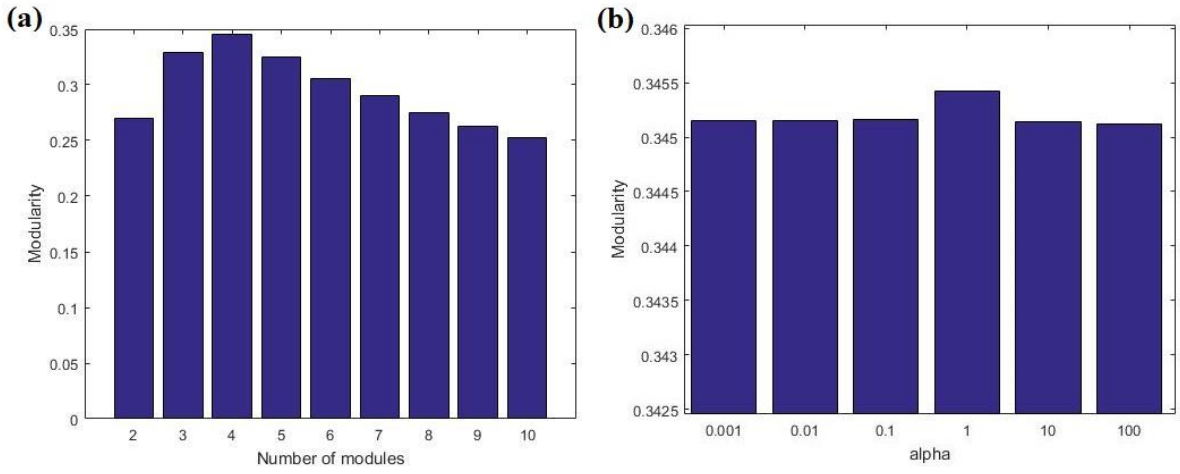

Figure 3.1 Modularity of clusterings obtained with different settings of parameters

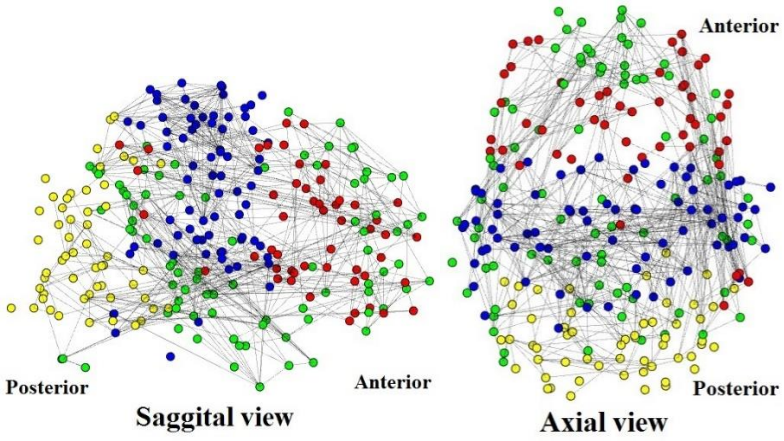

Figure 3.2 Clustering results of JSNMF. Only edges weighted larger than 0.8 are shown. I choose one from 40 runs that is most similar to all others.

Next, I compare my method with other competing methods introduced in Section 3.2.3 using indices: modularity, conductance and coverage. I measure mean values and standard deviations over all individual networks for each method. The reason I compare standard deviation is that since I believe different subjects have similar functional modules, the indices of individual networks should not disperse much. I set $K = 4$ for each method and calculate the indices using different edge weight thresholds.

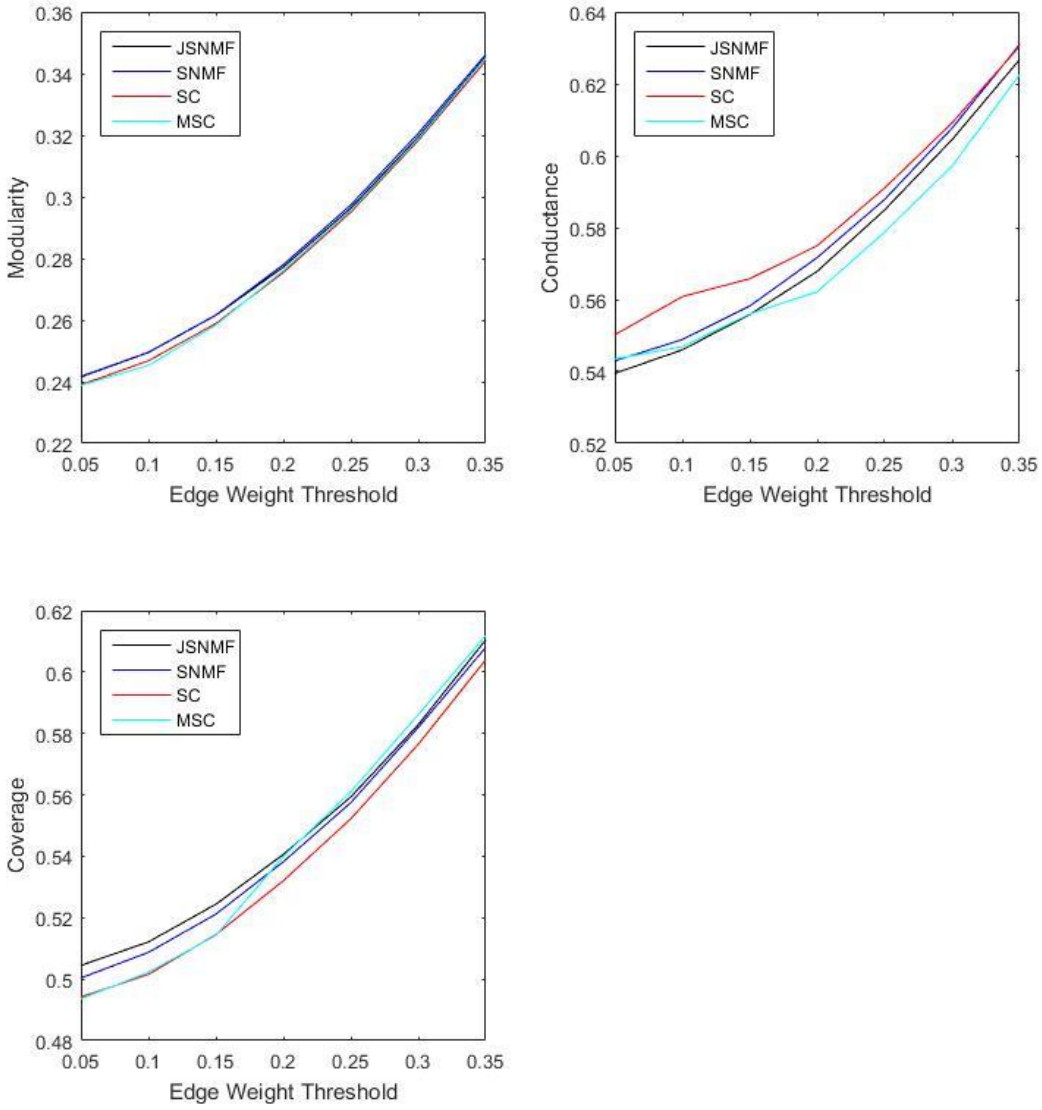

Figure 3.3 Comparison among clustering methods on the mean value of indices over all individual networks.

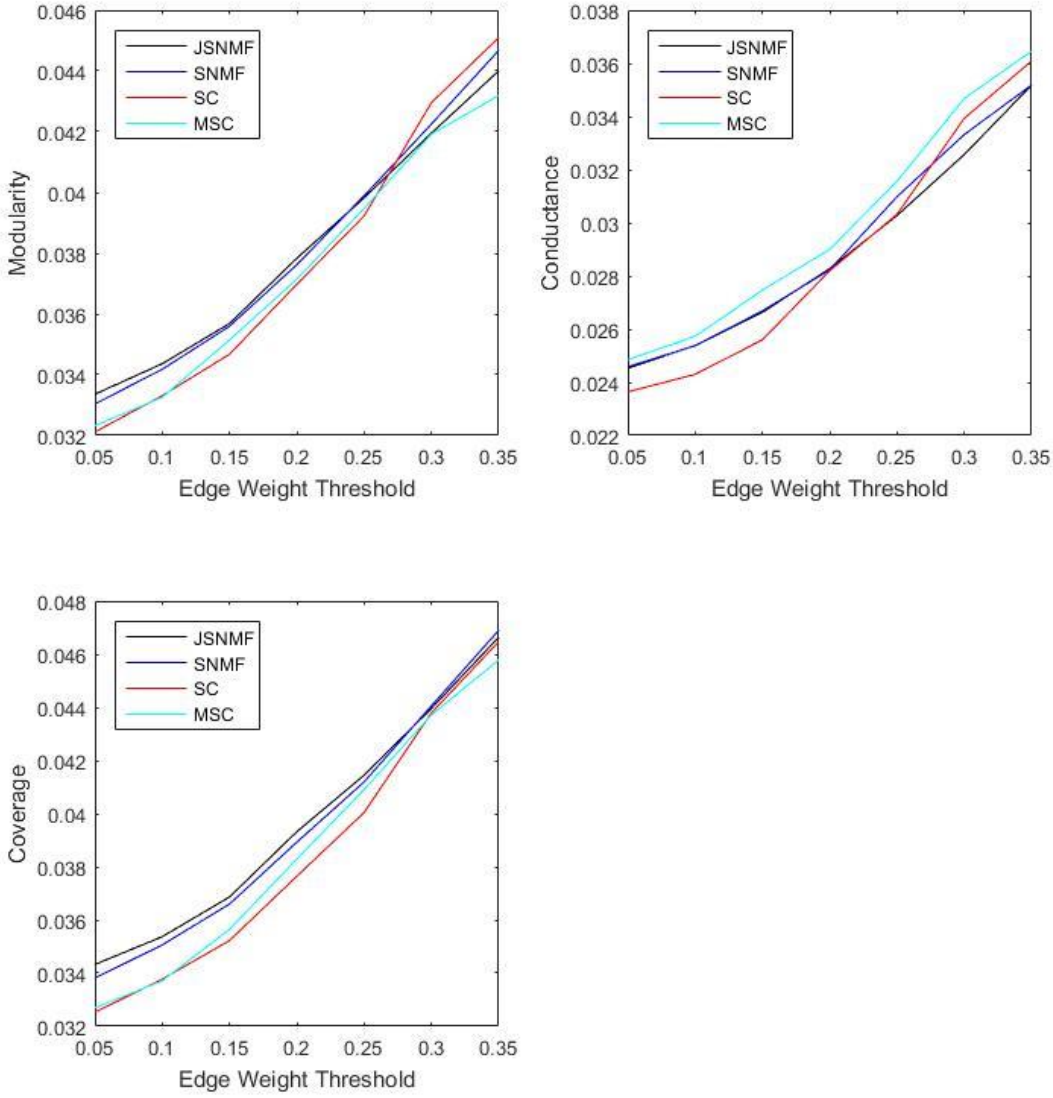

Figure 3.4 Comparison among clustering methods on the standard deviation of indices over all individual networks.

The results of mean values are shown in Figure 3.3. I can see that modularity values obtained from 4 methods are relatively close, my method still outperforms MS and MSC under all edge weight thresholds, and very close to SNMF. Coverage values of JSNMF are higher than other method under lower thresholds from 0.05 to 0.2, and still rank the second when the thresholds are larger. Although my method has lower modularity than NMF, I still believe my method is very promising, because judging from Figure 3.4 showing results of standard deviations, JSNMF has lower standard deviation of all three indices than NMF, especially under higher

thresholds. Since the three indices measure validity of a clustering on a network, lower standard deviation indicates that clustering from JSNMF is more equally valid on every individual network, but not biased toward some specific ones.

Previously, I analyze the validity of the clustering from my method using some external indices. Here, I demonstrate that the modules I identified are also valid from a neurological perspective, i.e., I will analyze components of each module and explain their exact functions. Since my method is randomly initialized and implemented on networks with various sparsity, which could yield numbers of different clusterings, I test the similarities of each pair of the clusterings using ARI. I find that the clusterings are relatively similar to each other that the lowest ARI value among all pairs are 0.91 which is still high considering ARI ranges from -1 to 1. Thus, from all clusterings, I choose one that is most similar to all other ones, i.e., if I calculate the mean value of ARI of a clustering with all other ones, I select the one with highest mean value to do further analyses.

The red module in Figure 3.2 mostly expands in middle frontal gyrus and frontal pole, partly in supramarginal gyrus. It contains several core regions previously identified as parts of frontal-parietal network (FPN), including intraparietal sulcus, frontal eye field or dorsolateral prefrontal cortex [96, 97]. FPN is an attention control system. Frontal eye field and intraparietal sulcus are reported to be activated in experiments where visual attentions of subjects are frequently drawn to different targets [96]. The dorsolateral prefrontal cortex is also reported highly connected to frontal eye field and intraparietal sulcus neurologically. Therefore, I label the red module as Frontal-parietal system.

The green module contains most of ROIs in middle and inferior temporal gyrus, cingulate gyrus, hippocampal gyrus, part of frontal gyrus and some other surrounding regions. I label this module as default mode network (DMN). Introduced in [98], the reason this system is called default mode is that it is more activated when brains are in resting state or mind-wondering. It is reported that default mode system has various functions such as emotion and memory [99]. Core regions of this system include medial prefrontal cortex, posterior cingulate cortex and hippocampal formation which are all contained in the green module in my partitioning [100].

Also, there are reports about participation of temporal lobule in default mode system on certain occasions [100].

The yellow module mainly covers the occipital lobule. I label this module as visual system because it is widely reported that occipital lobules are related to visual functions [101, 102, 103]. And blue module is labeled as somatosensory system. Blue module mostly located at parietal lobule. In [104], it is reported parietal lobule, especially its inferior part, is evoked in a series of tests where fingers of subjects are continuously stimulated.

3.4 Conclusion

In this chapter, I propose a new network clustering algorithm named JSNMF for identifying functional modules in FBNs. I illustrate that my algorithm outperforms other competing methods from certain perspectives and show the neurological validity of my modules. Although I must admit that the promotion of my method in modularity is rather trivial, in the next chapter, I will show that contains more significant alterations in modules I identify than other methods.

CHAPTER 4
STATISTICAL ANALYSIS OF MODULAR ALTERATIONS CAUSED BY ASD

4.1 Introduction

In the last chapter, I implement a new network clustering algorithm on the FBNs and obtain 4 functional modules. Since the major objective of this research is to develop a feature selection strategy that extracting features from a module, I still need to decide from which module I extract features. Therefore, in this chapter, I focus on evaluating alterations in modules. If a module shows most statistically significant alterations, then I believe features from this module are supposed to be more discriminant than others.

To find the module with most significant alterations, generally, I calculate some modular indices which describe the properties of modules from different views and use t-test to see if there are differences between ASD and TD group in the mean values of the indices. I also calculate some nodal indices and see if significantly altered vertices are enriched in certain modules. In the following sections, I will introduce the modular indices and t-test, as well as the results indicating from which modules I extract features.

Statistic tests are widely adopted in many studies about identifying alterations in brain networks caused by brain diseases. Most of such studies vary in methodologies but contribute to better understanding of pathophysiology of diseases. [29] calculates some indices measuring small-worldness and regional characteristics of structural brain networks, such as clustering coefficient, shortest path length and nodal efficiency, and uses linear regression to find significant alterations between AD and control group. They link alterations in structural connectivity with abnormal behaviors of AD patients. [28]and [26] obtain similar indices as [29]. Focusing on depression and mild cognitive impairment (MCI), [26] uses analysis of

variance (ANOVA) for difference test and provides evidence for associations between the two diseases. [27] uses t-test on regional connectivity and identifies potential biomarkers distinguishing AD subjects. For ASD studies, [22] and [105] identify alterations in long-range and short-range functional connections, indicating a deficit information integration system in ASD subjects. [106] and [14] use fMRI and DWI data respectively and locate disruptions of network organization in several cortical regions related to social cognitive functions. In [25] , comparison of voxelwise structural connectivity is made through analysis of covariance (ANCOVA). [23] connects strength of global functional connectivity with severity of ASD subjects and lowered fusiform-amygdala connectivity may also contribute to social impairment.

The indices calculated for alteration test in most papers are global or nodal ones, measuring the properties of the whole network or a small region. In this chapter, I adopt some modular indices. I believe that alterations in functional modules may provide clearer connections between cognitive functions and symptoms of ASD.

In this chapter, I also compare my network clustering algorithm, JSNMF, with other competing ones on how significant modular indices altered and how significantly altered vertices enriched in modules identified by different algorithms.

