Double Wavelet Transform for Task-induced and Resting State Functional Magnetic Resonance Imaging Data

By

Minchun Zhou

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Biostatistics

August 31, 2018

Nashville, Tennessee

Approved:

Jeffrey D. Blume , Ph.D.

Hakmook Kang , Ph.D.

Matthew S. Shotwell , Ph.D.

Baxter P. Rogers , Ph.D.

ACKNOWLEDGMENTS

I feel extremely fortunate to have the chance to study at Vanderbilt University and to finish this dissertation.

I would like to give special thanks to Dr. Hakmook Kang for being my adviser. His expertise and humility not only provide me inspiration and guidance on research, but also makes me realize how to be a better person. I would like to thank my committee member Dr. Matthew Shotwell, who opened the door of research for me, and provide support along the way. I would also like to thank my committee member Dr. Jeffrey Blume, Dr. Baxter Rogers to provide valuable suggestions on my research and push my research forward. Besides, I am grateful to all the faculty, staff and graduate students in the Department of Biostatistics, who make this such a wonderful family.

I would like to thank everyone I met during the last six years. Thanks to Dr. Malcolm Avison, Dr. Kevin Niswender and Dr. Heidi Silver, who I worked with for two years. I want to thank my soccer team. We won two intramural titles together, which were one of my best experience at Vanderbilt.

Finally, I would like to thank my parents, and my wife Shu. I'm blessed to have their love and support with me all the time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

In this dissertation, we develop a novel single level double-wavelet framework that takes into account the spatial and temporal correlation at ROI-level for both task-induced and resting state fMRI data analysis. Conventional approaches to fMRI analysis only take into account temporal correlations but do not rigorously model the underlying spatial correlation due to the complexity of estimating and inverting the high dimensional spatio-temporal covariance matrix. Other spatio-temporal model approaches estimate the covariance matrix with the assumption of stationary time series, which is not feasible sometimes. To address these limitations, we propose a double-wavelet approach for modeling the spatio-temporal brain process. Working with wavelet coefficients simplifies temporal and spatial covariance structure because under regularity conditions, wavelet coefficients are approximately uncorrelated. Different wavelet functions were used to capture different correlation structures in the spatio-temporal model. Main advantages of the wavelet approach are that it is scalable and that it deals with non-stationarity in brain signals.

For tasked-induced fMRI data analysis, we applied our method to fMRI data to study activation in pre-specified ROIs in the pre-fontal cortex. Data analysis showed that the result using the double-wavelet approach was more consistent than the conventional approach when sample size decreased. We also developed a MATLAB graphical user interface (GUI) for multi-subject task-induced fMRI data using double-wavelet transform, which can estimate the effect of user-specified stimulus functions and region of interests (ROI). For resting state fMRI data analysis, we applied our method to resting-state fMRI data to study the difference between healthy subjects and patients with major depressive disorder (MDD).

Chapter 1

INTRODUCTION

One of the most challenging problem in contemporary science is to understand the human brain and its activity pattern. To acquire good quality data of the human brain function, functional magnetic resonance imaging (fMRI) becomes a powerful tool in the last twenty years. Compared to other methods like PET (Positron Emission Tomography) and MEG (magnetoencephalography), fMRI has reasonable temporal resolution ($1 \sim 2$ seconds) and excellent spatial resolution (in mm^3). Typical fMRI data are collected by scanning the brain every few seconds and indirectly measure brain activity by detecting changes in oxygen level associated with blood flow. There are about 100,000 three-dimensional volume elements called voxels in each fMRI scan.

Several methods have been developed to take into account the spatial correlation in fMRI data. Worsley et al. (1996) proposed spatial smoothing using a Gaussian kernel, which is now commonly used as a preprocessing step in fMRI study. The goal of spatial smoothing is to increase the signal to noise ratio in fMRI data, however it induces more spatial correlation, which may result in invalid statistical inferences. Spatio-temporal modeling is another approach to not only model the spatial correlation but also consider the temporal correlation in fMRI data. Since Fourier coefficients are approximately uncorrelated across frequencies, Ombao et al. (2008) developed a spatio-spectral model to understand the underlying spatio-temporal processes. Based on that, Kang et al. (2012) proposed a spatio-spectral mixed-effects model to simultaneously estimate spatial correlation within a region and correlation across regions.

Brammer (1998) and Ruttimann et al. (1998) first applied wavelet transform on fMRI data. Long et al. (2004) performed wavelet transform on spatial data and calculated temporal noise parameters using iterative methods. Ville et al. (2004) proposed to simultaneously estimated two threshold using the original signal and the wavelet coefficients to denoise the signal. Aston et al. (2005) estimated the spatio-temporal model coefficients using the wavelet coefficients only from spatial data.

In this dissertation, we develop a novel single level double-wavelet framework that takes into account the spatial and temporal correlation at ROI-level for both task-induced and resting state fMRI data analysis.

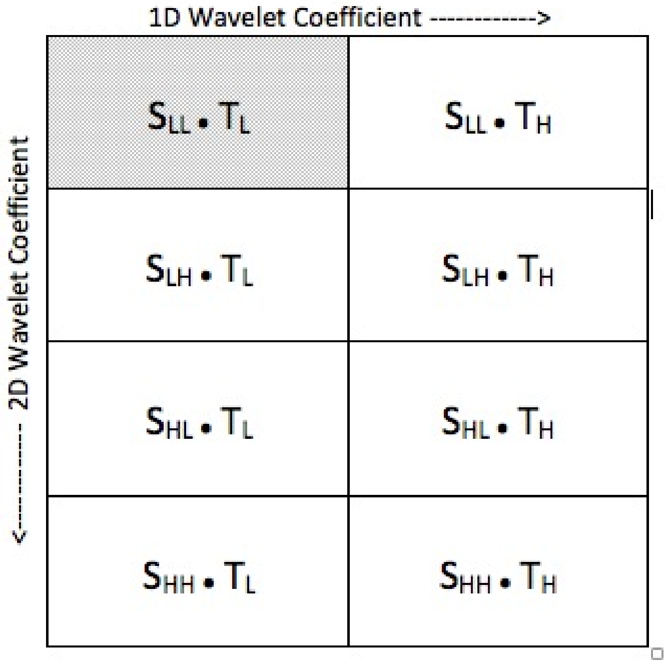
1.1 Wavelet Transform

Similar to Fourier Transform, Wavelet Transform is a linear transform. Wavelet coefficients are obtained by the inner product of the observed data and wavelet functions, which is similar to the sine and cousin functions in Fourier transform. There are two types of wavelet transform, zero integral mother wavelet function and the unit integral father wavelet function. Wavelet transform can be performed into different levels, which corresponding to different time interval length. In this paper, we only discuss single level wavelet transform, which decomposes data into two frequency bands at each dimension.

1.2 Double Wavelet Transform

The main idea of the double-wavelet transform is to apply different wavelet functions to different dimensional data in fMRI data analysis. The order of two wavelet transform is interchangeable since wavelet transform is a linear transform. Suppose we have 4-D fMRI data (1-D time series and 3-D volume). First we can obtain 3-D wavelet coefficients by applying 3-D wavelet transform on 3-D column data at each time point. For each 3-D wavelet transform, we have a time series of it. Second we apply 1-D wavelet transform on the time series of the time series of each 3-D wavelet coefficient. We also apply the 1-D wavelet transform on the stimulus function.

The wavelet transform naturally decomposes data into different scales. Each scale corresponds to different frequency bands for both spatial and temporal data. The single



1D Wavelet Coefficient ------------>

<------------ 2D Wavelet Coefficient

$S_{LL} \bullet T_L$	$S_{LL} \bullet T_H$
$S_{LH} \bullet T_L$	$S_{LH} \bullet T_H$
$S_{HL} \bullet T_L$	$S_{HL} \bullet T_H$
$S_{HH} \bullet T_L$	$S_{HH} \bullet T_H$

Figure 1.1: S_{LL}, S_{LH}, S_{HL} and S_{HH} represent the wavelet coefficients from the 2-D wavelet transform on the spatial data in low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH) respectively; T_L and T_H represent the wavelet coefficients from the 1-D wavelet transform on the temporal data in low and high frequency band respectively.

level discrete wavelet transform (SL-DWT) decomposes data into two frequency bands at each dimension. For example, a 1-D signal would be transformed into wavelet coefficients indicating information from one high frequency band and one low frequency band by SL-DWT. A 2-D image would be decomposed into high and low frequency bands on both vertical and horizontal directions by SL-DWT, which results into four parts, low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH). Figure 1.1 shows the double wavelet coefficients, where S indicates the wavelet coefficients from the 2-D wavelet transform on the spatial data and T indicates the wavelet coefficients from the 1-D wavelet transform on the temporal data.