4.2 Methods

4.2.1 T-test

In this chapter, I use two-sample t-test to compare the ASD and TD networks to see if there are significant differences between ASD and TD populations. T-test tests if a sample follows a t-distribution and it is commonly used to determine if a population mean value equals to a specified one (one-sample t-test) or if two population has identical mean values. For each node in an individual network, I use the metrics introduced above to measure its nodal properties. Since this research focuses on modules, I calculate the mean values of metrics over all vertex is the same module. As a result, for each metric and each node in each network, I obtain a value measuring the property of the module corresponding to the metric. If I assemble the values from

the same vertex, corresponding to the same metric, but from different networks, I will obtain two samples J_{TD} and J_{ASD} from TD and ASD networks, representing TD and ASD populations, respectively. I assume both populations follow normal distributions. The dataset I use contains 42 ASD networks and 37 TD networks, so the two samples size 42 and 37. Here, I use two-sample t-test with unequal sample sizes and assume that two populations have unequal variance. Given J_{ASD} and J_{TD}, the t-score can be calculated as

$$t = \frac{\bar{J}_{ASD} - \bar{J}_{TD}}{\sqrt{\frac{\sigma_{ASD}^2}{|J_{ASD}|} + \frac{\sigma_{TD}^2}{|J_{TD}|}}} \tag{4.1}$$

where $|J|$ is the sample size, $\bar{J}$ is the sample mean and σ is the standard deviation. The t-score follows a t-distribution with degree of freedom calculated by Welch-Satterthwaite equation

$$degree\ of\ freedom = \frac{\left(\frac{\sigma_{ASD}^2}{|J_{ASD}|} + \frac{\sigma_{TD}^2}{|J_{TD}|}\right)^2}{\frac{\left(\frac{\sigma_{ASD}^2}{|J_{ASD}|}\right)^2}{|J_{ASD}| - 1} + \frac{\left(\frac{\sigma_{TD}^2}{|J_{TD}|}\right)^2}{|J_{TD}| - 1}}. \tag{4.2}$$

In the t-test, the null hypothesis is that J_{ASD} and J_{TD} come from populations with equal means, while the alternative hypothesis is that the means are unequal. Given the t-score, the p-value to determine whether to reject null hypothesis is the area under density function curve of t-distribution to the right of t and left of $-t$, since it is a two-sided test.

4.2.2 Graph Theory Based Metrics

In this chapter, I use some metrics to measure the nodal properties of a vertex in a network, then furtherly calculate modular properties based on them. Thus, in this section, I introduce these metrics.

4.2.2.1 Degree centrality

Given the adjacency matrix A of a network with n vertices, the degree centrality of vertex i is defined as

$$DC(i) = \sum_{j=1}^{n} A_{ij} . \tag{4.3}$$

Generally, degree centrality is the total number of edges (in unweighted networks) or summation of weights of edges (in weighted networks) connected to the vertex. Degree centrality is the simplest metric to measure the connectivity of a vertex.

4.2.2.2 Closeness centrality

Closeness centrality of i is defined as reciprocal of the sum of distance between i and any other vertices and it is calculated as

$$CC(i) = \frac{1}{\sum_{j \neq i} d(i,j)}, \tag{4.4}$$

where $d(i,j)$ represents the distance between i and j in the network. The distance is defined as the length of shortest path. If vertex i is closer to other vertices, it means information from i can easily reach the rest part of the network, assuming information only flows on shortest paths. Closeness centrality is also used to measure the efficiency of a vertex.

4.2.2.3 Clustering coefficient

Clustering coefficient of a vertex i in an unweighted network can be calculated by

$$CC(i) = \frac{2Tri_i}{DC(i)(DC(i)-1)} \tag{4.5}$$

where Tri_i is the number of triangles attached to vertex i. A triangle in a network is a structure that three vertices are connected to each other. Another equivalent definition of clustering coefficient is that number of edges among its neighbors of vertex i over number all possible edges among them. So clustering coefficient measures how close a vertex and its neighbors being a clique, or the degree of a node to which it tends to participate in a cluster. Several modifications of Equation (4.5) have been made for weighted networks [107]. In this research, I adopt one presented in [108] based on intensity of subnetwork and it is defined as

$$CC_w(i) = \frac{2}{DC(i)(DC(i)-1)} \sum_{j,k} \frac{\left(A_{ij}A_{jk}A_{ik}\right)^{\frac{1}{3}}}{\max A} \tag{4.6}$$

where vertex j and k are neighbors to i. I should notice that $DC(i)$ in weighted clustering coefficient is still calculated in unweighted network. $\sum_{j,k}\frac{(A_{ij}A_{jk}A_{ik})^{1/3}}{\max A}$ is the intensity of the triangle to replace Tri_i. Dividing $\max A$ is to normalize the edge weights so that the product will not be too large or too small.

4.2.2.4 Current-flow closeness centrality

In closeness centrality, I assume information only flows on the shortest paths, however, spread of information onto other paths may also take place in real-world networks. Current-flow centralities borrow terms of electric current and voltage from electric circuits as an alternative way to measure information flow and distances among vertices. Current-flow closeness centrality of vertex i is defined as

$$CFCC(i) = \frac{1}{\sum_{j\neq i} Vol(i) - Vol(j)}, \tag{4.7}$$

where $Vol(i) - Vol(j)$ is the voltage difference between i and j. Redefining the distance as voltage guarantees the information is not necessarily on the shortest paths. I should notice that here, I always consider i as the source vertex of the current so $Vol(i) - Vol(j)$ will always be positive. [109] provides an efficient algorithm to calculate such centrality.

4.2.3 F-score

Given a set of vertices which show significant difference between ASD and TD networks, I may wonder if the modules I identify are enriched with these vertices. If a module contains many significant vertices, I can also consider the module as a significant one. To measure the enrichment, I here introduce the f-score. Given a vertex set C of a module, and the vertex set P that contains all significantly altered vertices in the network, f-score is defined as

$$fs = \frac{2|C \cap P|}{|C| + |P|}. \tag{4.8}$$

The f-score ranges from 0 and the maximum depends on the actual size of $C \cap P$ but it should be no larger than 1.

4.2 Results and Discussions

I use metrics introduced before to measure nodal properties. I are using weighted networks in all calculation. For closeness centrality, I use $1 - A_{ij}$ to measure the distances of neighboring vertices. And for current-flow centrality, edge weights are regarded as flow capacities which serves as an upper limit of the amount of a flow that can pass through.

To identify alterations in ASD networks, I implement t-test on my dataset in two ways, vertex-wise and module-wise. Vertex-wise t-test means, for each vertex, I can build a sample for ASD networks and TD networks and do the t-test to see if there are alterations on this vertex. Module-wise t-test means I take an average over all vertices in a module to calculate a metric value for it, then do the t-test on the ASD and TD samples of the module. The significant level is set to 0.05 for all t-tests. After the vertex-wise t-test, I use f-score to measure if a module is enriched with altered vertices. I believe if a module contains many altered vertices, I can consider it as an altered module. Also, I calculate p-values using module-wise t-test on each module to see if it shows significant differences with respect to the metrics.

The results of f-score is shown in Figure 4.1. The 4 modules are obtained from JSNMF. I can see that for the 4 metrics I measure, frontal-parietal control system and default mode system have higher f-score in most cases, indicating they contain more significantly altered vertices. These two modules also show lower p-values compared with other modules according to Figure 4.2. These results indicate that frontal-parietal control system and default mode system may be altered more.

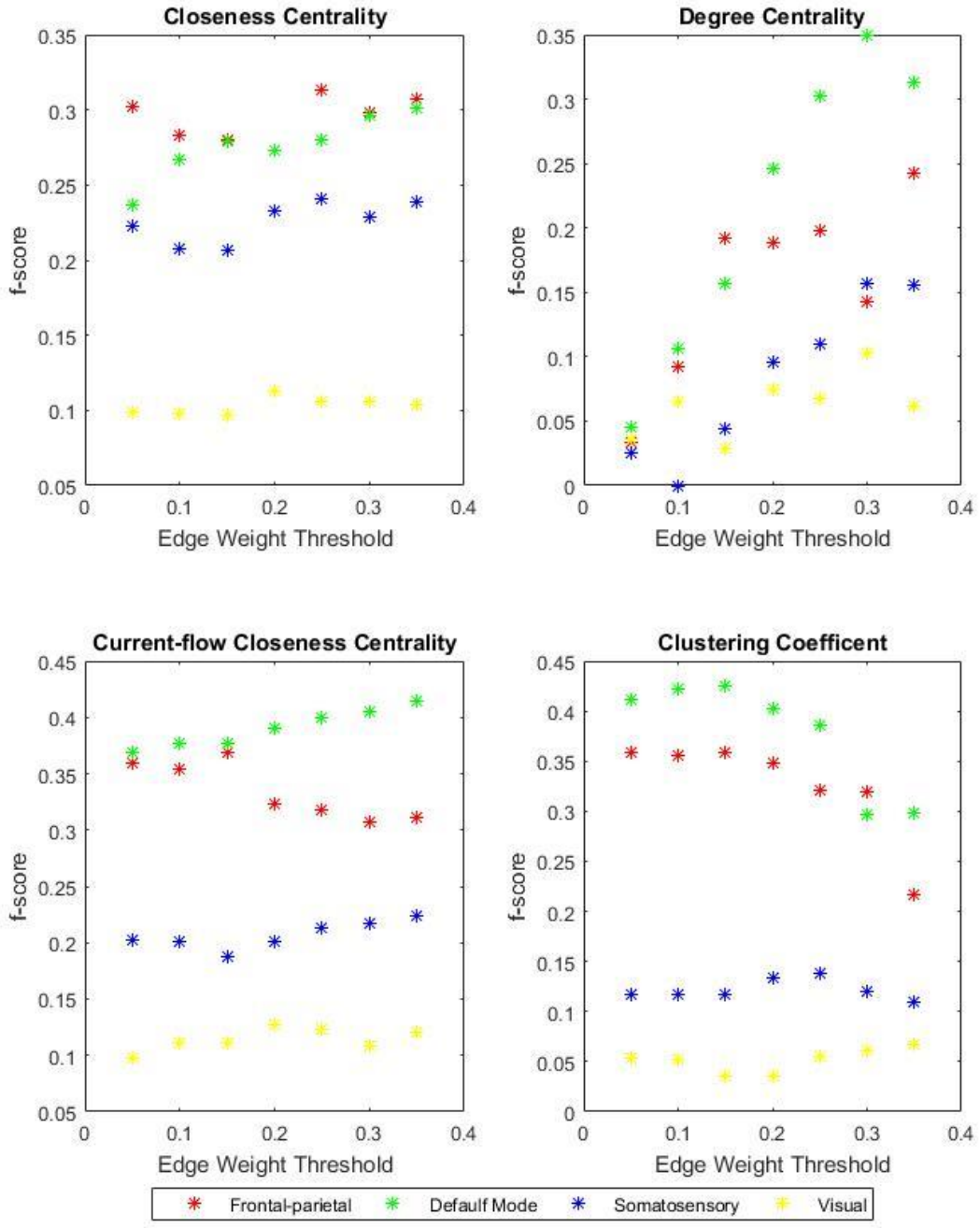

Figure 4.1 Comparison of f-scores among 4 modules.

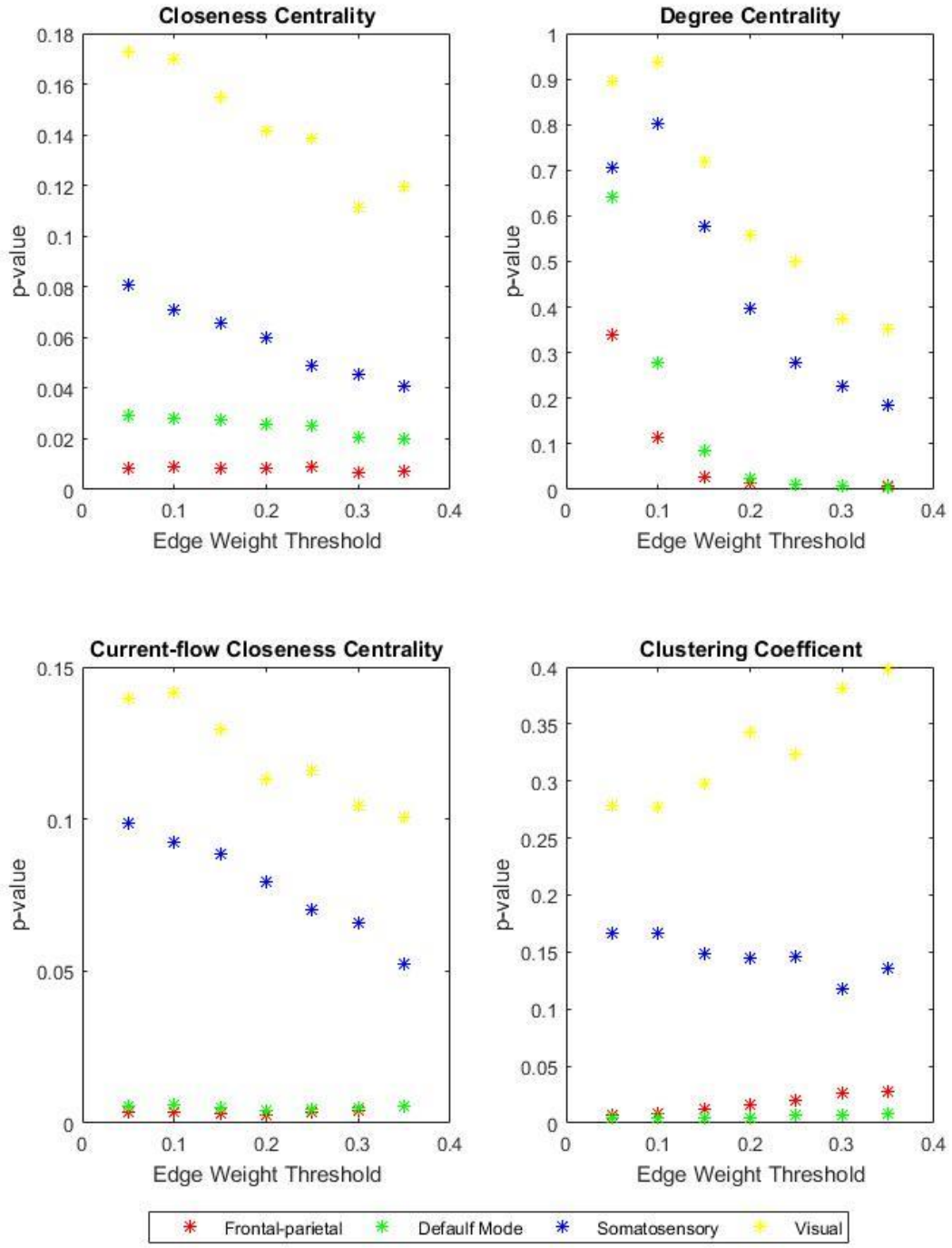

Figure 4.2 Comparison of p-values among 4 modules.

Altered degree centrality and clustering coefficient in default mode system of ASD network indicates alterations of functional connectivity inside the module, which is consistent with several researches [110, 111, 100, 95]. Such connectivity alterations can be a result of altered white matter volume uncovered on subjects with ASD [112, 113]. I also find alterations in

closeness centrality and current-flow closeness centrality, which indicates possible alterations of long-range connections between default mode system and other modules, since these two metrics measure the total distance from a vertex to the rest part of the network. Similar findings are also reported in [22] but here I use a different approach considering a whole module simultaneously to demonstrate the impact of ASD on the corresponding function. Several core functions of the default mode system, including understanding emotions and thoughts of others, are reported as major deficits of ASD [114]. Alterations in these indices can be an explanation to devastation of functions of default mode system.

Similar alterations of connectivity are also found in FPN, both intra-module and long-ranged, which are supported by several previous findings [115]. Also, visual attention related symptom is also reported which may be related to the connectivity loss in this module [116, 117]. Recent studies also hypothesize a possible controlling interaction from FPN to DMN [118, 119], which explains the co-occurrence of significant alterations in both DMN and FPN.

Previously, I have demonstrated that it is reasonable that I find reduced metrics value in two modules. Although I use a clustering from JSNMF to do that but results of p-values and f-scores from other clustering methods are relatively the same. So naturally, I can derive another strategy to compare these methods. If DMN identified through a method reach higher f-scores and lower p-values, i.e., show more significant alterations, then I can consider the method to be a better one. The results are presented in Figure 4.3 and Figure 4.4.

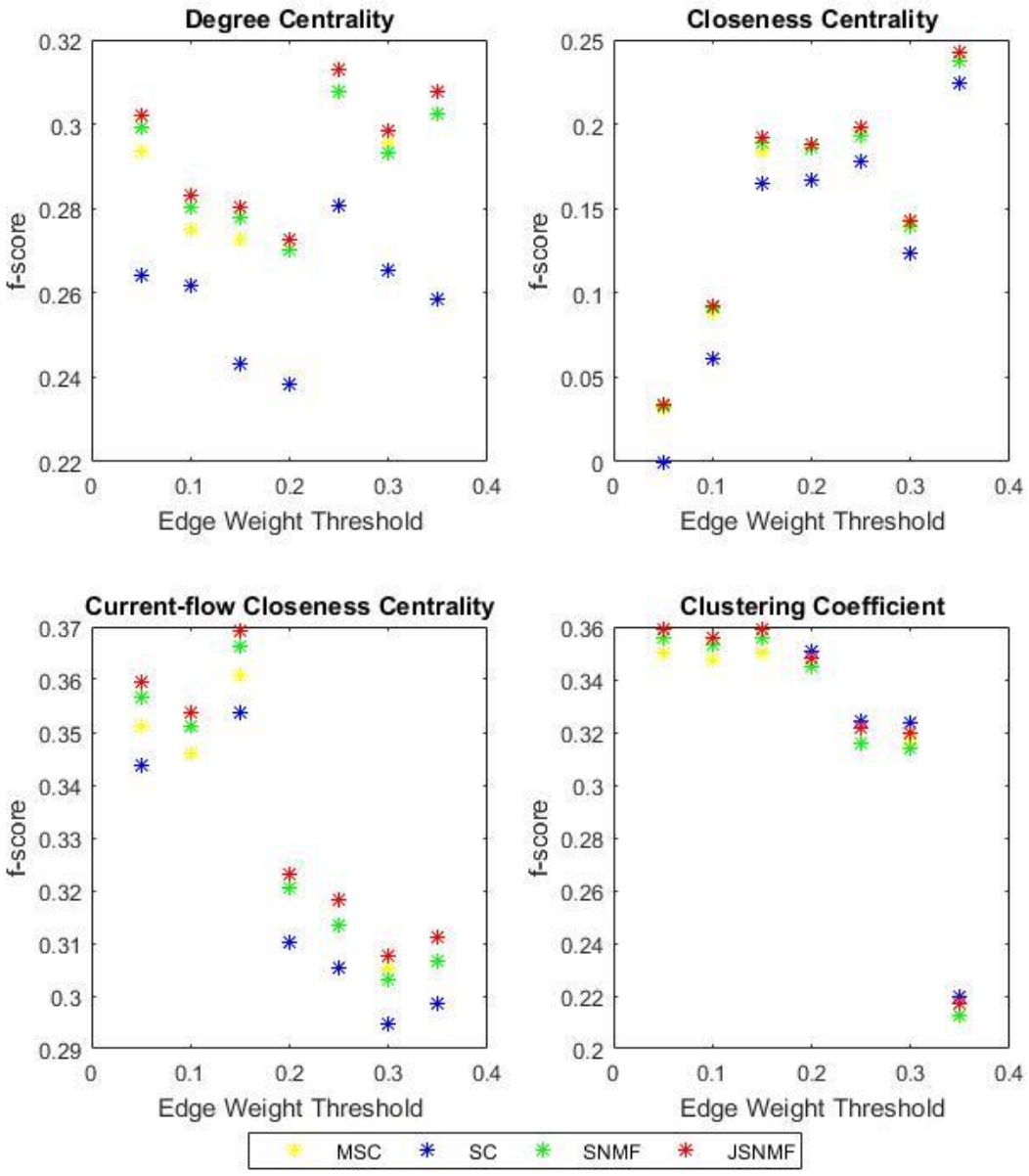

Figure 4.3 Comparison of f-scores among 4 clustering methods on DMN

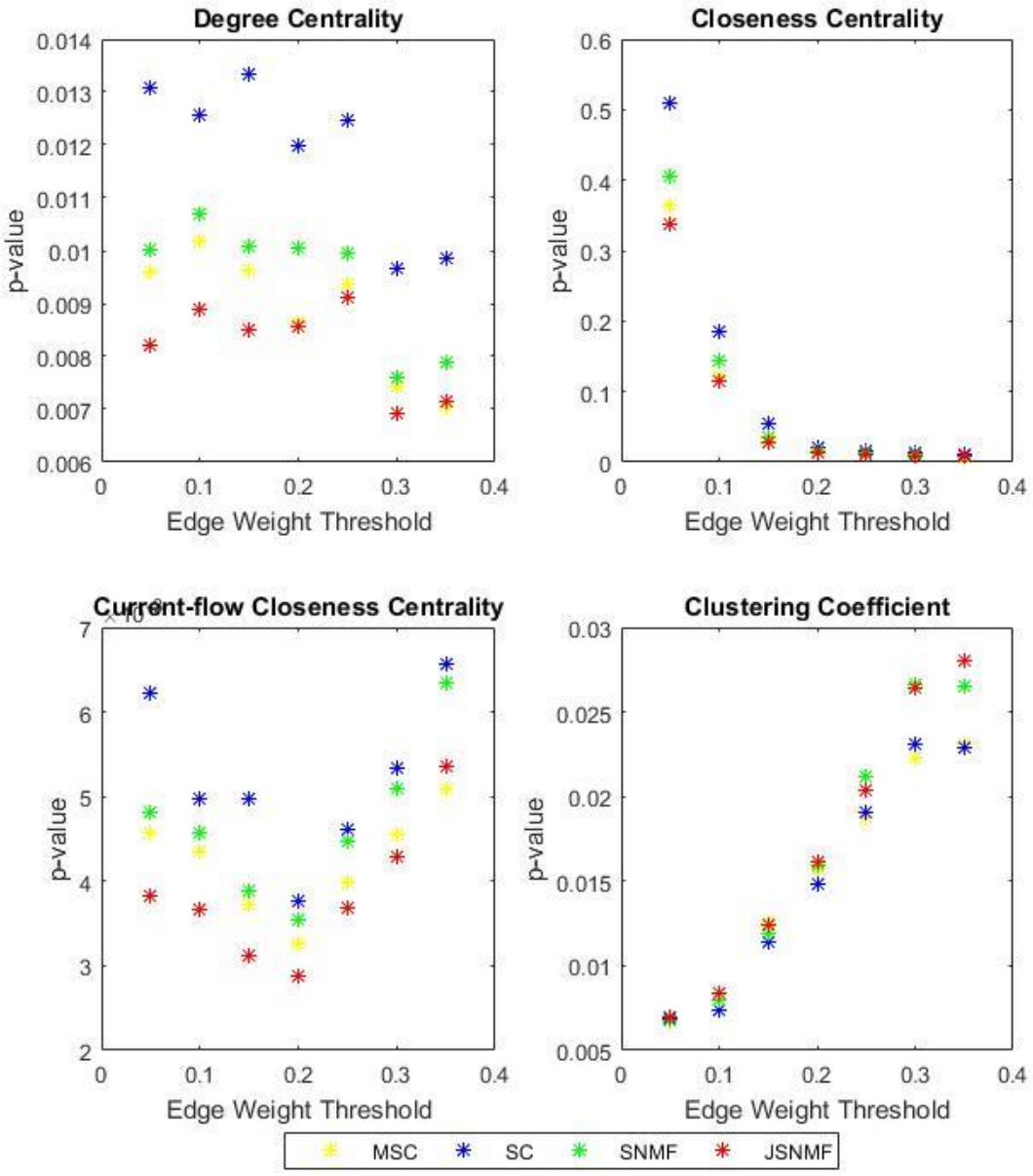

Figure 4.4 Comparison of p-values among 4 clustering methods on DMN

In Figure 4.3 and Figure 4.4, I show that for default mode system and 4 metrics I measure, JSNMF can reach highest f-scores and lowest p-values in most cases, especially with smaller edge weight thresholds, or very close to the top values.

4.4 Conclusion

Generally, there are two objectives for this chapter. One is to identify a module distinguishing ASD subjects from TD so that I can extract features from it and the other one is to furtherly demonstrate my clustering method is better than others.

I show in this chapter that DMN and FPN contain more significantly altered vertices and are also altered more as whole modules. Thus, I decide to extract features from these two modules for diagnosis of ASD.

In addition, I present comparison among 4 clustering algorithms and find that DMN and FPN identified by my method is more discriminant than those by other methods. This indicates that my method can identify the most discriminant module so it is better for studying modular alterations caused by ASD.

CHAPTER 5
AUTO-CLASSICATION OF ASD WITH MODULAR FEATURES

5.1 Introduction

In clinical practice, diagnosis of ASD mostly relies on interview- or observation-based instruments, including Autism Diagnostic Observation Schedule (ADOS) [120] and Autism Diagnostic Interview, Revised (ADI-R) [121], which are the most common tools used by clinicians. In ADOS, the subjects are asked to perform a series of tasks so that the clinicians can observe and evaluate the social and communication behaviors, while ADI-R focuses on interviewing parents and caretakers of the subjects. Therefore, the ratings of subjects' behaviors and the final diagnosis highly depend on the clinicians' personal expertise and caretakers' awareness of the symptoms, which may be biased by many factors. [57] reports that African-American children are much more likely than white children to be misdiagnosed as other diseases, such as conduct disorder, before they receive the diagnosis of ASD or ADHD. [57] indicates this is resulted from the often co-occurrence of other mental diseases with ASD and the expectation of clinicians towards different group of people having ASD, though no evidence shows differences in epidemiology of ASD by race.

Recent developments in machine learning inspire new approaches to study brain diseases, such as AD and schizophrenia. Such studies mostly use MRI data and focus on auto-classification of different group of subjects, e.g. ASD or AD vs. control group, or on identification of new biomarkers. These studies may vary by source of data, features selected and prediction models trained. [122] extracts grey matter volume as features and trains an support vector machine (SVM) model and an auto-encoder neural network to classify AD and MCI group. [123] also adopts SVM model but trains it with the probability that each voxel in T1-weighted images belongs to a tissue class and cortex thickness. [124] builds sparse FBNs from fMRI images and

calculates clustering coefficient to study MCI.

Considering the limitations of clinical diagnostic approaches, auto-classification of ASD can provide complementary information for clinicians' reference. In previous studies, different machine learning methods have been implemented on various datasets and yielded relatively high accuracy. Some studies directly calculate features from images. [125] uses regional and interregional morphological features, such as cortical thickness, subcortical volume and relative change in cortical thickness between pairs of regions. Histogram of oriented gradients, which describe the spatial gradient among voxels, are adopted in [126]. [127] calculates several metrics, e.g., concavity, curvature and metric distortion, measuring folding of cortex. Combined with different classifiers, the accuracies usually can reach 80% or higher in these studies. Except for morphological features from images, recent studies also extract features from brain networks. Most commonly used feature is network connectivity [128, 53, 52, 129, 55], which is basically the weight of edge, but other graph theory features are also used such as clustering coefficient, characteristic path length, etc. [54, 56]. With recent developments in deep learning, artificial neural networks (ANN) appear in ASD studies as both feature selection methods and classifiers. [53] uses deep neural network to select discriminant connectivity features from FBNs achieve accuracy of 86%. [128] feeds connectivity features to a probabilistic neural network for classification and have 90% accuracy. Comparisons among classifiers for ASD diagnosis are also made. [129] and [52] compare the performances of several classifiers trained with connectivity features, including SVM, logistic regression (LR), linear discriminant analysis (LDA), etc. [56] compares several neural networks trained with graph theory features. [130] compares SVM and other traditional machine learning methods with probabilistic neural network. The results of these researches indicate that the performances of classifiers vary much on different datasets and features.

In studies of auto-classification based on connectivity features, it is quite common to adopt feature selection strategies to find discriminant features before using them to train classifiers, because the dimension of feature vectors could be very high since it includes all the edge weights in it. Such high dimensional feature vectors may lead to overfitting issues, which means the prediction model is specified to the training dataset but performs poorly on other ones. Also,

less discriminant features may degrade the accuracy of prediction. In addition, a large number of features may increase the training time, especially when the classifier is complicated like a deep neural network. As mentioned above, some studies use ANN for feature selection [53], but also, there are other simpler strategies which ranks all the features and remove low-ranked ones [131].

In this chapter, I use previously obtained FBN clustering results as a strategy for feature selection. I extract features from one module and I show that it would improve the prediction performances of several commonly used classifiers. I also show that such a strategy can be combined with other feature selection methods for better results. In the following sections, I will introduce the whole process for auto-classification of ASD with features from one module and then present the classification results.

5.2 Methods

Generally, the pipeline for ASD diagnosis contains 3 stages—feature extraction, training the classifiers and validation of models. So, in this section, I introduce how I conduct each stage and classifiers I use will also be introduced in detail. In addition, I also give an introduction on the method to evaluating my classification results.

5.2.1 Feature Extraction

In this research, I directly use functional correlations, or the edge weights in FBNs as features. For whole-brain feature, I simply vectorize the connectivity matrices of all subjects and obtain a 79×34716 feature matrix where each row corresponds to a subject. The elements in the lower triangles and on the diagonals of connectivity matrices are removed. For DMN and FPN features, I firstly extract the subnetworks corresponding to the two modules from the whole networks, according to clustering results. Then I vectorize the subnetworks and obtain a 79×3403 and 79×1395 feature matrix, respectively. I should notice that any inter-modular edges are not included.

5.2.2 Classifiers

All the following classifiers are trained with whole-brain, DMN and FPN correlation features, separately.

5.2.2.1 Support vector machine

Given a training dataset $(x_1, y_1), \ldots, (x_n, y_n)$ with n observations, where x_i is the feature vector and y_i is the class label equal to either -1 or 1, linear SVM tries to find a linear hyperplane in space that divides the dataset into two classes. The hyperplane is supposed to be as far as possible to nearest observations in each class. In cases that observations are linearly separable, observations in each class should only appear in each side of the hyperplane. The hyperplane can be written as

$$f_{SVM}(x) = w \cdot x - b = 0, \tag{5.1}$$

where w and b can be determined by solving an optimization problem

$$\min \frac{1}{2} w \cdot w^T, s.t. y_i f_{SVM}(x_i) \geq 1 \tag{5.2}$$

However, in many cases, observations are not linearly separable, so I solve an adjusted optimization problem

$$\min_{w,b,\xi} \frac{1}{2} w \cdot w^T + \phi \sum_i \xi_i,$$

$$s.t. y_i f_{SVM}(x_i) \geq 1 - \xi_i, \xi_i \geq 0, \tag{5.3}$$

where ξ_i is a slack variable equal to $\max(0,\ 1 - y_i f_{SVM}(x_i))$ and ϕ is a penalty parameter. To solve the optimization problem, I can adopt Lagrangian multiplier to rewrite the objective function as

$$O_{SVM} = \frac{1}{2} w \cdot w^T + \phi \sum_i \xi_i - \sum_i \alpha_i (y_i f_{SVM}(x_i) - 1 + \xi_i) - \sum_i \xi_i \mu_i, \tag{5.4}$$

where α_i and μ_i are Lagrangian multipliers for w_i and ξ_i, respectively. Solving partial

derivative of O_{SVM} with respect to w, ξ_i and b equal to zero, I can obtain

$$w^T = \sum_i \alpha_i y_i x_i \tag{5.5}$$

$$\sum_i \alpha_i y_i = 0 \tag{5.6}$$

$$\alpha_i = \phi - \mu_i. \tag{5.7}$$

Substituting Equation (5.5) and (5.6) in to (5.4), I have

$$O_{SVM} = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \,. \tag{5.8}$$

From Equation (5.8), I can furtherly solve for α_i. I should notice that many α_i would be 0 and x_i corresponding to non-zero α_i are called marginal support vector. Then b can be obtained by

$$b = w \cdot x_i - y_i, \tag{5.9}$$

where x_i is a marginal support vector. I implement SVM with MATLAB Machine Learning Toolbox. The problem is solved with sequential minimal optimization, which is a stochastic gradient descent method updating two variables in the one iteration.

Linear SVM contains a linear kernel with kernel function $k_L(x_i, x_j) = x_i^T x_j$. For many cases, observations are not linearly separable, so I can try to project them into higher-dimensional space so that they became linearly separable. It could be very complicated to find a proper transformation. Therefore, I adopt the kernel functions, which are functions of inner product of observations, to simplify the problem. Another common kernel is Gaussian kernel defined as

$$k_G(x_i, x_j) = e^{-\|x_i - x_j\|^2}. \tag{5.10}$$

If I substitute the Equation (5.10) into (5.8), I obtain the Gaussian kernel SVM classifier.

Once the SVM is trained, I can test the machine with the testing dataset. So given an observation (x_t, y_t) with y_t being unknown, I can simply substitute x_t into $f_{SVM}(x)$. Then I can set up

a threshold for $f_{SVM}(x_t)$ to determine which class this observation belongs to. For example, under the default setting of MATLAB, $y_t = \begin{cases} -1, f_{SVM}(x_t) < 0 \\ 1, f_{SVM}(x_t) \geq 0 \end{cases}$. By choosing different threshold, the performance of the classifier may vary. Thus, I usually test a series of thresholds to find the best performance or evaluate the classifier comprehensively.