Figure 1.2 (a) is an example using the Daubechies 3 wavelet (spatial wavelet function) and the Symlet 8 wavelet (temporal wavelet function) on an activated 3-D ROI data ($\beta_1 - \beta_2 > 0$) in simulation (2-D in spatial domain and 1-D in temporal domain). Figure 1.2 (b) is an example using Reverse Biorthogonal 3.1 wavelet as spatial wavelet and Haar wavelet as temporal wavelet on a simulated 3-D resting state fMRI ROI data (2-D in spatial domain and 1-D in temporal domain).

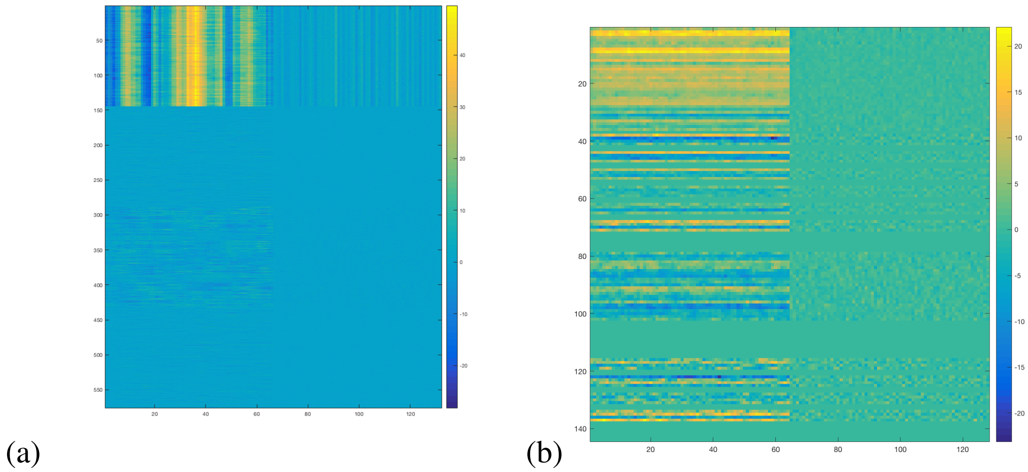


Figure 1.2: (a) is an example using the Daubechies 3 wavelet (spatial wavelet function) and the Symlet 8 wavelet (temporal wavelet function) on an activated 3-D ROI data ($\beta_1 - \beta_2 > 0$) in simulation (2-D in spatial domain and 1-D in temporal domain). (b) is an example using Reverse Biorthogonal 3.1 wavelet as spatial wavelet and Haar wavelet as temporal wavelet on a simulated 3-D resting state fMRI ROI data (2-D in spatial domain and 1-D in temporal domain).

Chapter 2

DOUBLE-WAVELET TRANSFORM FOR MULTI-SUBJECT TASK-INDUCED FUNCTIONAL MAGNETIC RESONANCE IMAGING DATA

The goal of this article is to model multi-subject task-induced fMRI response among predefined regions of interest (ROIs) of the human brain. Conventional approaches to fMRI analysis only take into account temporal correlations but do not rigorously model the underlying spatial correlation due to the complexity of estimating and inverting the high dimensional spatio-temporal covariance matrix. Other spatio-temporal model approaches estimate the covariance matrix with the assumption of stationary time series, which is not feasible sometimes. To address these limitations, we propose a double-wavelet approach for modeling the spatio-temporal brain process. Working with wavelet coefficients simplifies temporal and spatial covariance structure because under regularity conditions, wavelet coefficients are approximately uncorrelated. Different wavelet functions were used to capture different correlation structures in the spatio-temporal model. Main advantages of the wavelet approach are that it is scalable and that it deals with non-stationarity in brain signals. Simulation studies showed that our method could reduce false positive rates and false negative rates by taking into account spatial and temporal correlations simultaneously. We also applied our method to fMRI data to study activation in pre-specified ROIs in the prefontal cortex. Data analysis showed that the result using the double-wavelet approach was more consistent than the conventional approach when sample size decreased.

2.1 Introduction

For investigating the human brain, functional magnetic resonance imaging (fMRI) is a powerful tool because it has excellent spatial resolution (in mm^3) and reasonable temporal resolution (about 1 second) for capturing the evolution of the brain hemodynamic response.

Typical fMRI data are generated by scanning the brain every few seconds. During each scan, the signal from the brain is measured on three-dimensional volume elements called voxels.

The conventional analysis of fMRI data (AV-GLM) first applies spatial smoothing on the data to increase the signal to noise ratio (SNR) after a series of preprocessing steps, e.g., motion correction, slice timing correction, and co-registration, estimates the parameters from the mean time series of each ROI while taking into account the temporal correlation, e.g., auto-regressive order one (AR(1)) structure, and then uses a simple t-test on the parameters of the model from multiple subjects (Worsley & Friston, 1995; Weiskopf et al., 2003; Huettel et al., 2004). This approach only takes into account the underlying temporal correlation but does not rigorously model the underlying spatial correlation. Dubin (1988) argued that ignoring spatial correlation in data led to smaller standard errors and higher Type I errors.

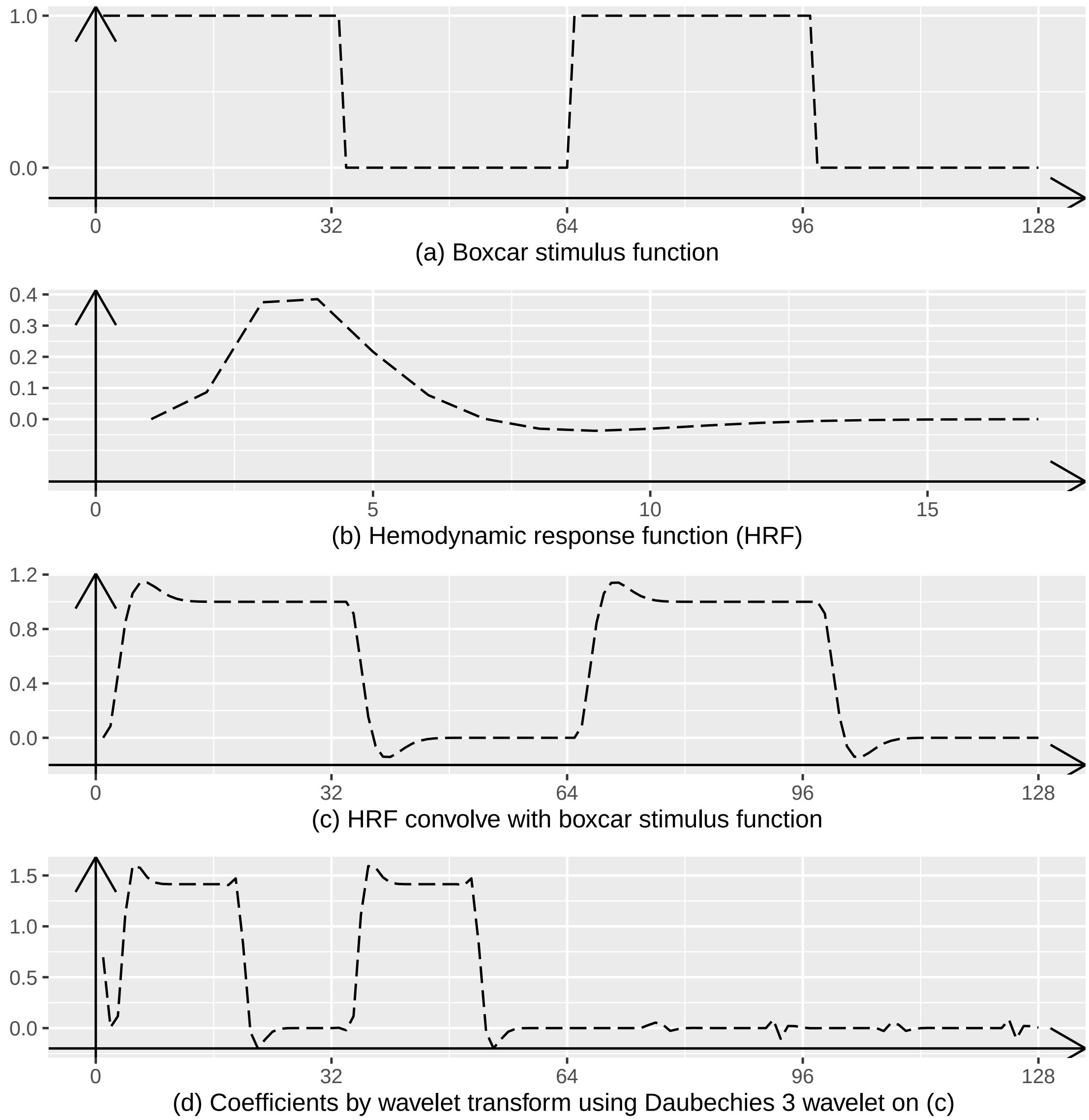
Spatial smoothing and spatio-temporal modeling are two common approaches that take into account spatial correlation in fMRI data analysis. Spatial smoothing using a Gaussian kernel was proposed by Worsley et al. (1996). Katanoda et al. (2002) proposed combining information from the six nearest neighboring voxels in the Fourier domain. Though these approaches increased the signal to noise ratio in fMRI data, they actually induce more spatial correlation, which may cause a higher error rate. Ombao et al. (2008) developed a spatio-spectral model using the Fourier bases to understand the underlying spatio-temporal processes. Kang et al. (2012) proposed a spatio-spectral mixed-effects model to estimate local and global spatial correlation. These approaches utilized the fact that the Fourier coefficients are approximately uncorrelated across frequencies. However, these approaches require temporal stationarity of the signals and the estimation and inversion of the covariance matrix, both of which are computationally expensive in fMRI data analysis since there could be over 100,000 voxels.

Recently, Karaman et al. (2014) used linear operators during spatial and temporal preprocessing and reconstruction operations. Degras & Lindquist (2014) developed a spatio-temporal hierarchical model by estimating the hemodynamic response function (HRF) and voxel activation simultaneously. Lindquist et al. (2009); Zhang et al. (2012, 2013, 2014) proposed both parametric and semi-parametric estimation of the HRF. Although these methods relaxed the stationary time series assumption, they still require stationary spatial correlation across voxels. Hyun et al. (2014) proposed a Gaussian predictive process model using a three-stage estimation to model the spatial correlation and cross-correlation simultaneously. Furthermore, Hyun et al. (2016) applied the spatio-temporal Gaussian process on the longitudinal neuroimaging data. However, the key assumptions of the spatio-temporal Gaussian process need to be rigorously validated.

The wavelet transform is a linear transformation similar to the Fourier transform. We start with a wavelet bases consisting of orthonormal functions. The signal being analyzed is represented in terms of the selected bases and the wavelet coefficients are the inner product (cross-correlation) between the observed signal and each of the wavelet bases functions. Fan (2003) proved that the discrete wavelet transform (DWT) coefficients of both stationary and non-stationary signals were approximately uncorrelated as long as the length of the wavelet filter was large enough and the length of signals were sufficiently long. The wavelet transform was first introduced to fMRI analysis by Brammer (1998) and Ruttimann et al. (1998). Brammer proposed manipulating the wavelet coefficients in the spatial domain and reconstructing the original data to optimize the detection of activation. Ruttimann discovered that the sum of the square of standardized wavelet coefficients had a χ^2 distribution. The brain signals were reconstructed using only wavelet coefficients with large magnitude (i.e., those that exceed a theoretically-derived threshold). Bullmore et al. (2003) used the decorrelating property of the wavelet transform to control the Type I errors. Long et al. (2004) performed spatio-temporal wavelet analysis for fMRI data by combining the wavelet transform with calculating the temporal noise parameters using iterative methods. To minimize the approximation errors, Ville et al. (2004) proposed using two thresholds.

One threshold was used before reconstruction and the other was used after reconstruction. Two thresholds, one in the wavelet domain and the other in the time domain, were simultaneously estimated to produce the reconstructed signal. Perhaps most closely related to the approach we developed, Aston et al. (2005) proposed estimating the model coefficients in the wavelet domain by applying one wavelet transform on the spatial data at each time point. They transformed the wavelet coefficient residuals back to the spatial domain and used the reconstructed residuals to estimate the variance of the coefficients.

In this article, we develop a novel single level double-wavelet framework that takes into account the spatial and temporal correlation for estimating the ROI-level activation patterns in multi-subject fMRI data analysis. Our approach does not require the estimation of the spatio-temporal covariance matrix and inversion of it, which is a big computational burden in the spatial-temporal model in fMRI data analysis. We first apply one wavelet transform (spatial wavelet function) on the spatial data at each time point, then apply another wavelet transform (temporal wavelet function) on the time series of each spatial wavelet coefficient. Different wavelet functions can be chosen to capture different correlation structures in the spatial and temporal data. The boxcar stimulus function convolved with the HRF is also transformed using the temporal wavelet function. For illustration purpose, a boxcar function, the canonical HRF given by SPM12 (http://www.fil.ion.ucl.ac.uk/spm/), the convolution between two functions, and the wavelet coefficients by performing a wavelet transform using Daubechies 3 wavelet on the convolution are depicted in Figure 2.1. All estimations and inferences were done using wavelet coefficients. It is noteworthy that the order of the computation of the two wavelet transforms has no effect on the result since the wavelet transform is a linear transformation. The double-wavelet transform also simplifies the data structure, where four-dimensional (4-D) data were converted into two-dimensional (2-D) data. We examine the validity of our approach via simulation studies with different spatial correlation structures. Finally we apply our approach to investigate higher cognitive control function in the anterior premotor cortex (prePMD), the lateral prefrontal cortex



(a) Boxcar stimulus function

(b) Hemodynamic response function (HRF)

(c) HRF convolve with boxcar stimulus function

(d) Coefficients by wavelet transform using Daubechies 3 wavelet on (c)

Figure 2.1: A typical boxcar stimulus function is illustrated in (a); the HRF is illustrated in (b); the convolution between the stimulus function and the HRF is in (c), which is used as one of covariates in linear model; (d) shows the wavelet coefficients by performing the single level wavelet transform using Daubechies 3 wavelet on (c). The units of x-axis in (a), (b), (c) and (d) are in seconds.

(PFC), and the primary visual cortex.

2.2 The Wavelet Transform

First, we will introduce the wavelet transform. The wavelet families form a series of orthonormal bases for different dimensional data. Wavelet coefficients are obtained by the inner product of the observed data and wavelet functions which contain the zero integral "mother" wavelet function and the unit integral "father" wavelet function. All other wavelet functions are dilated and shifted from the mother and father wavelet functions. For simplicity, we only introduce the one-dimensional (1-D) and two-dimensional (2-D) wavelet transform in the method and simulation section.

2.2.1 One-Dimensional Wavelet Transform

Let $\Psi_{a,b}(x)$ and $\Phi_{a,b}(x)$, $a \in \mathbb{R}\backslash\{0\}, b \in \mathbb{R}$ be two families of functions defined as translations and re-scales of functions $\Psi(x)$ and $\Phi(x)$, where

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{a}}\Psi\left(\frac{x-b}{a}\right), \quad \Phi_{a,b}(x) = \frac{1}{\sqrt{a}}\Phi\left(\frac{x-b}{a}\right)$$

The function $\Psi(x)$ is called the wavelet function or the mother wavelet function where $\int \Psi(x)dx = 0$ and $\int \Psi(x)^2 dx = 1$. The function $\Phi(x)$ is called the scaling function or the father wavelet function where $\int \Phi(x)dx = 1$.