5.2.2.2 PSOSVM

PSOSVM is an SVM based method but with particle swarm optimization (PSO) as a feature selection strategy. Inspired by biological activities, PSO optimizes the solution by iteratively updating a population of potential solutions, which are regarded as particles [132]. In each iteration, the velocities of particles are calculated following certain rules and then used to update positions of particles. The iteration ends until certain criterions are met.

Introduced in [133], the idea of PSO for feature selection is using particles' positions to represent the subsets of features fed to the classifier, which is SVM in my case. If totally I have N features, the position of a particle is represented by an N-dimensional binary vector with each element corresponding to a feature. For values of elements, "1" means the corresponding feature is selected and "0" means not. I first randomly initialize the positions and velocities of a swarm of particles. Then I update their velocities according to the following equation

$$v_i = m * v_i + c_1 * rand * (p_i - x_i) + c_2 * rand * (g_i - x_i) \tag{5.11}$$

Where m is a weight or inertia factor, $p_i - x_i$ the distance between best position and current position of particle i, $g - x_i$ is the distance between global best position and current position of i, and c_1 and c_2 are coefficients I set to 2. The distance of two particles are defined as number of different elements in their position vectors. The velocities are rounded to its nearest integers toward negative infinity. To decide if a position is good, I use a fitness function defined as

$$Fitness = \alpha * AUC(F_{select}) + \beta * \frac{|F_{all}| - |F_{select}|}{|F_{all}|} \tag{5.12}$$

Where $AUC(D)$ is the area under curve (AUC, which will be furtherly introduced in the following section) of an SVM classifier trained with feature subset F_{select}, $|F_{all}|$ and $|F_{select}|$ are number of all features and selected features, respectively, and α and β are two parameters

weighting the importance of accuracy and number of features and are set to 0.9 and 0.1, respectively. The position updating rules are as follows. For each particle, I first calculate the distance between it and global best position, $g_i - x_i$, and its velocity v_i. If $v_i \leq g_i - x_i$, the I randomly choose v_i elements in x_i from all that different from g_i and flip them. And if $v_i > g_i - x_i$, I first make x_i equal to g_i then randomly choose $v_i - g_i + x_i$ elements in x_i and flip them.

In my experiment, the swarm size is set to 1000. Since the PSO algorithm is randomly initialized, I run it for 10 times. PSOSVM is implemented with MATLAB Machine Learning Toolbox.

5.2.2.3 RFESVM

The idea of recursive feature elimination (RFE) is ranking the features and eliminating lowest-ranked one recursively [134]. Consider I have in total n observations $x_1 \dots x_n$ and their corresponding labels $y_1 \dots \mathrm{y_n}$. In SVM, I decide labels of test data using $f_{SVM}(x) = w \cdot x - b$, where $w = \sum_{i=1}^{n} \alpha_i y_i x_i$. RFE uses $\left(w_f\right)^2$, which estimates the effect on the model of removing the corresponding feature [135], as the ranking parameter for feature f. In this research, I choose top 10 features. RFESVM is implemented with Scikit-learn in Python.

5.2.2.4 Random forest

RF ensembles a set of decision trees and unknown labels of test data is predicted by voting of decision trees. A decision tree is like a flowchart where each node represents a test on the value of certain features and branches from the node represent outcomes. To build a single decision tree, I use the classification and regression tree algorithm and Gini index as the split criterion. The tree is built from root to leaves, where the leaves means a final decision which class the observation belongs to. To start with, I calculate Gini index for each feature and each cataloguing threshold. The Gini index is defined as

$$\mathrm{I_{Gini}}(f, th) = \sum_{i=1}^{2} \frac{|C_i|}{n} \left(1 - \sum\nolimits_{k=1}^{K} \left(\frac{|C_{i,k}|}{|C_i|}\right)^2\right), \tag{5.13}$$

where f is a feature, th is a cataloguing threshold, K is the number of classes, C_i is the set of observations catalogued to branch i by the threshold th, and $C_{i,k}$ is a subset of C_i

containing observations in class k. After going through all pairs of features and cataloguing thresholds, I choose the one with lowest $\mathrm{I}_{\mathrm{Gini}}$ and set it as the root node. For the two branches of the node, if observations in the branch belong to the same class, then the branch reaches the leaf, otherwise, I repeat the last step with the rest observations and features until all branches reach leaves.

The idea of random forest is to prevent overfitting issues, which is quite common in implementation of decision trees. To achieve this, I randomly choose only two thirds of all observations to build each tree. In total, I build 1000 trees. Given a test sample, the forest outputs the fractions of trees voting for each class to make the final decision. RF is implemented with MATLAB Machine Learning Toolbox.

5.2.2.5 Linear discriminant analysis

The basic idea of LDA is to project observations to a lower dimensional space so that classes are clearly separated. The whole process to use LDA for classification can be generalized as follows. I first calculate the scatter matrix of my training data as

$$S = \sum_{i=1}^{n} (x_i - \bar{x})\,(x_i - \bar{x})^{\mathrm{T}}, \tag{5.14}$$

where $\bar{x}$ is the mean vector over all observations. Scatter matrix is an estimation of covariance matrix. Then I find the eigenvectors of S corresponding to largest $K-1$ eigenvalues and make a new matrix E with them. Multiplying x_i with E, I obtain a new observation x_i' with lower dimension.

To do classification, I use Bayes' rule

$$P(y_i = k|x_i') = \frac{P(x_i'|y_i = k)P(y_i = k)}{P(x_i')} = \frac{P(x_i'|y_i = k)P(y_i = k)}{\sum_{l=1}^{K} P(x_i'|y_i = l)P(y_i = l)} \tag{5.15}$$

where $P(y_i = k|x_i)$ is the probability that observation x_i belongs to class k. $P(y_i = k)$ is the fraction of k-class observations in the whole training data. $P(x_i'|y_i = k)$ can be estimated with Gaussian distribution that

$$P(x_i'|y_i = k) = (2\pi\sigma_k^2)^{-\frac{1}{2}} e^{-\frac{(x_i' - \mu_k)^2}{2\sigma_k^2}}, \tag{5.16}$$

where μ_k and σ_k are the mean and standard deviation of observations in class k, respectively. So given a testing sample, I obtain a probability for each class. LDA is implemented with Scikit-learn in Python.

5.2.2.6 Lasso regularized logistic regression

In logistic regression, I fit the training data with a sigmoid function. Added with lasso regularization, I can obtain the optimization problem

$$\min_{w,\gamma,b} \|w\|_1 + \gamma \sum_{i=1}^{n} \log\left(e^{-y_i(w \cdot x_i + b)} + 1\right), \tag{5.17}$$

where γ is a penalty factor and $\|\cdot\|_1$ represents L_1 norm. Optimization (5.17) can be solved by the stochastic gradient descent method. LRLR is implemented with Scikit-learn in Python.

5.2.2.7 k nearest neighbors

In kNN, the label of a testing sample is decided by the labels of its k nearest neighbors. In my study, I measure the Euclidean distance and the number of neighbors is set to the one yielding best performance. kNN is implemented with MATLAB Machine Learning Toolbox.

5.2.3 Leave-one-out Cross Validation

Leave-one-out cross validation (LOOCV) is an extreme case of k-fold validation. In k-fold validation, the dataset is randomly divided into k subsets with equal sizes. k-1 subsets are used as training data and the rest one for testing. If I go through all the k subsets as testing data, I can obtain an average performance of the classifier and every observation is used once for both training and testing. The k value may vary depend on actual situation, but 10-fold is commonly used.

I can regard LOOCV as k-fold validation with k equal to the number of all observations. My dataset has in total 79 observations, so 78 of them will be used for training and the rest one for

testing. I use each observation for testing and obtain 79 prediction results. Then I evaluate the performance of the classifier based on the predictions. LOOCV tends to have lower bias but higher variance. However, considering the size of the dataset is relatively small, k-fold validation may also have high variance, so it is still reasonable to adopt LOOCV here.

5.2.4 Evaluation Indices for Prediction Results

The performances of classifiers are evaluated by plotting receiver operating characteristic (ROC) curves and calculating AUCs. I plot ROC curves with true positive rate (TPR) on the y-axis against false positive rate (FPR) on the x-axis. Given the Table (5.1) showing the possible outcomes of a classification process, I have

Table 5.1 A demonstration of all possible outcomes of a prediction

Number of samples		True class	
		Positive	Negative
Predicted class	Positive	True positive	False positive
	Negative	False negative	True negative

$$TPR = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{5.18}$$

$$FPR = \frac{\text{Frue positive}}{\text{False positive} + \text{True negative}} \tag{5.19}$$

To draw the ROC curves, I need to calculate TPR and FPR at a series of classification thresholds. The thresholds are for the output value of a classifier to decide which class a testing sample should belong to. For SVM based classifiers, the threshold ranges from the minimal value it can possibly reach to the maximum. And for other classifiers, the output is probabilities an testing sample belongs to classes. Thus, I put a threshold on the probability corresponding to the first class. If the probability larger than the threshold, the sample goes to the first class, or otherwise, it goes to the other class. ROC curves comprehensively show the performance of classifiers under different condition. A perfect classifier should have a ROC curve reaching point $(0,1)$, which means 100% TPR and 0% FPR.

AUC is the area under the ROC curve and it measures the general performance of a classifier. If the AUC is large, it means the ROC curve is close to point $(0,1)$, or at least it partially has high TPR or low FPR. In practice, I use trapezoidal integration method to calculate AUCs.

5.3 Results and Discussions

I train each classifier with DMN features, FPN features and whole-brain features, respectively, and validate all classifiers with LOOCV. To evaluate the performances, I draw the ROC curves and calculate the AUCs of each classifier trained with each type of features.

The main purpose of feature selection is to reduce the number of features so that the classifiers can be trained faster and to select more discriminant features to achieve better performance. In most of cases, many features can be redundant or irrelevant, which means these features cannot show the differences among groups. Since such features are equally weighted with discriminant features, the classifiers may try to find a way to distinguish groups according to these irrelevant features, which may yield poor classification performances.

Table 5.2 shows that the number of features has been significantly reduced by my feature selection strategy. As shown in Table 5.3, DMN features achieve higher AUCs for all classifiers except for RFESVM. The AUCs of RFESVM are very close and are all relatively high. And in actual implementation, RFESVM on DMN features runs much faster than it on whole-brain features. AUCs of Gaussian kernel SVM are higher than linear SVM on all three feature sets, and PSOSVM and RFESVM also outperform linear SVM and most of other classifiers, indicating that a second feature selection steps and kernel tricks have the potential to furtherly boost the performances. From Figure 5.1, I can see that DMN features have higher TPR especially when FPR is low, which indicates that DMN features may have more potential in clinical diagnosis where low misdiagnosis rate is necessary.

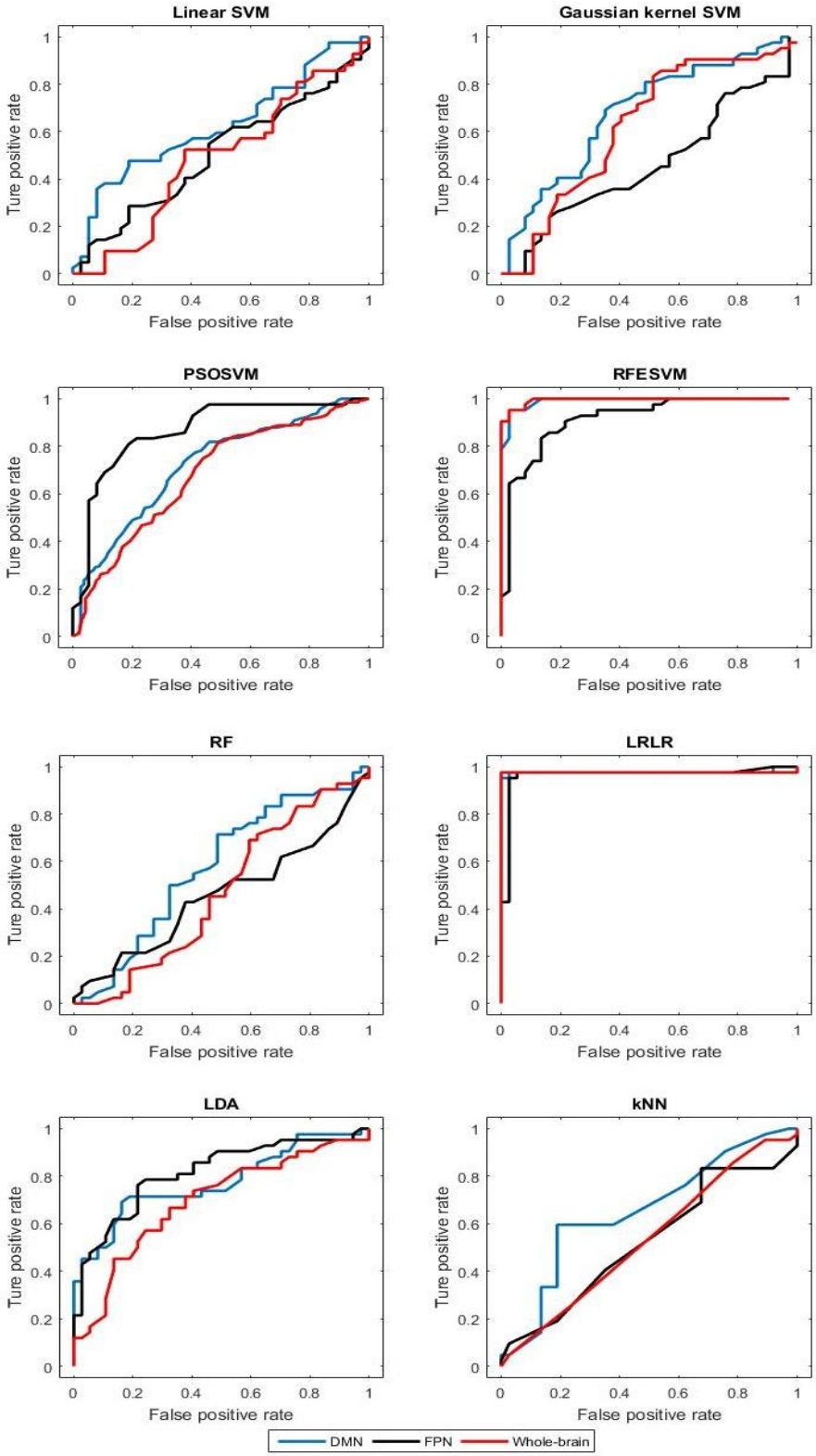

Figure 5.1 ROCs of different classifiers.

Even though DMN features achieve relatively good performances, the AUCs of FPN features

are not as high as FPN features. In some cases, they are even lower than AUCs of whole-brain features. Possibly, this is because correlation features that I use are prone to deliver more local properties other than modular ones. As I have shown in Chapter 4, f-score, which measures the enrichment of significantly altered vertices in a module, can be also regarded as a measurement describing the degree of local alterations. And I can see in Figure 4.1 that, in many cases, FPN do not have a higher score than DMN. This indicates, on one hand, if I use some features for properties in a modular level, it is possible to improve the classification performances of FPN.

Table 5.2 Comparison on numbers of features

	Whole-brain	DMN	FPN
Number of features	34716	3486	1378

Table 5.3 AUCs of classifiers trained with different features

Classifiers	Linear SVM	Gaussian kernel SVM	PSOSVM	RFESVM	RF	LDA	LRLR	kNN
DMN	0.6239	0.6538	0.7215	0.9640	0.5769	0.7754	0.9775	0.6541
FPN	0.5051	0.4347	0.9136	0.8877	0.4569	0.8144	0.9643	0.5248
Whole-brain	0.4871	0.6252	0.5822	0.9675	0.4678	0.6943	0.9762	0.5347

5.4 Conclusion

In this chapter, I extract correlation features from two modules previously identified and use the features to train several common classifiers. I evaluate the prediction results by plotting ROC curves and calculating AUCs. I found that AUCs of classifiers trained with DWN features are higher than those with whole-brain features. I show the further improvement of performances of modular features with another feature selection stage or kernel trick. I also

discuss the possible reasons that FPN features cannot have good performances. In general, I show that my new feature selection strategy that extracting features from a functional module but not from the whole brain is valid, and its potential to be furtherly improved.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this thesis, I propose a new feature selection strategy for auto-diagnosis of ASD based on fMRI data. This strategy is based on building FBNs from fMRI images, network clustering algorithm identifying functional modules from FBNs, then statistically analysis finding significantly altered modules to extract features. Finally, I train several classifiers with the features I obtain in this study and traditionally used whole-brain features, and find that my features can yield better classification performances than whole-brain features.

In Chapter 2, I demonstrate how to build FBNs from fMRI images. I introduce the database I obtain the data from and the dataset I use for this study. Then the whole preprocessing pipeline of fMRI images are presented, including steps I take and the corresponding tools for each step. The whole pipeline includes skull stripping, motion correction, removing outliers that move too much, noise regression, spatial smoothing and image registration to standard space. After preprocessing, I extract signals from 264 ROIs on the brain cortex and calculate the PCCs between any pair of them. The ROIs are referred as vertices in the FBNs and PCCs as weights of edges. I obtain an FBN for each subject and the FBNs are represented as 264×264 symmetrical connectivity matrices.