For discrete wavelet transform, we can select discrete values of a and b such that the transformation is invertible, where $a = 2^{-j}$, $b = k2^{-j}$, j indicates the scale, and k indicates the shift. More details can be found in Vidakovic (1999) and Nason (2008). In this paper, we only discuss the single level discrete wavelet transform (SL-DWT), where $j = 1$, $a = \frac{1}{2}$ and $b = \frac{k}{2}$. To simplify the notation, we denote $\varphi_\omega(x)$ for all scaled and shifted wavelet functions $\Psi_{a,b}(x)$ and $\Phi_{a,b}(x)$, where $\omega = 1, 2, ..., \Omega$, and Ω is the total number of 1-D single level wavelet coefficients. Different wavelet functions may have different Ω. Suppose we have a time series $g(x)$, the 1-D discrete wavelet transform of $g(x)$ can be expressed as

$$W_\omega = \sum_x g(x)\varphi_\omega(x)$$

where W_ω is a 1-D single level wavelet coefficient.

2.2.2 Two-Dimensional Wavelet Transform

Similar to 1-D wavelet transform, we have 2-D mother wavelet function $\Psi(x,y)$ such that

$\int\int \Psi(x,y)dxdy = 0$ and $\int\int \Psi(x,y)^2 dxdy = 1$, and 2-D father wavelet function $\Phi(x,y)$ such that $\int\int \Phi(x,y)dxdy = 1$. Their scaled and shifted wavelet functions are

$$\Psi_{(a,b_1,b_2)}(x,y) = \frac{1}{\sqrt{|a|}}\Psi\left(\frac{x-ab_1}{2}, \frac{y-ab_2}{a}\right)$$

$$\Phi_{(a,b_1,b_2)}(x,y) = \frac{1}{\sqrt{|a|}}\Phi\left(\frac{x-ab_1}{2}, \frac{y-ab_2}{a}\right).$$

For discrete wavelet transform, we can select discrete values of a, b_1 and b_2 such that the transformation is invertible, where $a = 2^{-j}$, $b_1 = k_1 2^{-j}$ and $b_2 = k_2 2^{-j}$, j indicates the scale, k_1 and k_2 indicate the shift. We will only use single level wavelet coefficient where $j = 1$, $a = \frac{1}{2}$, $b_1 = \frac{k_1}{2}$, and $b_2 = \frac{k_2}{2}$. We can simplify the notation of all 2-D single level wavelet functions $\Psi_{(\frac{1}{2},\frac{k_1}{2},\frac{k_2}{2})}(x,y)$ and $\Phi_{(\frac{1}{2},\frac{k_1}{2},\frac{k_2}{2})}(x,y)$ as $\phi_r(x,y)$, $r = 1,2,...,R$, and R is the total number of 2-D single level wavelet coefficients. Let $\{\zeta\} = \{x,y\}$ represent all pairs of 2-D data coordinates. Suppose we have 2-D data $g(\zeta) = g(x,y)$, the 2-D discrete wavelet transform of $g(\zeta)$ can be expressed as

$$W_r = \sum_\zeta g(\zeta)\phi_r(\zeta)$$

where W_r is a 2-D single level wavelet coefficient. The wavelet transform in higher dimensions can be performed similarly.

2.3 Method

2.3.1 Spatio-temporal Model

We now develop our model in more detail. Suppose that there are N subjects, P external stimuli, C many ROIs and V_c many voxels within the c-th ROI. Define the time series at voxel v in ROI c for subject n to be $Y_{ncv}(t)$, $t = 1, ..., T$, where T is the length of time series. We define two functions $\pi_b(\cdot)$ and $\pi_d(\cdot)$ that generate valid covariance matrices. Let $\pi_b(\cdot)$ be a function of the Euclidean distance between voxels within an ROI, and $\pi_d(\cdot)$ is corresponding to the covariance function between ROIs, which does not depend on the Euclidean distance. Using the model described below, we would need to consider three different correlations: the spatial correlation between voxels within an ROI, the temporal correlation within a voxel over time, and the correlation between ROIs. Consider the following spatio-temporal model for the fMRI time series:

$$Y_{ncv}(t) \quad = \quad \sum_{p=1}^{P} [\beta_{ncv}^{p} X^{p}(t)] + \varepsilon_{ncv}(t), \;\; \text{where} \tag{2.1}$$

$$\beta_{ncv}^{p} = \beta_{c}^{p} + b_{ncv}^{p}$$

$$\varepsilon_{ncv}(t) = d_{nc} + e_{ncv}(t)$$

- $X^{p}(t)$ is the expected BOLD response corresponding to the p^{th} stimulus which is formally the convolution between the HRF and the p^{th} impulse function. The HRF is the expected neuronal activation function given a stimuli. An example of these functions has been shown in Figure 2.1.

- β_{c}^{p} is the ROI-specific activation level fixed effect due to stimulus p;

- b^p_{ncv} is a zero-mean voxel-specific random effect that accounts for the spatial covariance between voxels v and v' within ROI c for subject n, where

$$cov(b_{ncv}, b_{n'c'v'}) = \begin{cases} \pi_b(\parallel v - v' \parallel), & \text{when } c = c', n = n', \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

- $\varepsilon_{ncv}(t)$ is the noise that takes into account the voxel-specific temporal correlation. d_{nc} is a zero-mean ROI-specific random effect with a covariance structure $cov(d_c, d_{c'}) = \pi_d(c, c')$ that is used to model the correlation between ROIs. $e_{ncv}(t)$ is the temporal error that is assumed to follow an AR(1) process.

To test whether ROI c is activated when the p^{th} stimulus is presented, we are interested in the hypothesis:

$$H_0 : \beta^p_c - \beta^1_c = 0 \tag{2.3}$$

where β^1_c indicates the baseline condition at ROI c.

2.3.2 Double-Wavelet Transform

The main idea of the double-wavelet transform is to first apply the 2-D/3-D wavelet transform on the spatial image/volume data at each time point, then apply the 1-D wavelet transform on the time series of each wavelet coefficient in the previous step. We also apply the 1-D wavelet transform on the stimulus function. Then we build all models and analyses using the double-wavelet coefficients instead of the original data. For simplicity, we assume our data at each time point are two dimensional and we use the 2-D wavelet transform here.

From equation (2.1), we apply the 2-D discrete wavelet transform on the data at each time point first. Assume that $\phi_r(v)$ are families of one specific 2-D wavelet transform function, then for ROI c, we have

$$U_{ncr}(t) = \sum_{v} Y_{ncv}(t)\phi_r(v) = \sum_{p=1}^{P} \lambda_{ncr}^{p} X^{p}(t) + \varepsilon_{ncv}(t) \text{ where}$$

$r = 1, 2, ..., R$, and R is the total number of 2-D wavelet coefficients at each time point. $\lambda_{ncr}^{p} = \sum_{v} \beta_{ncv}^{p} \phi_r(v)$ is the 2-D wavelet coefficient by applying the 2-D wavelet transform on the spatially dependent parameter β_{ncv}^{p} in ROI c for subject n.

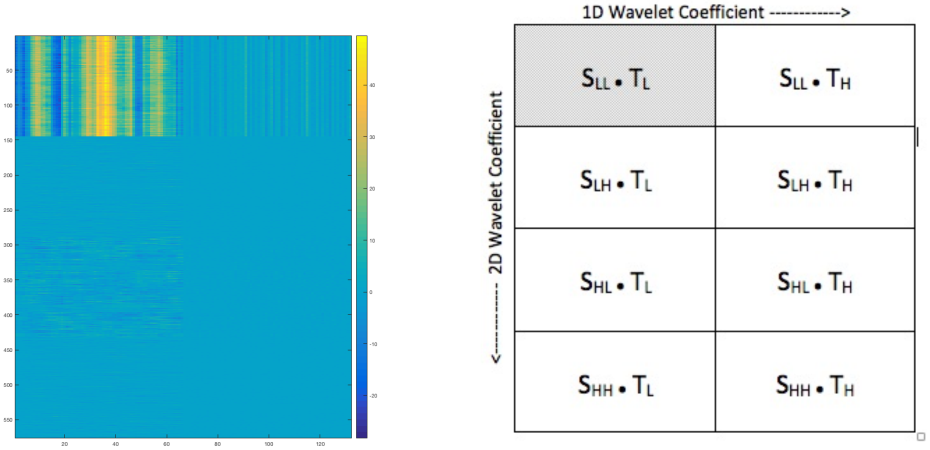
Secondly, we apply the 1-D wavelet transform on the time series of each 2-D wavelet coefficient $U_{ncr}(t)$. Assume that $\varphi_{\omega}(t)$ are families of one specific 1-D wavelet transform function, then we have

$$W_{ncr\omega} = \sum_{t} U_{ncr}(t)\varphi_{\omega}(t) = \sum_{p=1}^{P} \lambda_{ncr}^{p} V_{\omega}^{p} + \delta_{ncr\omega} \text{ where} \tag{2.4}$$

$\omega = 1, 2, ..., \Omega$, and Ω is the total number of 1-D wavelet coefficients for the time series of each 2-D wavelet coefficient. $V_{\omega}^{p} = \sum_{t} X^{p}(t)\varphi_{\omega}(t)$ is the 1-D wavelet coefficients by applying the 1-D wavelet transform on the stimulus function $X^{p}(t)$. $\delta_{ncr\omega} = \sum_{t} \varepsilon_{ncr}(t)\varphi_{\omega}(t)$ is the 1-D wavelet coefficients by applying the 1-D wavelet transform on the time dependent error term $\varepsilon_{ncr}(t)$ at the 2-D wavelet coefficient r in ROI c for subject n. An example of the double-wavelet coefficients $W_{ncr\omega}$ in ROI c for subject n of an activated ROI ($\beta_1 - \beta_2 > 0$) in simulation is illustrated in Figure 2.2(a). After the double-wavelet transform, all $\delta_{ncr\omega}$ are approximately uncorrelated, as long as the two wavelet filters are long enough (Fan, 2003).

2.3.3 Denoising

The wavelet transform naturally decomposes data into different scales. Each scale corresponds to different frequency bands for both spatial and temporal data. The single level discrete wavelet transform (SL-DWT) decomposes data into two frequency bands at each dimension. For example, a 1-D signal would be transformed into wavelet coeffi-



(a) (b)

Figure 2.2: The double-wavelet coefficient structure. (a) is an example using the Daubechies 3 wavelet (spatial wavelet function) and the Symlet 8 wavelet (temporal wavelet function) on an activated 3-D ROI data ($\beta_1 - \beta_2 > 0$) in simulation (2-D in spatial domain and 1-D in temporal domain). (b): S_{LL}, S_{LH}, S_{HL} and S_{HH} represent the wavelet coefficients from the 2-D wavelet transform on the spatial data in low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH) respectively; T_L and T_H represent the wavelet coefficients from the 1-D wavelet transform on the temporal data in low and high frequency band respectively.

cients indicating information from one high frequency band and one low frequency band by SL-DWT. A 2-D image would be decomposed into high and low frequency bands on both vertical and horizontal directions by SL-DWT, which results into four parts, low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH). For subject n at ROI c, $W_{ncr\omega}$ in equation (2.4) would have the same shape as in Figure 3.1(b), where S indicates the wavelet coefficients from the 2-D wavelet transform on the spatial data and T indicates the wavelet coefficients from the 1-D wavelet transform on the temporal data.

In simulation, we found that more than 95% of the wavelet periodograms were contained in the $S_{LL} \cdot T_L$ part, which included $1/8$ of the original data. The data in $S_{LL} \cdot T_L$ were the double-wavelet coefficients from the father wavelet (low-pass filter) in both time domain and spatial domain. The double-wavelet coefficients other than $S_{LL} \cdot T_L$ were the wavelet coefficients passing at least one high-pass filter in time domain or spatial domain. To remove these noise from data, we simply excluded the wavelet coefficients not in the $S_{LL} \cdot T_L$ part when estimating parameters: we only used the double-wavelet coefficients in the $S_{LL} \cdot T_L$ part, where $r = 1, 2, ..., R/4$ and $\omega = 1, 2, ..., \Omega/2$.

2.3.4 Estimation

We first define some notations. We define $N_q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as multi-variate normal distribution with $q \times 1$ mean vector $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$, and denote the $q \times q$ identity matrix by $\boldsymbol{I_q}$

We can rewrite equation (2.4) in matrix notation, we have

$$\boldsymbol{W}_{ncr} \quad = \quad \boldsymbol{V}^T \boldsymbol{\lambda}_{ncr} + \boldsymbol{\delta}_{ncr}, \text{ where} \tag{2.5}$$

- $\boldsymbol{W}_{ncr} = [W_{ncr1}, W_{ncr2}, ..., W_{ncr(\Omega/2)}]^T$ is a $\Omega/2 \times 1$ vector, which are the double-wavelet coefficients in $S_{LL} \cdot T_L$ part by performing double-wavelet transform on data Y_{ncv} for subject n in ROI c. We assume that $\boldsymbol{W}_{ncr} \sim N_{\Omega/2}(\boldsymbol{V}^T \boldsymbol{\lambda}_{ncr}, cov(\boldsymbol{\delta}_{ncr}))$

- $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V^1} \\ \boldsymbol{V^2} \\ .. \\ \boldsymbol{V^P} \end{bmatrix} = \begin{bmatrix} V_1^1 & V_2^1 & V_3^1 & \cdots & V_{\Omega/2}^1 \\ V_1^2 & V_2^2 & V_3^2 & \cdots & V_{\Omega/2}^2 \\ & & \cdots\cdots\cdots & & \\ V_1^P & V_2^P & V_3^P & \cdots & V_{\Omega/2}^P \end{bmatrix}_{P \times \Omega/2}$

  where $\boldsymbol{V}$ is a $P \times \Omega/2$ matrix, in which elements are the 1-D wavelet coefficients by performing 1-D wavelet transform on $X^p(t)$ corresponding the p^{th} stimulus function;

- $\boldsymbol{\lambda}_{ncr} = [\lambda_{ncr}^1, \lambda_{ncr}^2, ..., \lambda_{ncr}^P]^T$ is a $P \times 1$ vector, consisting of the 2-D wavelet coefficients by performing 2-D wavelet transform on the β_{ncv}^p;

- $\boldsymbol{\delta}_{ncr} = [\delta_{ncr1}, \delta_{ncr2}, ..., \delta_{ncr(\Omega/2)}]^T$ is a $\Omega/2 \times 1$ vector, consisting of the 1-D wavelet coefficients by performing 1-D wavelet transform on the error term ϵ_{ncv}, and $\boldsymbol{\delta}_{ncr} \sim N_{\Omega/2}(\underline{0}, \sigma^2 \boldsymbol{I}_{\Omega/2})$, where σ^2 is the variance of the wavelet coefficients $\boldsymbol{\delta}_{ncr}$.

Then we estimate $\hat{\boldsymbol{\lambda}}_{ncr}$ by using the ordinary least squares estimator

$$\hat{\boldsymbol{\lambda}}_{ncr} = (\boldsymbol{V}\boldsymbol{V}^T)^{-1}\boldsymbol{V}\boldsymbol{W}_{ncr}$$

The boxcar stimuli are orthogonal to each other in task-induced fMRI data, which means $\boldsymbol{X}^{p'} \cdot (\boldsymbol{X}^{\boldsymbol{p}})^T = 0$ when $p \neq p'$, then

$$\boldsymbol{V}^{p\prime} \cdot (\boldsymbol{V}^{\boldsymbol{p}})^T = (\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{p\prime})(\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{\boldsymbol{p}})^T = \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{p\prime}(\boldsymbol{X}^{\boldsymbol{p}})^T\boldsymbol{\phi}(\boldsymbol{x})^T$$

where $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(x) \quad \phi_2(x) \quad ... \quad \phi_{\Omega/2}(x)]$ and $\phi_k(x), k = 1, 2, ..., \Omega/2$ are wavelet functions. Then we have $\boldsymbol{V}^{p\prime} \cdot (\boldsymbol{V}^{\boldsymbol{p}})^T = 0$ and $(\boldsymbol{V}\boldsymbol{V}^T)^{-1}$ is a diagonal matrix.