In Chapter 3, I propose a new NMF based network clustering algorithm named JSNMF. This algorithm can factorize multiple symmetrical matrices simultaneously and result in a consensus matrix where each row is a lower-dimensional feature vector corresponding to a vertex. I obtain four functional modules from the consensus feature matrix. I evaluate the clustering results with 3 indices—modularity, conductance and coverage. The indices are calculated both on average

and individual FBNs. I find that, compared with other competing methods, my method yields better clustering results in some terms, though the improvements could be rather moderate. Also, I discuss the neurological meanings of each module and link brain functions to the modules.

To choose from which module I extract features and to demonstrate JSNMF is better from a different perspective, in Chapter 4, I calculate some indices measuring modular properties, including average centralities and enrichment of significantly altered vertices. Then I do the t-test on the average centralities between ASD and TD group and calculate the f-score to evaluate the enrichment. I show that modules I labeled DMN and FPN are both enriched with significantly altered vertices and have low p-value from t-test, thus I decide to extract features from these two modules. Also, comparisons are made among clustering algorithms. I find that the DMN and FPN module identified by my algorithm have lower p-value and higher f-score. Therefore, I believe my algorithm is better than others because it can find more discriminant modules and help understand the pathophysiology of ASD. I also combine the results of this research with some previous ones to demonstrate the neurological validity of the alterations.

In Chapter 5, I extract features from DMN, FPN and the whole brain to train several classifiers. I use correlation values, i.e., the weights of edges, as the features. The predictions are validated by LOOCV. I draw the ROC curves and calculate AUCs to evaluate the performances of classifiers. I show that classifiers trained by DMN features achieve better AUCs. And combined with a second feature selection step or kernel trick, the performances can be boosted.

In general, the new feature selection strategy that extracting features from a functional module is presented in this study and the I show that, in diagnosing ASD, this strategy can improve classification performances compared with cases that use whole-brain features. Such strategy can also largely reduce the numbers of features which reduces the training time and potentially avoid overfitting issue. In addition, this method is based a new network clustering algorithm which also outperforms other competing ones in FBN clustering.

6.2 Future works

As I show in Chapter 5, when I implement PSOSVM, RFESVM and Gaussian kernel SVM, I obtain better AUCs than ordinary SVM, which indicates that other feature selection methods and kernel tricks may boost the classification performance. Therefore, the next step of this work can be focused on effectively combining multiple feature selection strategies and kernel methods to achieve a better classifier for ASD diagnosis.

In this study, I try to distinguish ASD subjects from TD group. There are several other neurodevelopmental diseases that is closely related to ASD such as ADHD or Asperger syndrome which are often studied along with ASD. Thus, another direction of future work is to include subjects with such diseases into my study and perform multi-class diagnosis with modular features. In addition, my feature selection strategy can be potentially introduced to diagnose of other brain diseases, like AD and MCI.

Thirdly, in Chapter 3, I show the improvements of my clustering algorithm are relatively small. But I believe that the NMF-based method can be furtherly modified for better clustering results. In Equation (3.4), I only use one regularization term to make the consensus matrix sparse, however, I can find several other terms from previous studies to regularize the result, such as a graph regularization term introduced in [136].

Last but not the least, I can also try to interrogate multiple sources of data such as DWI, CT of PET for both network clustering and brain disease diagnosis.

REFERENCES

[1] P. A. Levine, M. R. Paling, W. C. Back and R. W. Cantrell, "MRI vs. high-resolution CT scanning: evaluation of the anterior skull base," *Otolaryngology—Head and Neck Surgery,* vol. 96, pp. 260-267, 1987.

[2] D. W. McRobbie, E. A. Moore, M. J. Graves and M. R. Prince, MRI from Picture to Proton, Cambridge university press, 2017.

[3] K. A. Jellinger, "Magnetic Resonance Imaging, David D. Stark and William G. Bradley, Jr (eds). Volumes I--III. Mosby, St. Louis, MO, USA, 1999. 1936+≈ 100 pp. ISBN 0-8151-8518-9; 28607.," *European Journal of Neurology,* vol. 8, pp. 96-97, 2001.

[4] R. A. Sadek, "An improved MRI segmentation for atrophy assessment," *International Journal of Computer Science Issues (IJCSI),* vol. 9, p. 569, 2012.

[5] R. A. Sadek, "Regional atrophy analysis of MRI for early detection of alzheimer's disease," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 6, pp. 49-58, 2013.

[6] C. A. Mutch, J. F. Talbott and A. Gean, "Imaging evaluation of acute traumatic brain injury," *Neurosurgery Clinics,* vol. 27, pp. 409-439, 2016.

[7] M. D'esposito, E. Zarahn and G. K. Aguirre, "Event-related functional MRI: implications for cognitive psychology.," *Psychological bulletin,* vol. 125, p. 155, 1999.

[8] S. A. Huettel, A. W. Song, G. McCarthy and others, Functional magnetic resonance imaging, vol. 1, Sinauer Associates Sunderland, MA, 2004.

[9] V. Subbaraju, S. Sundaram and S. Narasimhan, "Identification of lateralized compensatory neural activities within the social brain due to autism spectrum disorder in adolescent males," *European Journal of Neuroscience,* vol. 47, pp. 631-642, 2018.

[10] D. Tombari, I. Loubinoux, J. Pariente, A. Gerdelat, J.-F. Albucher, J. Tardy, E. Cassol and F. Chollet, "A longitudinal fMRI study: in recovering and then in clinically stable sub-cortical stroke patients," *Neuroimage,* vol. 23, pp. 827-839, 2004.

[11] G. J. Siegle, C. S. Carter and M. E. Thase, "Use of FMRI to predict recovery from unipolar depression with cognitive behavior therapy," *American Journal of Psychiatry,* vol. 163, pp. 735-738, 2006.

[12] X. Delbeuck, M. Van der Linden and F. Collette, "Alzheimer'disease as a disconnection syndrome?," *Neuropsychology review,* vol. 13, pp. 79-92, 2003.

[13] M. Pievani, F. Agosta, E. Pagani, E. Canu, S. Sala, M. Absinta, C. Geroldi, R. Ganzola, G. B. Frisoni and M. Filippi, "Assessment of white matter tract damage in mild cognitive impairment and Alzheimer's disease," *Human brain mapping,* vol. 31, pp. 1862-1875, 2010.

[14] S. H. Ameis, J. Fan, C. Rockel, A. N. Voineskos, N. J. Lobaugh, L. Soorya, A. T. Wang, E. Hollander and E. Anagnostou, "Impaired structural connectivity of socio-emotional circuits in autism spectrum disorders: a diffusion tensor imaging study," *PloS one,* vol.

6, p. e28044, 2011.

[15] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen and O. Sporns, "Mapping the structural core of human cerebral cortex," *PLoS biology,* vol. 6, p. e159, 2008.

[16] P. Hagmann, M. Kurant, X. Gigandet, P. Thiran, V. J. Wedeen, R. Meuli and J.-P. Thiran, "Mapping human whole-brain structural networks with diffusion MRI," *PloS one,* vol. 2, p. e597, 2007.

[17] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar and others, "Functional network organization of the human brain," *Neuron,* vol. 72, pp. 665-678, 2011.

[18] B. Biswal, F. Zerrin Yetkin, V. M. Haughton and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar MRI," *Magnetic resonance in medicine,* vol. 34, pp. 537-541, 1995.

[19] S. Achard, R. Salvador, B. Whitcher, J. Suckling and E. D. Bullmore, "A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs," *Journal of Neuroscience,* vol. 26, pp. 63-72, 2006.

[20] D. S. Bassett and E. D. Bullmore, "Small-world brain networks," *The neuroscientist,* vol. 12, pp. 512-523, 2006.

[21] N. A. Crossley, A. Mechelli, P. E. Vértes, T. T. Winton-Brown, A. X. Patel, C. E. Ginestet, P. McGuire and E. T. Bullmore, "Cognitive relevance of the community structure of the human brain functional coactivation network," *Proceedings of the National Academy of Sciences,* vol. 110, pp. 11583-11588, 2013.

[22] P. Barttfeld, B. Wicker, S. Cukier, S. Navarta, S. Lew and M. Sigman, "A big-world network in ASD: dynamical connectivity analysis reflects a deficit in long-range connections and an excess of short-range connections," *Neuropsychologia,* vol. 49, pp. 254-263, 2011.

[23] N. M. Kleinhans, T. Richards, L. Sterling, K. C. Stegbauer, R. Mahurin, L. C. Johnson, J. Greenson, G. Dawson and E. Aylward, "Abnormal functional connectivity in autism spectrum disorders during face processing," *Brain,* vol. 131, pp. 1000-1012, 2008.

[24] J. D. Rudie, J. A. Brown, D. Beck-Pancer, L. M. Hernandez, E. L. Dennis, P. M. Thompson, S. Y. Bookheimer and M. Dapretto, "Altered functional and structural brain network organization in autism," *NeuroImage: clinical,* vol. 2, pp. 79-94, 2013.

[25] R. J. Jou, A. P. Jackowski, X. Papademetris, N. Rajeevan, L. H. Staib and F. R. Volkmar, "Diffusion tensor imaging in autism spectrum disorders: preliminary evidence of abnormal neural connectivity," *Australian and New Zealand Journal of Psychiatry,* vol. 45, pp. 153-162, 2011.

[26] F. Bai, N. Shu, Y. Yuan, Y. Shi, H. Yu, D. Wu, J. Wang, M. Xia, Y. He and Z. Zhang, "Topologically convergent and divergent structural connectivity patterns between patients with remitted geriatric depression and amnestic mild cognitive impairment," *Journal of Neuroscience,* vol. 32, pp. 4307-4318, 2012.

[27] M. Daianu, N. Jahanshad, T. M. Nir, C. R. Jack Jr, M. W. Weiner, M. A. Bernstein, P. M. Thompson and A. D. N. Initiative, "Rich club analysis in the Alzheimer's disease

connectome reveals a relatively undisturbed structural core network," *Human brain mapping,* vol. 36, pp. 3087-3103, 2015.

[28] K. Supekar, V. Menon, D. Rubin, M. Musen and M. D. Greicius, "Network analysis of intrinsic functional brain connectivity in Alzheimer's disease," *PLoS computational biology,* vol. 4, p. e1000100, 2008.

[29] C.-Y. Lo, P.-N. Wang, K.-H. Chou, J. Wang, Y. He and C.-P. Lin, "Diffusion tensor tractography reveals abnormal topological organization in structural cortical networks in Alzheimer's disease," *Journal of Neuroscience,* vol. 30, pp. 16876-16885, 2010.

[30] G. Repovs, J. G. Csernansky and D. M. Barch, "Brain network connectivity in individuals with schizophrenia and their siblings," *Biological psychiatry,* vol. 69, pp. 967-973, 2011.

[31] A. F. Alexander-Bloch, P. E. Vertes, R. Stidd, F. Lalonde, L. Clasen, J. Rapoport, J. Giedd, E. T. Bullmore and N. Gogtay, "The anatomical distance of functional connections predicts brain network topology in health and schizophrenia," *Cerebral cortex,* vol. 23, pp. 127-138, 2012.

[32] Q. Yu, J. Sui, S. Rachakonda, H. He, W. Gruner, G. Pearlson, K. A. Kiehl and V. D. Calhoun, "Altered topological properties of functional network connectivity in schizophrenia during resting state: a small-world brain network study," *PloS one,* vol. 6, p. e25423, 2011.

[33] o. American Psychiatric Association, Diagnostic and statistical manual of mental disorders, American Psychiatric Pub, 2013.

[34] T. Vos, C. Allen, M. Arora, R. M. Barber, Z. A. Bhutta, A. Brown, A. Carter, D. C. Casey, F. J. Charlson, A. Z. Chen and others, "Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990--2015: a systematic analysis for the Global Burden of Disease Study 2015," *The Lancet,* vol. 388, pp. 1545-1602, 2016.

[35] L. Wing and D. Potter, "The epidemiology of autistic spectrum disorders: is the prevalence rising?," *Mental retardation and developmental disabilities research reviews,* vol. 8, pp. 151-161, 2002.

[36] E. Duchan and D. R. Patel, "Epidemiology of autism spectrum disorders," *Pediatric Clinics,* vol. 59, pp. 27-43, 2012.

[37] A. C. Stanfield, A. M. McIntosh, M. D. Spencer, R. Philip, S. Gaur and S. M. Lawrie, "Towards a neuroanatomy of autism: a systematic review and meta-analysis of structural magnetic resonance imaging studies," *European psychiatry,* vol. 23, pp. 289-299, 2008.

[38] G. S. Dichter, J. N. Felder and J. W. Bodfish, "Autism is characterized by dorsal anterior cingulate hyperactivation during social target detection," *Social cognitive and affective neuroscience,* vol. 4, pp. 215-226, 2009.

[39] E. Redcay, D. Dodell-Feder, P. L. Mavros, M. Kleiner, M. J. Pearrow, C. Triantafyllou, J. D. Gabrieli and R. Saxe, "Atypical brain activation patterns during a face-to-face joint attention game in adults with autism spectrum disorder," *Human brain mapping,* vol. 34, pp. 2511-2523, 2013.

[40] N. J. Minshew and T. A. Keller, "The nature of brain dysfunction in autism: functional

brain imaging studies," *Current opinion in neurology,* vol. 23, p. 124, 2010.

[41] N. Hadjikhani, R. M. Joseph, J. Snyder and H. Tager-Flusberg, "Abnormal activation of the social brain during face perception in autism," *Human brain mapping,* vol. 28, pp. 441-449, 2007.

[42] M. Zilbovicius, I. Meresse, N. Chabane, F. Brunelle, Y. Samson and N. Boddaert, "Autism, the superior temporal sulcus and social perception," *Trends in neurosciences,* vol. 29, pp. 359-366, 2006.

[43] K. Supekar, L. Q. Uddin, A. Khouzam, J. Phillips, W. D. Gaillard, L. E. Kenworthy, B. E. Yerys, C. J. Vaidya and V. Menon, "Brain hyperconnectivity in children with autism and its links to social deficits," *Cell reports,* vol. 5, pp. 738-747, 2013.

[44] J.-Y. Koh, J. S. Lim, H.-R. Byun and M.-H. Yoo, "Abnormalities in the zinc-metalloprotease-BDNF axis may contribute to megalencephaly and cortical hyperconnectivity in young autism spectrum disorder patients," *Molecular brain,* vol. 7, p. 64, 2014.

[45] A. Di Martino, K. Ross, L. Q. Uddin, A. B. Sklar, F. X. Castellanos and M. P. Milham, "Functional brain correlates of social and nonsocial processes in autism spectrum disorders: an activation likelihood estimation meta-analysis," *Biological psychiatry,* vol. 65, pp. 63--74, 2009.

[46] L. Q. Uddin, K. Supekar and V. Menon, "Reconceptualizing functional brain connectivity in autism from a developmental perspective," *Frontiers in human neuroscience,* vol. 7, p. 458, 2013.

[47] J. S. Nomi and L. Q. Uddin, "Developmental changes in large-scale network connectivity in autism," *NeuroImage: Clinical,* pp. 732--741, 2015.

[48] D. E. Welchew, C. Ashwin, K. Berkouk, R. Salvador, J. Suckling, S. Baron-Cohen and E. Bullmore, "Functional disconnectivity of the medial temporal lobe in Asperger's syndrome," *Biological psychiatry,* vol. 57, pp. 991-998, 2005.

[49] K. Konrad and S. B. Eickhoff, "Is the ADHD brain wired differently? A review on structural and functional connectivity in attention deficit hyperactivity disorder," *Human brain mapping,* vol. 31, pp. 904-916, 2010.

[50] L. Cocchi, I. E. Bramati, A. Zalesky, E. Furukawa, L. F. Fontenelle, J. Moll, G. Tripp and P. Mattos, "Altered functional brain connectivity in a non-clinical sample of young adults with attention-deficit/hyperactivity disorder," *Journal of Neuroscience,* vol. 32, pp. 17753-17761, 2012.

[51] J. R. Koza, F. H. Bennett, D. Andre and M. A. Keane, "Automated design of both the topology and sizing of analog electrical circuits using genetic programming," in *Artificial Intelligence in Design'96*, Springer, 1996, pp. 151-170.

[52] C. P. Chen, C. L. Keown, A. Jahedi, A. Nair, M. E. Pflieger, B. A. Bailey and R.-A. Müller, "Diagnostic classification of intrinsic functional connectivity highlights somatosensory, default mode, and visual regions in autism," *NeuroImage: Clinical,* vol. 8, pp. 238-245, 2015.