A simple t-test is used on a linear contrast of $\boldsymbol{\lambda}_{ncr}$ based on multi-subject data. Since there is a one to one relationship between β^p_{ncv} and λ^p_{ncr}, the hypothesis in equation (4.2) is equivalent to

$$H_0 : \lambda^p_c - \lambda^1_c = 0$$

where λ^1_c is the mean of the estimator λ^1_{ncr} and λ^p_c is the mean of the estimator λ^p_{ncr} across subjects for ROI c using double-wavelet coefficients in $S_{LL} \cdot T_L$ part as discussed in Section 2.3.3.

2.4 Simulation Study

We explored and validated our approach *via* simulation studies. We generated multi-subject spatially and temporally correlated data. Then, we compared our double-wavelet (DW) approach with the conventional AV-GLM approach in terms of false positive and false negative rates at each ROI.

2.4.1 Data Generation

We have two simulation studies with different number of ROIs. In the first simulation, we generated data with two ROIs. One ROI was assumed to be null ROI ($\beta^2 - \beta^1 = 0$) and the other ROI was assumed to be non-null ROI ($\beta^2 - \beta^1 \neq 0$). We varied the correlation between ROIs and the effect size ($\beta^2 - \beta^1$) from 0 to 1. Three to ten subjects were used in the analysis.

In the second simulation, we generated data with six ROIs. Three ROIs (ROI 1, ROI 2 and ROI 3) were assumed to be null ROIs and three ROIs (ROI 4, ROI 5 and ROI 6) were assumed to be non-null ROIs with the effect size 0.6. These six ROIs have the variance $\sigma^2 = 1$ and the correlation matrix in π_d in equation (2.1) as

$$\begin{pmatrix} 1.0 & 0.6 & 0.0 & 0.5 & 0.0 & 0.2 \\ 0.6 & 1.0 & 0.2 & 0.1 & 0.0 & 0.1 \\ 0.0 & 0.2 & 1.0 & 0.0 & 0.1 & 0.0 \\ 0.5 & 0.1 & 0.0 & 1.0 & 0.2 & 0.6 \\ 0.0 & 0.0 & 0.1 & 0.2 & 1.0 & 0.0 \\ 0.2 & 0.1 & 0.0 & 0.6 & 0.0 & 1.0 \end{pmatrix}$$

In both simulation studies, each ROI contained 100 voxels (10×10). At each voxel, spatially and temporally correlated time series with $T = 128$ were generated using AR(1) structure. The parameter of AR(1) model was 0.6. Two boxcar external stimuli were used to generate each signal.

We considered three different spatial correlations. First we assumed all voxels were independent within an ROI, where $\pi_b(\cdot) = 0$ in equation (2.2). Secondly, we assumed that $\pi_b(\cdot)$ was an exponential covariance function with the decaying parameter 0.5. Thirdly, we assumed that all observations were the same across voxels, meaning that all voxels were extremely correlated with one another. All results were based on 100 repetitions for each simulation scenario.

2.4.2 Estimation and Results

We tried 54 different wavelet functions in the MATLAB (MathWorks, Natick, MA) wavelet toolbox for both 2-D wavelet transform and 1-D wavelet transform. There were a total of 2916 (54×54) combinations of the double-wavelet transform. In simulation, we found that using the Daubechies 3 wavelet (Db3) on the spatial data, and using the Symlets 8 wavelet (Sym8) on the temporal data minimized both the Type I errors and Type II errors.

2.4.2.1 Two ROIs

Figure 2.3 shows the Type I errors and the Type II errors for the DW and AV-GLM approach based on 10 subjects under three different spatial correlations when the effect size $(\beta_1 - \beta_2)$ was 0.6 and there was no correlation between the two ROIs. When all voxels were uncorrelated, the Type II errors using the DW approach (3%) were less than half of the Type II errors using the AV-GLM approach (7%) while their Type I errors were close (6% vs 5%). When the spatial correlations among voxels were based on the exponential covariance function, the Type I and Type II errors using the DW approach (4% and 5%) were smaller than using the AV-GLM approach (7% and 8%). When all voxels were identical, the Type I and Type II errors using the DW approach (8% and 25%) were slightly smaller than using the AV-GLM approach (9% and 26%).

Figure 2.4 shows the Type I errors and the Type II errors for the DW and AV-GLM approach based on 10 subjects when the correlation between the two ROIs varied from 0 to 0.9 and the effect size $(\beta_1 - \beta_2)$ was 0.6. There was no effect of between-ROI correlation on the Type I and Type II errors. When all voxels were uncorrelated, the AV-GLM approach had slightly smaller Type I errors than the DW approach (5% vs. 6%) while the Type II errors using the DW approach (3%) were less than half of the Type II errors using the AV-GLM approach (7%). When the spatial correlations among voxels were based on the

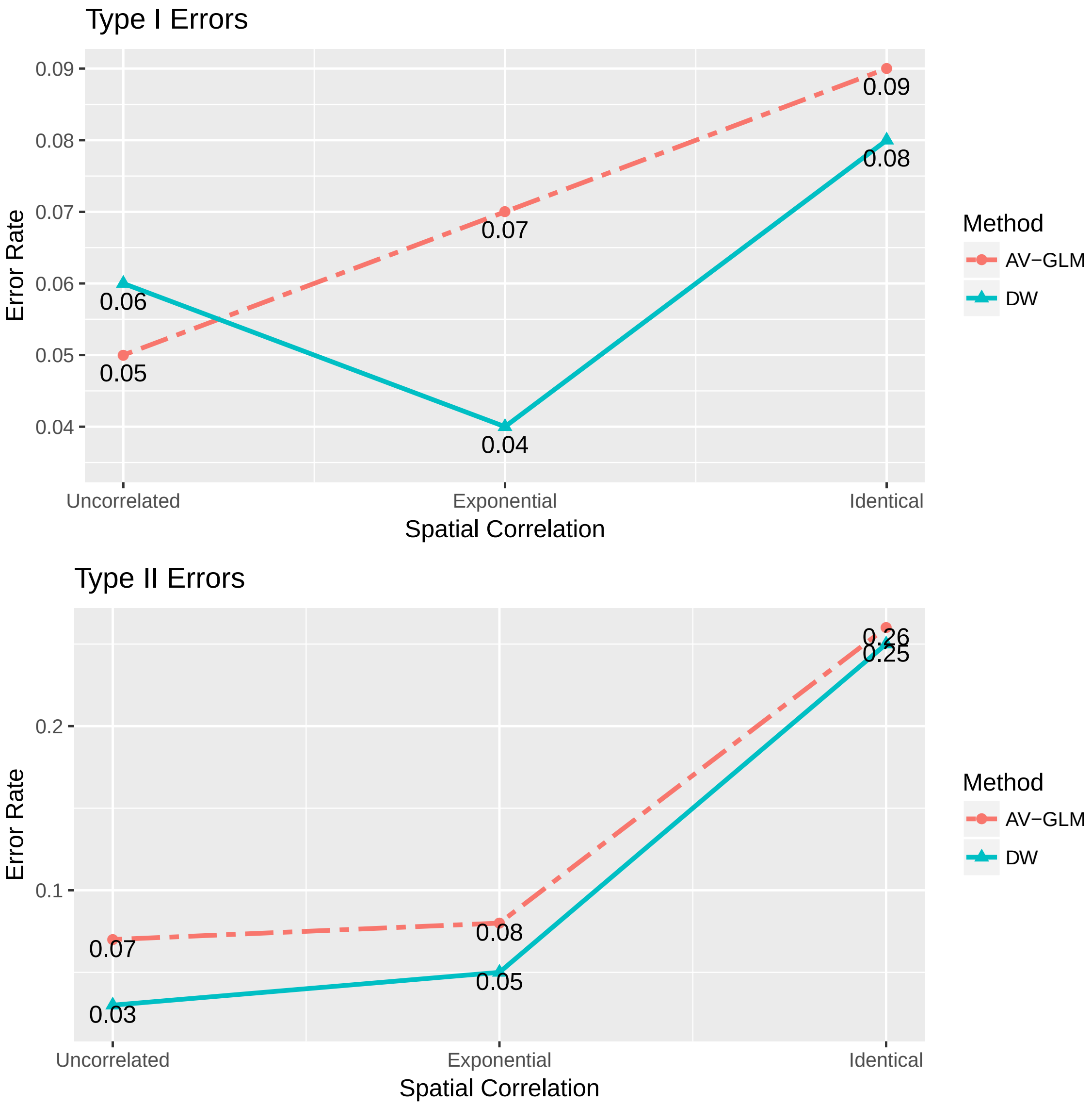


Figure 2.3: Type I and Type II errors for the DW and AV-GLM approach based on 10 subjects with different spatial correlation structures, when the effect size ($\beta_1 - \beta_2$) was 0.6 and the correlation between the two ROIs was 0.