[53] X. Guo, K. C. Dominick, A. A. Minai, H. Li, C. A. Erickson and L. J. Lu, "Diagnosing autism spectrum disorder from brain resting-state functional connectivity patterns using

a deep neural network with a novel feature selection method," *Frontiers in neuroscience,* vol. 11, p. 460, 2017.

[54] E. Tolan and Z. Isik, "Graph Theory Based Classification of Brain Connectivity Network for Autism Spectrum Disorder," in *International Conference on Bioinformatics and Biomedical Engineering*, 2018.

[55] H. Chen, X. Duan, F. Liu, F. Lu, X. Ma, Y. Zhang, L. Q. Uddin and H. Chen, "Multivariate classification of autism spectrum disorder using frequency-specific resting-state functional connectivity—a multi-center study," *Progress in Neuro-Psychopharmacology and Biological Psychiatry,* vol. 64, pp. 1-9, 2016.

[56] X.-a. Bi, Y. Liu, Q. Jiang, Q. Shu, Q. Sun and J. Dai, "The diagnosis of autism spectrum disorder based on the random neural network cluster," *Frontiers in human neuroscience,* vol. 12, p. 257, 2018.

[57] D. S. Mandell, R. F. Ittenbach, S. E. Levy and J. A. Pinto-Martin, "Disparities in diagnoses received prior to a diagnosis of autism spectrum disorder," *Journal of autism and developmental disorders,* vol. 37, pp. 1795-1802, 2007.

[58] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto and others, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Molecular psychiatry,* vol. 19, p. 659, 2014.

[59] J. P. Mugler III and J. R. Brookeman, "Three-dimensional magnetization-prepared rapid gradient-echo imaging (3D MP RAGE)," *Magnetic Resonance in Medicine,* vol. 15, pp. 152-157, 1990.

[60] J. A. Brown, J. D. Rudie, A. Bandrowski, J. D. Van Horn and S. Y. Bookheimer, "The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis," *Frontiers in neuroinformatics,* vol. 6, p. 28, 2012.

[61] R. W. Cox, "AFNI: software for analysis and visualization of functional magnetic resonance neuroimages," *Computers and Biomedical research,* vol. 29, pp. 162-173, 1996.

[62] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich and S. M. Smith, "Fsl," *Neuroimage,* vol. 62, pp. 782-790, 2012.

[63] M. Jenkinson, P. Bannister, M. Brady and S. Smith, "Improved optimization for the robust and accurate linear registration and motion correction of brain images," *Neuroimage,* vol. 17, pp. 825-841, 2002.

[64] M. S. Dagli, J. E. Ingeholm and J. V. Haxby, "Localization of cardiac-induced signal change in fMRI," *Neuroimage,* vol. 9, pp. 407-415, 1999.

[65] C. Windischberger, H. Langenberger, T. Sycha, E. M. Tschernko, G. Fuchsjäger-Mayerl, L. Schmetterer and E. Moser, "On the origin of respiratory artifacts in BOLD-EPI of the human brain," *Magnetic resonance imaging,* vol. 20, pp. 575-582, 2002.

[66] C. Chang, J. P. Cunningham and G. H. Glover, "Influence of heart rate on the BOLD signal: the cardiac response function," *Neuroimage,* vol. 44, pp. 857-869, 2009.

[67] M. D. Fox, D. Zhang, A. Z. Snyder and M. E. Raichle, "The global signal and observed

anticorrelated resting state brain networks," *Journal of neurophysiology,* vol. 101, pp. 3270-3283, 2009.

[68] Y. Zhang, M. Brady and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE transactions on medical imaging,* vol. 20, pp. 45-57, 2001.

[69] Z. S. Saad, S. J. Gotts, K. Murphy, G. Chen, H. J. Jo, A. Martin and R. W. Cox, "Trouble at rest: how correlation patterns and group differences become distorted after global signal regression," *Brain connectivity,* vol. 2, pp. 25-32, 2012.

[70] K. Murphy, R. M. Birn, D. A. Handwerker, T. B. Jones and P. A. Bandettini, "The impact of global signal regression on resting state correlations: are anti-correlated networks introduced?," *Neuroimage,* vol. 44, pp. 893-905, 2009.

[71] J. Wang, L. Wang, Y. Zang, H. Yang, H. Tang, Q. Gong, Z. Chen, C. Zhu and Y. He, "Parcellation-dependent small-world brain functional networks: a resting-state fMRI study," *Human brain mapping,* vol. 30, pp. 1511-1523, 2009.

[72] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman and others, "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest," *Neuroimage,* vol. 31, pp. 968-980, 2006.

[73] B. Fischl, A. Van Der Kouwe, C. Destrieux, E. Halgren, F. Ségonne, D. H. Salat, E. Busa, L. J. Seidman, J. Goldstein, D. Kennedy and others, "Automatically parcellating the human cerebral cortex," *Cerebral cortex,* vol. 14, pp. 11-22, 2004.

[74] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer and M. Joliot, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *Neuroimage,* vol. 15, pp. 273-289, 2002.

[75] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001.

[76] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature,* vol. 401, p. 788, 1999.

[77] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational statistics & data analysis,* vol. 52, pp. 155-173, 2007.

[78] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013.

[79] X. Cao, X. Wang, D. Jin, Y. Cao and D. He, "Identifying overlapping communities as well as hubs and outliers via nonnegative matrix factorization," *Scientific reports,* vol. 3, p. 2993, 2013.

[80] L. Ou-Yang, D.-Q. Dai and X.-F. Zhang, "Protein complex detection via weighted ensemble clustering based on Bayesian nonnegative matrix factorization," *PloS one,* vol. 8, p. e62158, 2013.

[81] W. Ou, S. Yu, G. Li, J. Lu, K. Zhang and G. Xie, "Multi-view non-negative matrix factorization by patch alignment framework with view consistency," *Neurocomputing,* vol. 204, pp. 116-124, 2016.

[82] L. Zong, X. Zhang, L. Zhao, H. Yu and Q. Zhao, "Multi-view clustering via multi-manifold regularized non-negative matrix factorization," *Neural Networks,* vol. 88, pp. 74-89, 2017.

[83] A. Kumar, P. Rai and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in neural information processing systems*, 2011.

[84] J. Liu, C. Wang, J. Gao and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013.

[85] Y. Li, F. Nie, H. Huang and J. Huang, "Large-Scale Multi-View Spectral Clustering via Bipartite Graph.," in *AAAI*, 2015.

[86] C. Ding, X. He and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005.

[87] H. Wang, H. Huang and C. Ding, "Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011.

[88] T. M. Cover and J. A. Thomas, "Entropy, relative entropy and mutual information," *Elements of information theory,* vol. 2, pp. 1-55, 1991.

[89] K. Y. Yeung and W. L. Ruzzo, "Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data," *Bioinformatics,* vol. 17, pp. 763-774, 2001.

[90] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics,* vol. 20, pp. 53-65, 1987.

[91] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," 1973.

[92] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering,* vol. 25, pp. 1336-1353, 2013.

[93] L.-P. Tian, P. Luo, H. Wang, H. Zheng and F.-X. Wu, "CASNMF: A Converged Algorithm for symmetrical nonnegative matrix factorization," *Neurocomputing,* vol. 275, pp. 2031-2040, 2018.

[94] X. Y. Stella and J. Shi, "Multiclass spectral clustering," p. 313, 2003.

[95] D. P. Kennedy and E. Courchesne, "The intrinsic functional organization of the brain is altered in autism," *Neuroimage,* vol. 39, pp. 1877-1885, 2008.

[96] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature reviews neuroscience,* vol. 3, p. 201, 2002.

[97] N. U. F. Dosenbach, D. A. Fair, F. M. Miezin, A. L. Cohen, K. K. Wenger, R. A. T.

Dosenbach, M. D. Fox, A. Z. Snyder, J. L. Vincent, M. E. Raichle and others, "Distinct brain networks for adaptive and stable task control in humans," *Proceedings of the National Academy of Sciences,* vol. 104, pp. 11073-11078, 2007.

[98] G. L. Shulman, J. A. Fiez, M. Corbetta, R. L. Buckner, F. M. Miezin, M. E. Raichle and S. E. Petersen, "Common blood flow changes across visual tasks: II. Decreases in cerebral cortex," *Journal of cognitive neuroscience,* vol. 9, pp. 648-663, 1997.

[99] J. R. Andrews-Hanna, "The brain's default network and its adaptive role in internal mentation," *The Neuroscientist,* vol. 18, pp. 251-270, 2012.

[100] R. L. Buckner, J. R. Andrews-Hanna and D. L. Schacter, *The brain's default network-Anatomy, function, and relevance to disease. Year Cogn. Neurosci. 2008 (1124), 1--38,* 2008.

[101] M. Bedny, A. Pascual-Leone, S. Dravida and R. Saxe, "A sensitive period for language in the visual cortex: distinct patterns of plasticity in congenitally versus late blind adults," *Brain and language,* vol. 122, pp. 162-170, 2012.

[102] S. Clarke and J. Miklossy, "Occipital cortex in man: organization of callosal connections, related myelo-and cytoarchitecture, and putative boundaries of functional visual areas," *Journal of Comparative Neurology,* vol. 298, pp. 188-214, 1990.

[103] I. Bodis-Wollner, A. Atkin, E. Raab and M. Wolkstein, "Visual association cortex and vision in man: Pattern-evoked occipital potentials in a blind boy," *Science,* vol. 198, pp. 629-631, 1977.

[104] H. Burton, R. J. Sinclair, J. R. Wingert and D. L. Dierker, "Multiple parietal operculum subdivisions in humans: tactile activation maps," *Somatosensory & motor research,* vol. 25, pp. 149-162, 2008.

[105] J. S. Anderson, T. J. Druzgal, A. Froehlich, M. B. DuBray, N. Lange, A. L. Alexander, T. Abildskov, J. A. Nielsen, A. N. Cariello, J. R. Cooperrider and others, "Decreased interhemispheric functional connectivity in autism," *Cerebral cortex,* vol. 21, pp. 1134-1146, 2010.

[106] T. Itahashi, T. Yamada, H. Watanabe, M. Nakamura, D. Jimbo, S. Shioda, K. Toriizuka, N. Kato and R. Hashimoto, "Altered network topologies and hub organization in adults with autism: a resting-state fMRI study," *PloS one,* vol. 9, p. e94115, 2014.

[107] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski and J. Kertesz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E,* vol. 75, p. 027105, 2007.

[108] J.-P. Onnela, J. Saramäki, J. Kertész and K. Kaski, "Intensity and coherence of motifs in weighted complex networks," *Physical Review E,* vol. 71, p. 065103, 2005.

[109] U. Brandes and D. Fleischer, "Centrality measures based on current flow," in *Annual symposium on theoretical aspects of computer science*, 2005.

[110] M. Assaf, K. Jagannathan, V. D. Calhoun, L. Miller, M. C. Stevens, R. Sahl, J. G. O'boyle, R. T. Schultz and G. D. Pearlson, "Abnormal functional connectivity of default mode sub-networks in autism spectrum disorder patients," *Neuroimage,* vol. 53, pp. 247-256, 2010.

[111] S. D. Washington, E. M. Gordon, J. Brar, S. Warburton, A. T. Sawyer, A. Wolfe, E. R. Mease-Ference, L. Girton, A. Hailu, J. Mbwana and others, "Dysmaturation of the default mode network in autism," *Human brain mapping,* vol. 35, pp. 1284-1296, 2014.

[112] J. Radua, E. Via, M. Catani and D. Mataix-Cols, "Voxel-based meta-analysis of regional white-matter volume differences in autism spectrum disorder versus healthy controls," *Psychological medicine,* vol. 41, pp. 1539-1550, 2011.

[113] G. D. Waiter, J. H. G. Williams, A. D. Murray, A. Gilchrist, D. I. Perrett and A. Whiten, "Structural white matter deficits in high-functioning individuals with autistic spectrum disorder: a voxel-based investigation," *Neuroimage,* vol. 24, pp. 455-461, 2005.

[114] B. E. Yerys, E. M. Gordon, D. N. Abrams, T. D. Satterthwaite, R. Weinblatt, K. F. Jankowski, J. Strang, L. Kenworthy, W. D. Gaillard and C. J. Vaidya, "Default mode network segregation and social deficits in autism spectrum disorder: Evidence from non-medicated children," *NeuroImage: Clinical,* vol. 9, pp. 223-232, 2015.

[115] J. L. P. Velazquez, F. Barcelo, Y. Hung, Y. Leshchenko, V. Nenadovic, J. Belkas, V. Raghavan, J. Brian and L. G. Dominguez, "Decreased brain coordinated activity in autism spectrum disorders during executive tasks: reduced long-range synchronization in the fronto-parietal networks," *International Journal of Psychophysiology,* vol. 73, pp. 341-349, 2009.

[116] A. E. Richard and R. Lajiness-O'Neill, "Visual attention shifting in autism spectrum disorders," *Journal of clinical and experimental neuropsychology,* vol. 37, pp. 671-687, 2015.

[117] W. Jones and A. Klin, "Attention to eyes is present but in decline in 2--6-month-old infants later diagnosed with autism," *Nature,* vol. 504, p. 427, 2013.

[118] D. Sridharan, D. J. Levitin and V. Menon, "A critical role for the right fronto-insular cortex in switching between central-executive and default-mode networks," *Proceedings of the National Academy of Sciences,* vol. 105, pp. 12569-12574, 2008.

[119] A. C. Chen, D. J. Oathes, C. Chang, T. Bradley, Z.-W. Zhou, L. M. Williams, G. H. Glover, K. Deisseroth and A. Etkin, "Causal interactions between fronto-parietal central executive and default-mode networks in humans," *Proceedings of the National Academy of Sciences,* vol. 110, pp. 19944-19949, 2013.

[120] C. Lord, M. L. Rutter, S. Goode, J. Heemsbergen, H. Jordan, L. Mawhood and E. Schopler, "Autism diagnostic observation schedule: a standardized observation of communicative and social behavior.," *Journal of autism and developmental disorders,* 1989.

[121] C. Lord, M. Rutter and A. Le Couteur, "Autism Diagnostic Interview-Revised: a revised version of a diagnostic interview for caregivers of individuals with possible pervasive developmental disorders," *Journal of autism and developmental disorders,* vol. 24, pp. 659-685, 1994.

[122] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis and D. Feng, "Early diagnosis of Alzheimer's disease with deep learning," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, 2014.

[123] C.-Y. Wee, P.-T. Yap, D. Zhang, L. Wang and D. Shen, "Constrained sparse functional

connectivity networks for MCI classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2012.

[124] R. Cuingnet, E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali, O. Colliot, A. D. N. Initiative and others, "Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database," *neuroimage,* vol. 56, pp. 766-781, 2011.

[125] C.-Y. Wee, L. Wang, F. Shi, P.-T. Yap and D. Shen, "Diagnosis of autism spectrum disorders using regional and interregional morphological features," *Human brain mapping,* vol. 35, pp. 3414-3430, 2014.

[126] S. Ghiassian, R. Greiner, P. Jin and M. Brown, "Learning to classify psychiatric disorders based on fMR images: Autism vs healthy and ADHD vs healthy," in *Proceedings of 3rd NIPS Workshop on Machine Learning and Interpretation in NeuroImaging*, 2013.

[127] Y. Jiao, R. Chen, X. Ke, K. Chu, Z. Lu and E. H. Herskovits, "Predictive models of autism spectrum disorder based on brain regional cortical thickness," *Neuroimage,* vol. 50, pp. 589-599, 2010.

[128] T. Iidaka, "Resting state functional magnetic resonance imaging and neural network classified autism and control," *Cortex,* vol. 63, pp. 55-67, 2015.

[129] M. Plitt, K. A. Barnes and A. Martin, "Functional connectivity classification of autism identifies highly predictive brain features but falls short of biomarker standards," *NeuroImage: Clinical,* vol. 7, pp. 359-366, 2015.

[130] P. Kassraian-Fard, C. Matthis, J. H. Balsters, M. H. Maathuis and N. Wenderoth, "Promises, pitfalls, and basic guidelines for applying machine learning classifiers to psychiatric imaging data, with autism as an example," *Frontiers in psychiatry,* vol. 7, p. 177, 2016.

[131] B. Miao and Y. Zhang, "A feature selection method for classification of ADHD," in *Information, Cybernetics and Computational Social Systems (ICCSS), 2017 4th International Conference on*, 2017.

[132] M. Clerc, Particle swarm optimization, vol. 93, John Wiley & Sons, 2010.

[133] X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern recognition letters,* vol. 28, pp. 459-471, 2007.

[134] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning,* vol. 46, pp. 389-422, 2002.

[135] Y. LeCun, J. S. Denker and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990.

[136] D. Cai, X. He, J. Han and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1548--1560, 2011.

APPENDIX A
A LIST OF REGIONS OF INTEREST

Table 1 A list of ROIs and their coordinates, anatomical and functional labels

ROI	x, y, z coordinates in MNI152 space			Anatomical label	Functional label
1	17	-91	-14	Right Occipital Pole	Visual network
2	8	-91	-7	Right Occipital Pole	Visual network
3	-7	-71	42	Left Precuneous Cortex	Default mode network
4	15	-63	26	Right Precuneous Cortex	Visual network
5	-12	-95	-13	Left Occipital Pole	Visual network
6	26	-79	-16	Right Occipital Fusiform Gyrus	Visual network
7	6	-72	24	Right Cuneal Cortex	Visual network
8	-40	-88	-6	Left Lateral Occipital Cortex inferior division	Visual network
9	11	-66	42	Right Precuneous Cortex	Visual network
10	-26	-90	3	Left Lateral Occipital Cortex inferior division	Visual network
11	-25	-98	-12	Left Occipital Pole	Visual network
12	27	-97	-13	Right Occipital Pole	Visual network
13	-24	-91	19	Left Occipital Pole	Visual network
14	37	-81	1	Right Lateral Occipital Cortex inferior division	Visual network
15	-33	-79	-13	Left Occipital Fusiform Gyrus	Visual network
16	-18	-76	-24	Left Occipital Fusiform Gyrus	Visual network
17	6	-81	6	Right Intracalcarine Cortex	Visual network
18	20	-86	-2	Right Occipital Fusiform Gyrus	Visual network
19	43	-72	28	Right Lateral Occipital Cortex superior division	Visual network
20	-8	-81	7	Left Intracalcarine Cortex	Visual network
21	24	-87	24	Right Lateral Occipital Cortex superior division	Visual network
22	-14	-91	31	Left Occipital Pole	Visual network
23	-3	-81	21	Left Cuneal Cortex	Visual network
24	33	-53	44	Right Superior Parietal Lobule	Visual network
25	27	-59	-9	Right Temporal Occipital Fusiform Cortex	Visual network
26	-28	-79	19	Left Lateral Occipital Cortex superior division	Visual network
27	29	-77	25	Right Lateral Occipital Cortex superior division	Visual network
28	-27	-71	37	Left Lateral Occipital Cortex superior division	Visual network
29	-16	-77	34	Left Cuneal Cortex	Visual network

30	37	-84	13	Right Lateral Occipital Cortex superior division	Visual network
31	8	-72	11	Right Intracalcarine Cortex	Visual network
32	43	-78	-12	Right Lateral Occipital Cortex inferior division	Visual network
33	-42	-60	-9	Left Inferior Temporal Gyrus temporooccipital part	Visual network
34	15	-87	37	Right Occipital Pole	Visual network
35	27	-37	-13	Right Parahippocampal Gyrus posterior division	Visual network
36	58	-53	-14	Right Inferior Temporal Gyrus temporooccipital part	Visual network
37	15	-77	31	Right Cuneal Cortex	Visual network
38	20	-66	2	Right Intracalcarine Cortex	Visual network
39	-28	-58	48	Left Superior Parietal Lobule	Visual network
40	-47	-76	-10	Left Lateral Occipital Cortex inferior division	Visual network
41	-18	-68	5	Left Intracalcarine Cortex	Visual network
42	46	-47	-17	Right Temporal Occipital Fusiform Cortex	Visual network
43	42	-66	-8	Right Lateral Occipital Cortex inferior division	Visual network
44	-47	-51	-21	Left Inferior Temporal Gyrus temporooccipital part	Visual network
45	18	-47	-10	Right Lingual Gyrus	Visual network
46	-15	-72	-8	Left Lingual Gyrus	Visual network
47	-16	-52	-1	Left Lingual Gyrus	Visual network
48	-42	-74	0	Left Lateral Occipital Cortex inferior division	Visual network
49	4	-48	51	Right Precuneous Cortex	Default mode network
50	40	-72	14	Right Lateral Occipital Cortex inferior division	Visual network
51	22	-65	48	Right Lateral Occipital Cortex superior division	Visual network
52	42	0	47	Right Precentral Gyrus	Frontal-parietal network
53	25	-58	60	Right Lateral Occipital Cortex superior division	Visual network
54	46	-59	4	Right Middle Temporal Gyrus temporooccipital part	Visual network
55	-38	-27	69	Left Postcentral Gyrus	Somatosensory network
56	-38	-15	69	Left Precentral Gyrus	Somatosensory

					network
57	-23	-30	72	Left Postcentral Gyrus	Somatosensory network
58	13	-33	75	Right Postcentral Gyrus	Somatosensory network
59	-13	-17	75	Left Precentral Gyrus	Somatosensory network
60	-40	-19	54	Left Precentral Gyrus	Somatosensory network
61	29	-17	71	Right Precentral Gyrus	Somatosensory network
62	2	-28	60	Right Precentral Gyrus	Somatosensory network
63	33	-12	-34	Right Temporal Fusiform Cortex posterior division	Default mode network
64	-16	-46	73	Left Postcentral Gyrus	Somatosensory network
65	-7	-33	72	Left Postcentral Gyrus	Somatosensory network
66	42	-20	55	Right Postcentral Gyrus	Somatosensory network
67	-7	-21	65	Left Precentral Gyrus	Somatosensory network
68	-21	-31	61	Left Postcentral Gyrus	Somatosensory network
69	66	-8	25	Right Postcentral Gyrus	Somatosensory network
70	10	-17	74	Right Precentral Gyrus	Somatosensory network
71	-37	-29	-26	Left Temporal Fusiform Cortex posterior division	Somatosensory network
72	20	-29	60	Right Precentral Gyrus	Somatosensory network
73	-31	-10	-36	Left Temporal Fusiform Cortex anterior division	Default mode network
74	10	-46	73	Right Postcentral Gyrus	Somatosensory network
75	22	-42	69	Right Superior Parietal Lobule	Somatosensory network
76	3	-17	58	Right Precentral Gyrus	Somatosensory network
77	50	-20	42	Right Postcentral Gyrus	Somatosensory network

78	38	-17	45	Right Precentral Gyrus	Somatosensory network
79	-29	-43	61	Left Superior Parietal Lobule	Somatosensory network
80	29	-39	59	Right Superior Parietal Lobule	Somatosensory network
81	-16	-65	-20	Left VI	Visual network
82	52	-34	-27	Right Inferior Temporal Gyrus posterior division	Frontal-parietal network
83	22	-58	-23	Right VI	Somatosensory network
84	1	-62	-18	Vermis VI	Somatosensory network
85	-14	-18	40	Left Precentral Gyrus	Somatosensory network
86	-49	-11	35	Left Precentral Gyrus	Somatosensory network
87	-53	-10	24	Left Postcentral Gyrus	Somatosensory network
88	51	-6	32	Right Precentral Gyrus	Somatosensory network
89	-17	-59	64	Left Lateral Occipital Cortex superior division	Somatosensory network
90	-54	-23	43	Left Postcentral Gyrus	Somatosensory network
91	43	-23	20	Right Parietal Operculum Cortex	Somatosensory network
92	-55	-40	14	Left Planum Temporale	Somatosensory network
93	36	-9	14	Right Insular Cortex	Somatosensory network
94	-56	-45	-24	Left Inferior Temporal Gyrus temporooccipital part	Default mode network
95	-45	-32	47	Left Postcentral Gyrus	Somatosensory network
96	-38	-33	17	Left Planum Temporale	Somatosensory network
97	44	-8	57	Right Precentral Gyrus	Somatosensory network
98	-53	-22	23	Left Central Opercular Cortex	Somatosensory network
99	0	-15	47	Left Cingulate Gyrus posterior division	Somatosensory network

100	11	-39	50	Right Precuneous Cortex	Somatosensory network
101	-49	-26	5	Left Planum Temporale	Somatosensory network
102	-55	-9	12	Left Central Opercular Cortex	Somatosensory network
103	-5	-28	-4	Brain-Stem	Frontal-parietal network
104	58	-16	7	Right Planum Temporale	Somatosensory network
105	32	-26	13	Right Insular Cortex	Somatosensory network
106	56	-5	13	Right Central Opercular Cortex	Somatosensory network
107	10	-62	61	Right Lateral Occipital Cortex superior division	Visual network
108	-7	-52	61	Left Precuneous Cortex	Somatosensory network
109	10	-2	45	Right Cingulate Gyrus anterior division	Somatosensory network
110	-10	-2	42	Left Juxtapositional Lobule Cortex	Somatosensory network
111	-30	-27	12	Left Insular Cortex	Somatosensory network
112	59	-17	29	Right Postcentral Gyrus	Somatosensory network
113	19	-8	64	Right Superior Frontal Gyrus	Somatosensory network
114	-32	-55	-25	Left VI	Visual network
115	-60	-25	14	Left Planum Temporale	Somatosensory network
116	-52	-63	5	Left Lateral Occipital Cortex inferior division	Somatosensory network
117	47	-30	49	Right Postcentral Gyrus	Somatosensory network
118	6	-24	0	Right Thalamus	Frontal-parietal network
119	29	-5	54	Right Precentral Gyrus	Somatosensory network
120	-31	-11	0	Left Putamen	Somatosensory network
121	-16	-5	71	Left Superior Frontal Gyrus	Somatosensory network

122	-50	-34	26	Left Parietal Operculum Cortex	Somatosensory network
123	12	-17	8	Right Thalamus	Somatosensory network
124	-10	-18	7	Left Thalamus	Somatosensory network
125	31	-14	2	Right Putamen	Somatosensory network
126	54	-28	34	Right Supramarginal Gyrus anterior division	Somatosensory network
127	13	-1	70	Right Superior Frontal Gyrus	Somatosensory network
128	65	-33	20	Right Superior Temporal Gyrus posterior division	Somatosensory network
129	-33	-46	47	Left Superior Parietal Lobule	Visual network
130	-45	0	9	Left Central Opercular Cortex	Somatosensory network
131	-3	2	53	Left Juxtapositional Lobule Cortex	Somatosensory network
132	29	1	4	Right Putamen	Somatosensory network
133	37	1	-4	Right Insular Cortex	Somatosensory network
134	7	8	51	Right Juxtapositional Lobule Cortex	Frontal-parietal network
135	49	8	-1	Right Central Opercular Cortex	Frontal-parietal network
136	-51	8	-2	Left Central Opercular Cortex	Frontal-parietal network
137	-34	3	4	Left Insular Cortex	Frontal-parietal network
138	36	10	1	Right Insular Cortex	Frontal-parietal network
139	-1	15	44	Left Paracingulate Gyrus	Frontal-parietal network
140	23	10	1	Right Putamen	Frontal-parietal network
141	-42	38	21	Left Frontal Pole	Frontal-parietal network
142	31	33	26	Right Middle Frontal Gyrus	Frontal-parietal network
143	36	22	3	Right Insular Cortex	Frontal-parietal network

144	-32	-1	54	Left Middle Frontal Gyrus	Frontal-parietal network
145	-35	20	0	Left Insular Cortex	Frontal-parietal network
146	47	10	33	Right Precentral Gyrus	Frontal-parietal network
147	15	5	7	Right Pallidum	Frontal-parietal network
148	-5	18	34	Left Cingulate Gyrus anterior division	Frontal-parietal network
149	-47	11	23	Left Inferior Frontal Gyrus pars opercularis	Frontal-parietal network
150	10	22	27	Right Cingulate Gyrus anterior division	Frontal-parietal network
151	-39	51	17	Left Frontal Pole	Frontal-parietal network
152	38	43	15	Right Frontal Pole	Frontal-parietal network
153	-22	7	-5	Left Putamen	Frontal-parietal network
154	5	23	37	Right Paracingulate Gyrus	Frontal-parietal network
155	37	32	-2	Right Frontal Orbital Cortex	Frontal-parietal network
156	-21	41	-20	Left Frontal Pole	Frontal-parietal network
157	24	32	-18	Right Frontal Orbital Cortex	Frontal-parietal network
158	9	-4	6	Right Thalamus	Frontal-parietal network
159	-23	11	64	Left Superior Frontal Gyrus	Frontal-parietal network
160	43	49	-2	Right Frontal Pole	Frontal-parietal network
161	24	45	-15	Right Frontal Pole	Frontal-parietal network
162	49	-42	45	Right Supramarginal Gyrus posterior division	Frontal-parietal network
163	-41	6	33	Left Middle Frontal Gyrus	Frontal-parietal network
164	-11	26	25	Left Cingulate Gyrus anterior division	Frontal-parietal network
165	48	25	27	Right Middle Frontal Gyrus	Frontal-parietal

					network
166	48	22	10	Right Inferior Frontal Gyrus pars triangularis	Frontal-parietal network
167	34	16	-8	Right Insular Cortex	Frontal-parietal network
168	-15	4	8	Left Caudate	Frontal-parietal network
169	-42	25	30	Left Middle Frontal Gyrus	Frontal-parietal network
170	34	54	-13	Right Frontal Pole	Frontal-parietal network
171	31	56	14	Right Frontal Pole	Frontal-parietal network
172	26	50	27	Right Frontal Pole	Frontal-parietal network
173	0	30	27	Left Cingulate Gyrus anterior division	Frontal-parietal network
174	-42	45	-2	Left Frontal Pole	Frontal-parietal network
175	55	-45	37	Right Supramarginal Gyrus posterior division	Frontal-parietal network
176	-3	26	44	Left Paracingulate Gyrus	Frontal-parietal network
177	-34	55	4	Left Frontal Pole	Frontal-parietal network
178	-28	52	21	Left Frontal Pole	Frontal-parietal network
179	12	36	20	Right Cingulate Gyrus anterior division	Default mode network
180	-2	38	36	Left Paracingulate Gyrus	Default mode network
181	-53	-49	43	Left Supramarginal Gyrus posterior division	Default mode network
182	40	18	40	Right Middle Frontal Gyrus	Default mode network
183	-3	42	16	Left Cingulate Gyrus anterior division	Default mode network
184	44	-53	47	Right Angular Gyrus	Default mode network
185	-42	-55	45	Left Angular Gyrus	Default mode network
186	34	38	-12	Right Frontal Pole	Frontal-parietal network
187	32	14	56	Right Middle Frontal Gyrus	Frontal-parietal network
188	-44	2	46	Left Precentral Gyrus	Default mode network
189	-10	11	67	Left Superior Frontal Gyrus	Default mode network

190	27	16	-17	Right Frontal Orbital Cortex	Default mode network
191	55	-31	-17	Right Inferior Temporal Gyrus posterior division	Default mode network
192	-49	25	-1	Left Frontal Operculum Cortex	Default mode network
193	53	33	1	Right Inferior Frontal Gyrus pars triangularis	Frontal-parietal network
194	-20	45	39	Left Frontal Pole	Default mode network
195	-11	45	8	Left Cingulate Gyrus anterior division	Default mode network
196	-2	-13	12	Left Thalamus	Default mode network
197	22	39	39	Right Frontal Pole	Default mode network
198	65	-31	-9	Right Middle Temporal Gyrus posterior division	Default mode network
199	8	42	-5	Right Paracingulate Gyrus	Default mode network
200	49	35	-12	Right Frontal Pole	Default mode network
201	54	-43	22	Right Angular Gyrus	Frontal-parietal network
202	2	-24	30	Right Cingulate Gyrus posterior division	Default mode network
203	37	-65	40	Right Lateral Occipital Cortex superior division	Default mode network
204	-8	48	23	Left Paracingulate Gyrus	Default mode network
205	-2	-37	44	Left Cingulate Gyrus posterior division	Default mode network
206	-31	19	-19	Left Frontal Orbital Cortex	Default mode network
207	-16	29	53	Left Superior Frontal Gyrus	Default mode network
208	13	30	59	Right Superior Frontal Gyrus	Default mode network
209	9	54	3	Right Paracingulate Gyrus	Default mode network
210	-7	51	-1	Left Paracingulate Gyrus	Default mode network
211	56	-46	11	Right Middle Temporal Gyrus temporooccipital part	Somatosensory network
212	-20	64	19	Left Frontal Pole	Default mode network
213	65	-24	-19	Right Middle Temporal Gyrus posterior division	Default mode network
214	-35	20	51	Left Middle Frontal Gyrus	Default mode network
215	35	-67	-34	Right Crus I	Default mode network
216	23	33	48	Right Superior Frontal Gyrus	Default mode network
217	6	54	16	Right Paracingulate Gyrus	Default mode network
218	51	-29	-4	Right Middle Temporal Gyrus posterior division	Default mode network
219	13	55	38	Right Frontal Pole	Default mode network
220	-46	31	-13	Left Frontal Orbital Cortex	Default mode network
221	52	-33	8	Right Superior Temporal Gyrus	Somatosensory