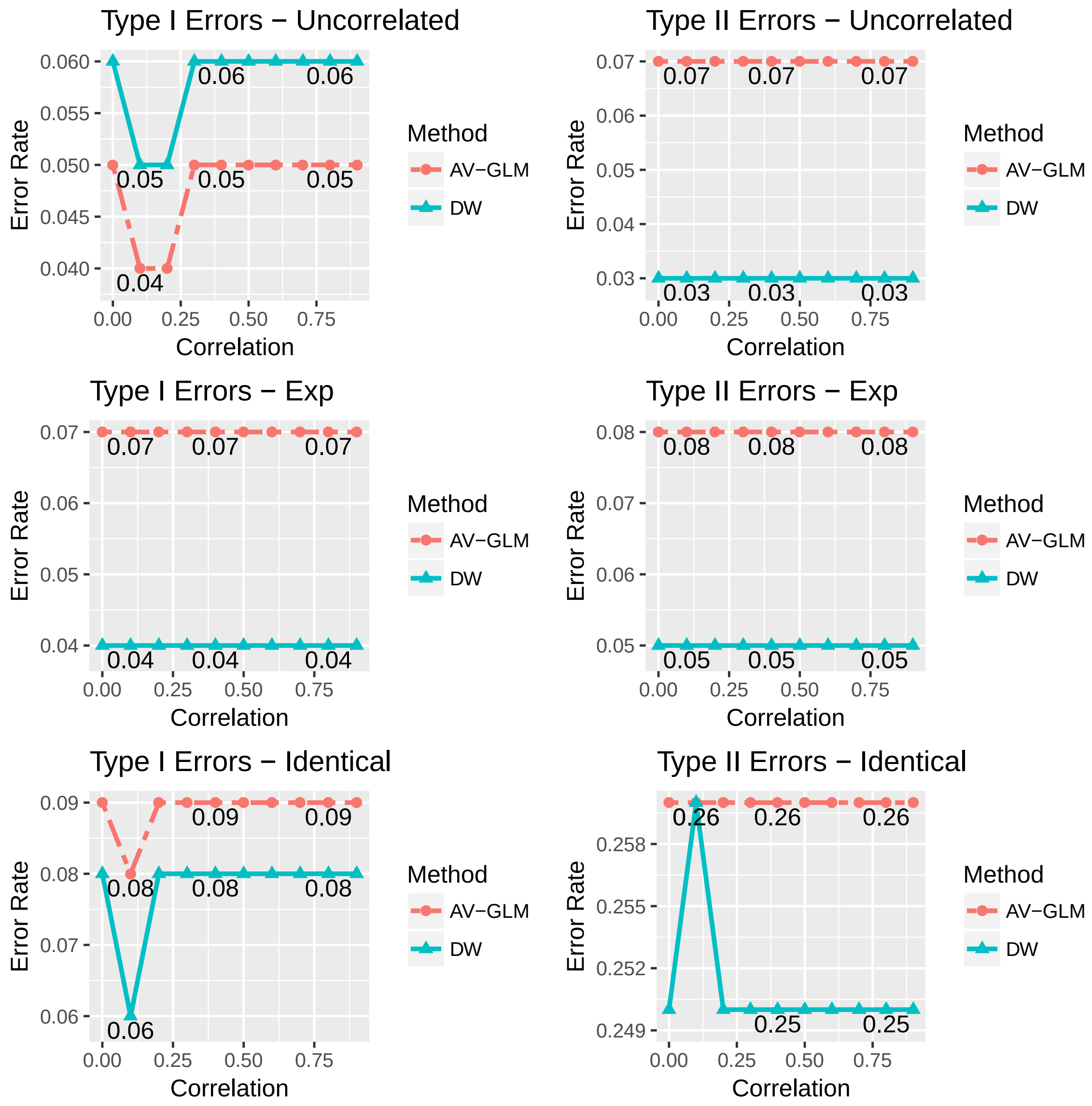


Figure 2.4: Type I and Type II errors for the DW and AV-GLM approach based on 10 subjects with different correlations between ROIs, when the effect size ($\beta_1 - \beta_2$) was 0.6.

exponential covariance function, the Type I and Type II errors using the DW approach (4% and 5%) were both smaller than using the AV-GLM approach (7% and 8%). The DW approach had smaller Type I and Type II errors (8% and 25%) than the AV-GLM approach (9% and 26%) when all voxels were identical.

Figure 2.5 shows the Type I errors and the Type II errors for the DW and AV-GLM approach based on 10 subjects when the effect size ($\beta_1 - \beta_2$) changed from 0.1 to 1 under three different spatial correlations. The correlation between the two ROIs was 0.2. There was no change in the Type I errors with the effect size. Two methods had no difference when all voxels were independent, identical or the spatial correlations among voxels were based on the exponential covariance function in terms of the Type II errors. The AV-GLM approach had smaller Type I errors than the DW approach (5% vs. 6%) when all voxels were uncorrelated. The DW approach had smaller Type I errors than the AV-GLM approach (4% vs. 7%) when the spatial correlations among voxels were based on the exponential covariance function. When all voxels were identical, the DW approach had slightly lower Type I errors than the AV-GLM approach (8% vs. 9%).

Figure 2.6 shows the Type I errors and the Type II errors for the DW approach and the AV-GLM approach based on different number of subjects under three different spatial correlations. The correlation between the two ROIs was 0.3 and the effect size ($\beta_1 - \beta_2$) was 0.9. When all voxels were independent, the AV-GLM approach performed better than the DW approach in terms of Type I errors, while their type II errors were very similar. When the spatial correlations among voxels were based on the exponential covariance function, the DW approach had smaller Type I and Type II errors than the AV-GLM approach. When all voxels were identical, the DW approach performed better than the AV-GLM approach in terms of Type I errors, while their type II errors were very similar.

The simulation studies showed that the DW approach outperformed the AV-GLM approach when voxels were identical and the spatial correlations among voxels were based on the exponential covariance function. The AV-GLM approach only outperformed the DW