				posterior division	network
222	47	-50	29	Right Angular Gyrus	Default mode network
223	-56	-50	10	Left Middle Temporal Gyrus temporooccipital part	Somatosensory network
224	52	-59	36	Right Lateral Occipital Cortex superior division	Default mode network
225	-68	-41	-5	Left Middle Temporal Gyrus posterior division	Default mode network
226	-58	-30	-4	Left Middle Temporal Gyrus posterior division	Default mode network
227	-10	39	52	Left Superior Frontal Gyrus	Default mode network
228	6	64	22	Right Frontal Pole	Default mode network
229	-39	-75	44	Left Lateral Occipital Cortex superior division	Default mode network
230	-2	-35	31	Left Cingulate Gyrus posterior division	Default mode network
231	-18	63	-9	Left Frontal Pole	Default mode network
232	8	41	-24	Right Frontal Medial Cortex	Default mode network
233	-49	-42	1	Left Middle Temporal Gyrus posterior division	Default mode network
234	-41	-75	26	Left Lateral Occipital Cortex superior division	Default mode network
235	17	-80	-34	Right Crus II	Default mode network
236	-58	-26	-15	Left Middle Temporal Gyrus posterior division	Default mode network
237	-10	55	39	Left Frontal Pole	Default mode network
238	-3	44	-9	Left Paracingulate Gyrus	Default mode network
239	8	48	-15	Right Frontal Medial Cortex	Default mode network
240	-44	-65	35	Left Lateral Occipital Cortex superior division	Default mode network
241	-68	-23	-16	Left Middle Temporal Gyrus posterior division	Default mode network
242	28	-77	-32	Right Crus I	Default mode network
243	-34	-38	-16	Left Temporal Fusiform Cortex posterior division	Default mode network
244	6	67	-4	Right Frontal Pole	Default mode network
245	-46	-61	21	Left Lateral Occipital Cortex superior division	Default mode network
246	-13	-40	1	Left Hippocampus	Default mode network
247	52	7	-30	Right Temporal Pole	Default mode network
248	49	-3	-38	Right Inferior Temporal Gyrus anterior division	Default mode network
249	17	-28	-17	Right Parahippocampal Gyrus	Default mode network

				posterior division	
250	65	-12	-19	Right Middle Temporal Gyrus posterior division	Default mode network
251	52	-2	-16	Right Superior Temporal Gyrus anterior division	Default mode network
252	46	16	-30	Right Temporal Pole	Default mode network
253	6	-59	35	Right Precuneous Cortex	Default mode network
254	-26	-40	-8	Left Lingual Gyrus	Default mode network
255	8	-48	31	Right Cingulate Gyrus posterior division	Default mode network
256	-3	-49	13	Left Cingulate Gyrus posterior division	Default mode network
257	-21	-22	-20	Left Parahippocampal Gyrus anterior division	Default mode network
258	-50	-7	-39	Left Inferior Temporal Gyrus anterior division	Default mode network
259	-53	3	-27	Left Middle Temporal Gyrus anterior division	Default mode network
260	11	-54	17	Right Precuneous Cortex	Default mode network
261	-56	-13	-10	Left Middle Temporal Gyrus posterior division	Default mode network
262	-7	-55	27	Left Precuneous Cortex	Default mode network
263	-11	-56	16	Left Precuneous Cortex	Default mode network
264	-44	12	-34	Left Temporal Pole	Default mode network

APPENDIX B
PROGRAMS

Programs for Network Clustering

JSNMF in MATLAB

```
function H=MF4CNDA(A, k, errs, alpha)

% A is a 3-dimentional matrix containing all the individual networks.
% k is the number of clusters desired
% alpha is the regularization factor

n=size(A,3); % the number of samples
m=size(A,1); % the dimension of similarity matrix of each sample;
H=0.01*abs(randn(m, k)); %initialization
S=0.01*abs(randn(k,k,n));

for i=1:n
    S(:,:,i)=S(:,:,i)+S(:,:,i)';
end
Serr10=10000;
err0=1;
while err0>errs
     H1=H'*H;
     NU=zeros(m,k);
     DU=zeros(m,k);
     for i=1:n
          NUS = H'*A(:,:,i)*H;
          DUS = H1*S(:,:,i)*H1;
          S(:,:,i)=S(:,:,i).*(NUS./DUS);
          NU=NU+A(:,:,i)*H*S(:,:,i);
          DU=DU+H*S(:,:,i)*H1*S(:,:,i);
     end
     H=H.*(NU./(DU+alpha)).^(1/4);
     Serr1=0;
     for i=1:n
          Serr1 = Serr1+ (norm(A(:,:,i)-H*S(:,:,i)*H', 'fro'))^2;
     end
     Serr1=Serr1+alpha*sum(sum(H));
    err0=abs(Serr10-Serr1)/Serr10; %calculate the error
    Serr10=Serr1;
end
```

SNMF in MATLAB

```
function [iter,OBJ, U] = SNMFWU(A,U,tol,maxiter)

% A is the connectivity matrix and U is the feature matrix initilized
% in the main script.

n=length(A(1,:));
r=length(U(1,:));
err0=tol;

OBJinit=0.5*sum(sum((A-U*U').^2));

for iter=1:maxiter,
    for k=1:r
        NU=U(:,k)'*U(:,k);
        for i=1:n
            Uik0=U(i,k);
            bi=A(i,i)-U(i,:)*U(i,:)';
            if(NU<err0 & bi>=0)
                U(i,k)=sqrt(bi);
            else
                gik=(U*U(i,:)'-A(:,i))'*U(:,k);
                d=abs(gik)/NU;
                Dik=max(0, -bi+U(i,k)^2+2*U(i,k)*d+d^2/2);
                DD=2*(NU+Dik);
                Uik1=U(i,k)-gik/DD;
                U(i,k)=max(0,Uik1);
            end
            NU=NU-Uik0^2+U(i,k)^2;
        end
    end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  OBJ=0.5*sum(sum((A-U*U').^2));
  err=abs(OBJinit-OBJ)/OBJ;
  OBJinit=OBJ;
  if (err < err0)
     break;
  end
end
```

SC in MATLAB

```
function [group, eigengap] = SpectralClustering(W, NUMC)
```

```
% calculate degree matrix
degs = sum(W, 2);
D       = sparse(1:size(W, 1), 1:size(W, 2), degs);

% compute unnormalized Laplacian
L = D - W;
k = max(NUMC);
% compute normalized Laplacian if needed

% avoid dividing by zero
degs(degs == 0) = eps;
% calculate D^(-1/2)
D = spdiags(1./(degs.^0.5), 0, size(D, 1), size(D, 2));
% calculate normalized Laplacian
L = D * L * D;

% compute the eigenvectors corresponding to the k smallest
% eigenvalues
[U, eigenvalue] = eigs(L, k, eps);
[a,b] = sort(diag(eigenvalue),'ascend');
eigenvalue = eigenvalue(:,b);
U = U(:,b);
eigengap = abs(diff(diag(eigenvalue)));
U = U(:,1:k);
% in case of the Jordan-Weiss algorithm, I need to normalize
% the eigenvectors row-wise
%U = bsxfun(@rdivide, U, sqrt(sum(U.^2, 2)));
%U = U./repmat(sqrt(sum(U.^2,2)),1,size(U,2));

flag =0;
for ck = NUMC
      Cindex = find(NUMC==ck);
      UU = U(:,1:ck);
      UU = UU./repmat(sqrt(sum(UU.^2,2)),1,size(UU,2));
      [EigenvectorsDiscrete]=discretisation(UU);
      [~,temp] = max(EigenvectorsDiscrete,[],2);
%       for i = 1 : ck
%             initcenter(i,:) = mean(UU(temp==i,:));
%       end

      Cluster{Cindex} = temp;
end

if length(NUMC)==1
```

```
        group=Cluster{1};
else
        group = Cluster;
end

end

function [EigenvectorsDiscrete,EigenVectors]=discretisation(EigenVectors)
%
% EigenvectorsDiscrete=discretisation(EigenVectors)
%
% Input: EigenVectors = continuous Ncut vector, size = ndata x nbEigenvectors
% Output EigenvectorsDiscrete = discrete Ncut vector, size = ndata x nbEigenvectors
%
% Timothee Cour, Stella Yu, Jianbo Shi, 2004

[n,k]=size(EigenVectors);

vm = sqrt(sum(EigenVectors.*EigenVectors,2));
EigenVectors = EigenVectors./repmat(vm+eps,1,k);

R=zeros(k);
% R(:,1)=EigenVectors(1+round(rand(1)*(n-1)),:)';
  R(:,1)=EigenVectors(round(n/2),:)';
%R(:,1)=EigenVectors(n,:)';
c=zeros(n,1);
for j=2:k
        c=c+abs(EigenVectors*R(:,j-1));
        [minimum,i]=min(c);
        R(:,j)=EigenVectors(i,:)';
end

lastObjectiveValue=0;
exitLoop=0;
nbIterationsDiscretisation = 0;
nbIterationsDiscretisationMax = 20;%voir
while exitLoop== 0
        nbIterationsDiscretisation = nbIterationsDiscretisation + 1 ;
        EigenvectorsDiscrete = discretisationEigenVectorData(EigenVectors*R);
        [U,S,V] = svd(EigenvectorsDiscrete'*EigenVectors+eps,0);
        NcutValue=2*(n-trace(S));

        if abs(NcutValue-lastObjectiveValue) < eps | nbIterationsDiscretisation >
nbIterationsDiscretisationMax
```

```
            exitLoop=1;
        else
            lastObjectiveValue = NcutValue;
            R=V*U';
        end
end

function Y = discretisationEigenVectorData(EigenVector)
% Y = discretisationEigenVectorData(EigenVector)
%
% discretizes previously rotated eigenvectors in discretisation
% Timothee Cour, Stella Yu, Jianbo Shi, 2004

[n,k]=size(EigenVector);


[Maximum,J]=max(EigenVector');

Y=sparse(1:n,J',1,n,k);
% Y = J';
```

MSC in MATLAB

```
function [U,O_cur]=MultiCoreguSC_CB(L,U,K,lambda)

n=length(L);
L_m=zeros(size(L{1}));
% main circle
    for v=1:n
        L_m=L_m+lambda(v)*U{v}*U{v}';
    end

    [Us,V]=eigs(L_m,K);

    for v=1:n
        [U{v},V]=eigs(L{v}+lambda(v)*Us*Us',K);
    end
    O_cur=Us;
```

Clustering Evaluation Indices in MATLAB

```
function [mdul,cov,cond]=NetMetrics(A,c)

% A is a network and c is the clusteirng to be evaluated
```

```
Nn=length(A(:,1,1));
Nc=max(c);

% Calcuate coverage
num=0;
den=0;
for i=1:Nn
    for j=1:Nn
        if c(i)==c(j)
            num=num+A(i,j);
        end
        den=den+A(i,j);
    end
end
cov=num/den;

% Calculate modularity
W=A/sum(sum(A));
Vol=zeros(Nc,1);
WW=zeros(Nc,Nc);
for i=1:Nn,
    Vol(c(i)) = Vol(c(i))+sum(W(i,:));
    for j=1:Nn,
        WW(c(i), c(j))= WW(c(i), c(j)) + W(i,j);
    end
end
mdul=trace(WW)-Vol'*Vol;

%Calculate conductance
A=triu(A);
for cc=1:Nc
    num=0;
    aS=0;
    aS_bar=0;
    for i=1:Nn
        for j=1:Nn
            if c(i)==cc
                aS=aS+A(i,j);
                if c(i)~=c(j)
                    num=num+A(i,j);
                end
            else
                aS_bar=aS_bar+A(i,j);
```

```
                end
            end
        end
        phiS(cc)=num/min(aS,aS_bar);
end
cond=1-mean(phiS);
```

Classification Algorithms

PSOSVM

```
% Parameter settings for PSO

m=1000;
train=imconn{1,4};
S=length(train(1,:)); %total number of features
VelMax=floor(S/3); %max velocity
VelMin=1; %min velocity
MaxIt=50; %Max iteration
weight=1.4; %Initial inertia weight
c1=2; %acceleration terms for gbest and pbest
c2=2;
alpha=0.9; %weights in cost function
beta=1-alpha;

% Initialization

empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.TP=[];
empty_particle.FP=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
empty_particle.Best.TP=[];
empty_particle.Best.FP=[];

particle=repmat(empty_particle,m,1);

BestSol.Cost=0;

for i=1:m

    % Initialize Position
```

```
        particle(i).Position=randi([0 1],S,1);

        % Initialize Velocity
        particle(i).Velocity=randi([0 VelMax],1,1);

        % Evaluation
        [particle(i).Cost,                                                    particle(i).TP,
particle(i).FP]=CostFunction(particle(i).Position,train,alpha,beta);

        % Update Personal Best
        particle(i).Best.Position=particle(i).Position;
        particle(i).Best.Cost=particle(i).Cost;
        particle(i).Best.TP=particle(i).TP;
        particle(i).Best.FP=particle(i).FP;

        % Update Global Best
        if particle(i).Best.Cost>BestSol.Cost

            BestSol=particle(i).Best;

        end

end


% nn=1;
% PSO Main Loop

for it=1:MaxIt

    weight=(weight-0.4)*(MaxIt-it)/MaxIt+0.4; %recalculate the weight for each iteration,
following Wang et al. 2007

    for i=1:m

        % Calculate distances
        pbestdiff=particle(i).Best.Position-particle(i).Position;
        gbestdiff=BestSol.Position-particle(i).Position;
        pbestdist=length(find(pbestdiff==1))-length(find(pbestdiff==-1));
        gbestdist=length(find(gbestdiff==1))-length(find(gbestdiff==-1));

        % Update Velocity
        particle(i).Velocity = floor(weight*particle(i).Velocity ...
            + c1*rand(1,1)*pbestdist + c2*rand(1,1)*gbestdist);
```

```
            % Apply Velocity Limits
            particle(i).Velocity = max(particle(i).Velocity,VelMin);
            particle(i).Velocity = min(particle(i).Velocity,VelMax);

            % Update Position
            xg=length(find(gbestdiff~=0));
            if particle(i).Velocity<=xg

                ChangedBitsIndex=randsample(find(gbestdiff~=0), particle(i).Velocity);
                for j=1:length(ChangedBitsIndex)


particle(i).Position(ChangedBitsIndex(j))=BestSol.Position(ChangedBitsIndex(j));

                end

            else

                particle(i).Position=BestSol.Position;
                ChangedBitsIndex=randsample(find(gbestdiff==0), particle(i).Velocity);
                for j=1:length(ChangedBitsIndex)

                    particle(i).Position(ChangedBitsIndex(j))=-
(BestSol.Position(ChangedBitsIndex(j)))+1;

                end

            end

            % Evaluation
            [particle(i).Cost,          particle(i).TP,          particle(i).FP]          =
CostFunction(particle(i).Position,train,alpha,beta);
%               nn=nn+1;

            % Update Personal Best
            if particle(i).Cost>particle(i).Best.Cost

                particle(i).Best.Position=particle(i).Position;
                particle(i).Best.Cost=particle(i).Cost;
                particle(i).Best.TP=particle(i).TP;
                particle(i).Best.FP=particle(i).FP;

                % Update Global Best
```

```
            if particle(i).Best.Cost>BestSol.Cost

                BestSol=particle(i).Best;

            end

        end

    end

    BestCost(it)=BestSol.Cost;

    if it>1 && abs(BestCost(it)-BestCost(it-1))/BestCost(it-1)<=0.000001
        break;
    end

end

function [Cost, TP, FP]=CostFunction(Position,train_orig,alpha,beta)

C=length(train_orig(1,:));
train_orig=train_orig(:,find(Position==1));

%leave one out
% This part is in the main script and the same for each classifier.
% I can easily change svmtrain to other classifier training functions such
% as RF, kNN, etc. Some classifiers are implemented in Python, but the
% structure of their programs are also similar.
for s=1:79
    train=train_orig;
    test=train(s,:);
    train(s,:)=[];
    if s<=37
        group=cat(2,zeros(1,36),ones(1,42));
    else
        group=cat(2,zeros(1,37),ones(1,41));
    end
    svmmodel = svmtrain(train,group,'kernel_function','linear','method','LS');
    testscale=(test+svmmodel.ScaleData.shift).*svmmodel.ScaleData.scaleFactor;

svmvalue(s)=sum(svmmodel.SupportVectors*testscale'.*svmmodel.Alpha)+svmmodel.Bias;

end
maxthld=max(svmvalue)+eps;
```

```
minthld=min(svmvalue)-eps;
svmthreshold=minthld:(maxthld-minthld)/50:maxthld;
for t=1:length(svmthreshold)
    for s=1:79
        prediction(t,s)=(svmvalue(s)<svmthreshold(t));

    end
    TP(t)=length(find(prediction(t,38:79)==1))/42;
    FP(t)=length(find(prediction(t,1:37)==1))/37;
%       TN(t)=length(find(prediction(t,1:37)==0))/79;

end
AUC=trapz(FP,TP);
Cost=alpha*AUC+beta*(C-length(find(Position==1)))/C;
```