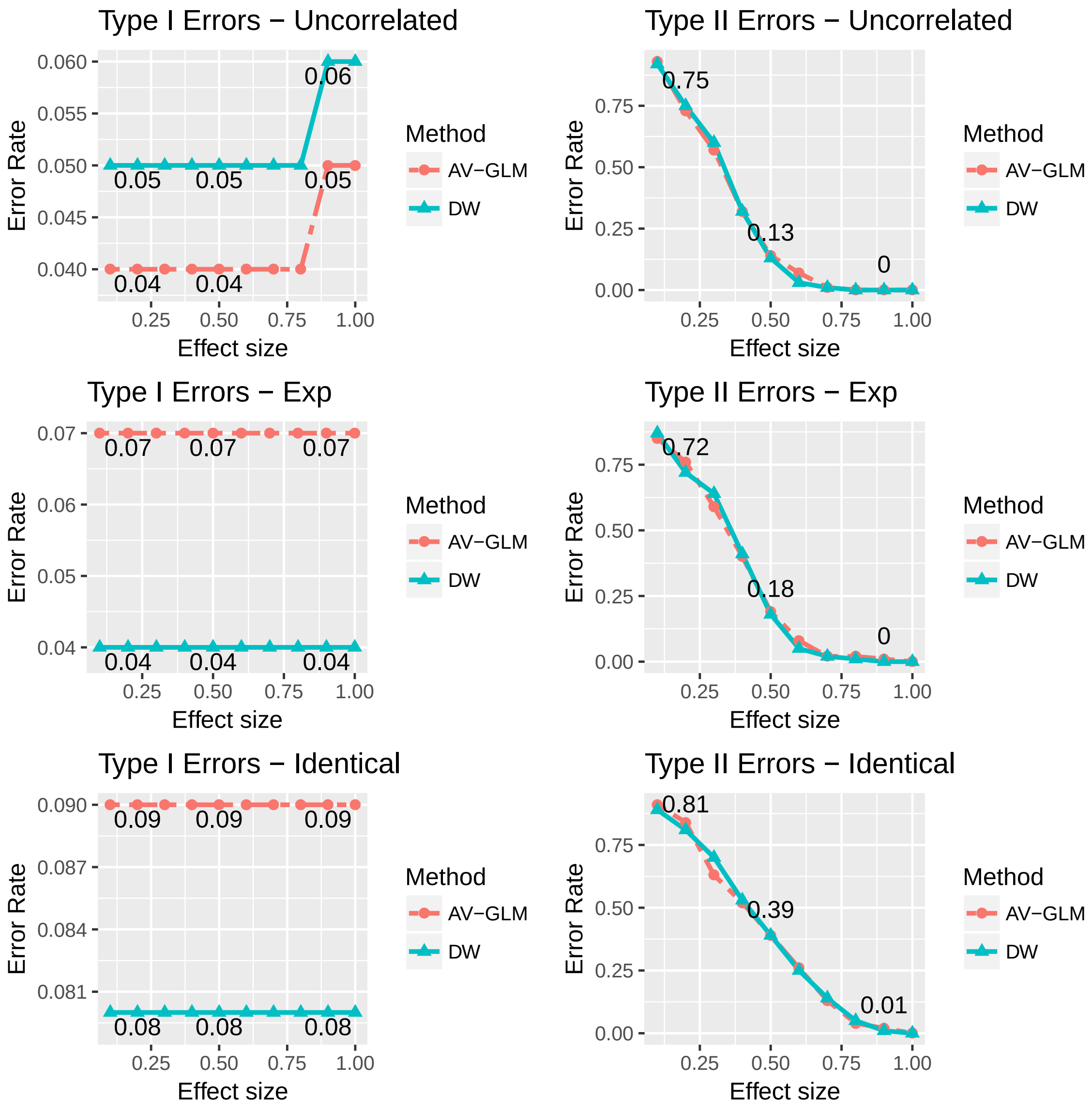


Figure 2.5: Type I and Type II errors for the DW and AV-GLM approach based on 10 subjects with different effect size ($\beta_1 - \beta_2$), when the correlation between the two ROIs was 0.2.

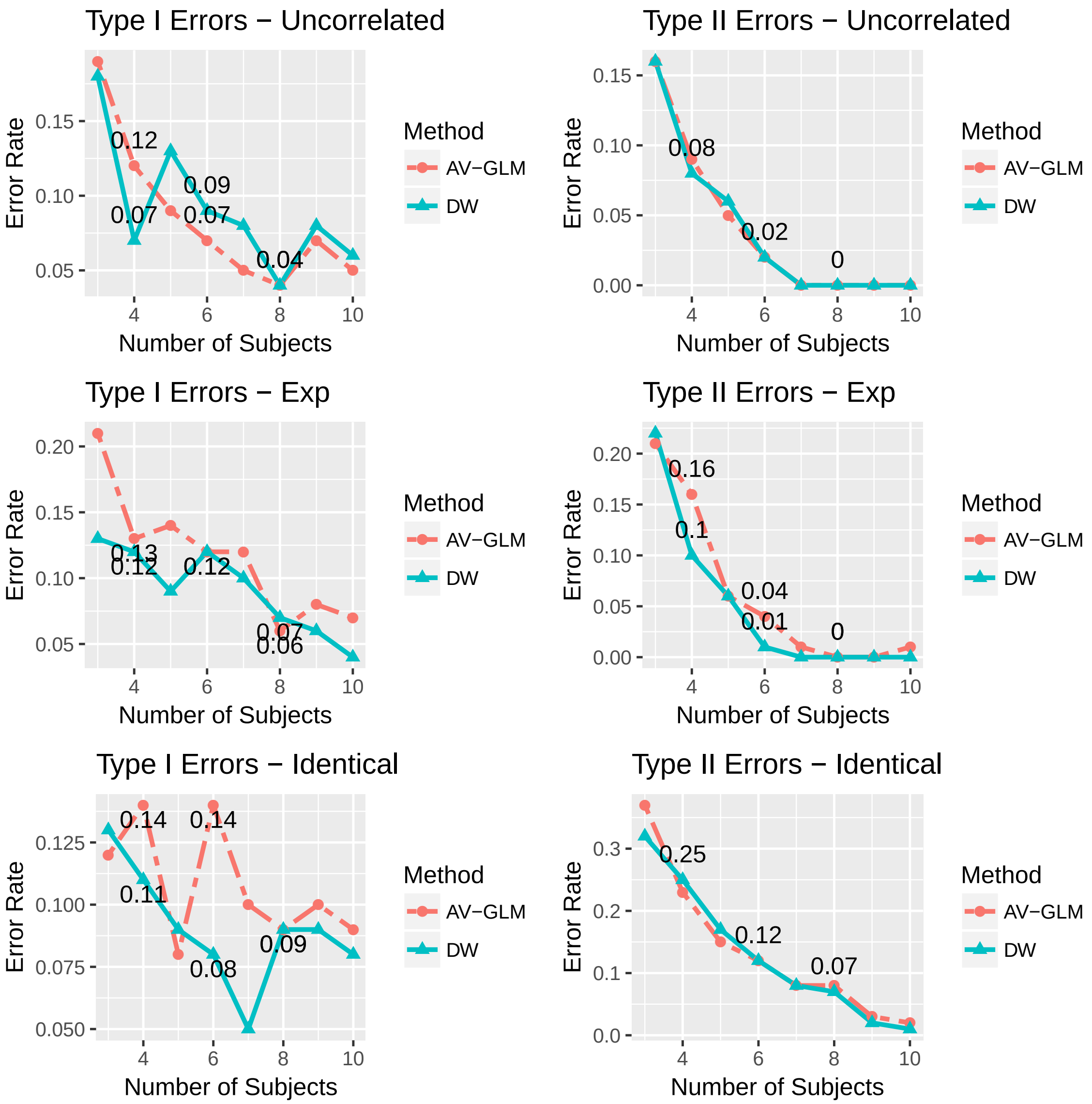


Figure 2.6: Type I and Type II errors for the DW and AV-GLM approach based on different number of subjects. The correlation between the two ROIs was 0.3. The effect size ($\beta_1 - \beta_2$) was 0.9.

Table 2.1: Mean Type I errors for the DW and the AV-GLM approach controlling the FDR at 0.05 when the spatial correlations among voxels were based on exponential covariance function. The effect size ($\beta_1 - \beta_2$) was 0.6

	# of Subject	ROI 1	ROI 2	ROI 3	Mean Type I
DW	4	0.00	0.02	0.05	0.02
AV-GLM	4	0.02	0.03	0.06	0.04
DW	8	0.02	0.01	0.05	0.03
AV-GLM	8	0.05	0.00	0.04	0.03
DW	12	0.01	0.03	0.06	0.03
AV-GLM	12	0.03	0.03	0.06	0.04

approach when all voxels were independent, which is not feasible in practice. The DW approach outperformed the AV-GLM approach in our simulation, partly because it did not require two assumptions. First, the AV-GLM approach assumed that the mean time series of each ROI had AR(1) structure. Second, the AV-GLM approach assumed that all voxels were uncorrelated by not rigorously modeling the underlying spatial correlations in estimating model parameters. These two assumptions used in our simulation studies can be violated in common fMRI data analysis.

2.4.2.2 Six ROIs

For this simulation, there were six correlated ROIs and the effect size ($\beta_1 - \beta_2$) was 0.6. We control the FDR at 0.05 level. For three different types of spatial correlation between voxels, the mean Type I errors for three null ROIs and the mean Type II errors for three non-null ROIs are presented in Figure 2.7. The DW approach performed as good as the AV-GLM approach when all voxels were independent and identical. When the spatial correlations among voxels were based on the exponential covariance function, we found that the DW approach had slightly smaller mean Type I errors for three non-null ROIs than the AV-GLM approach. The Type I errors for 4, 8 and 12 subjects were listed in Table 1.

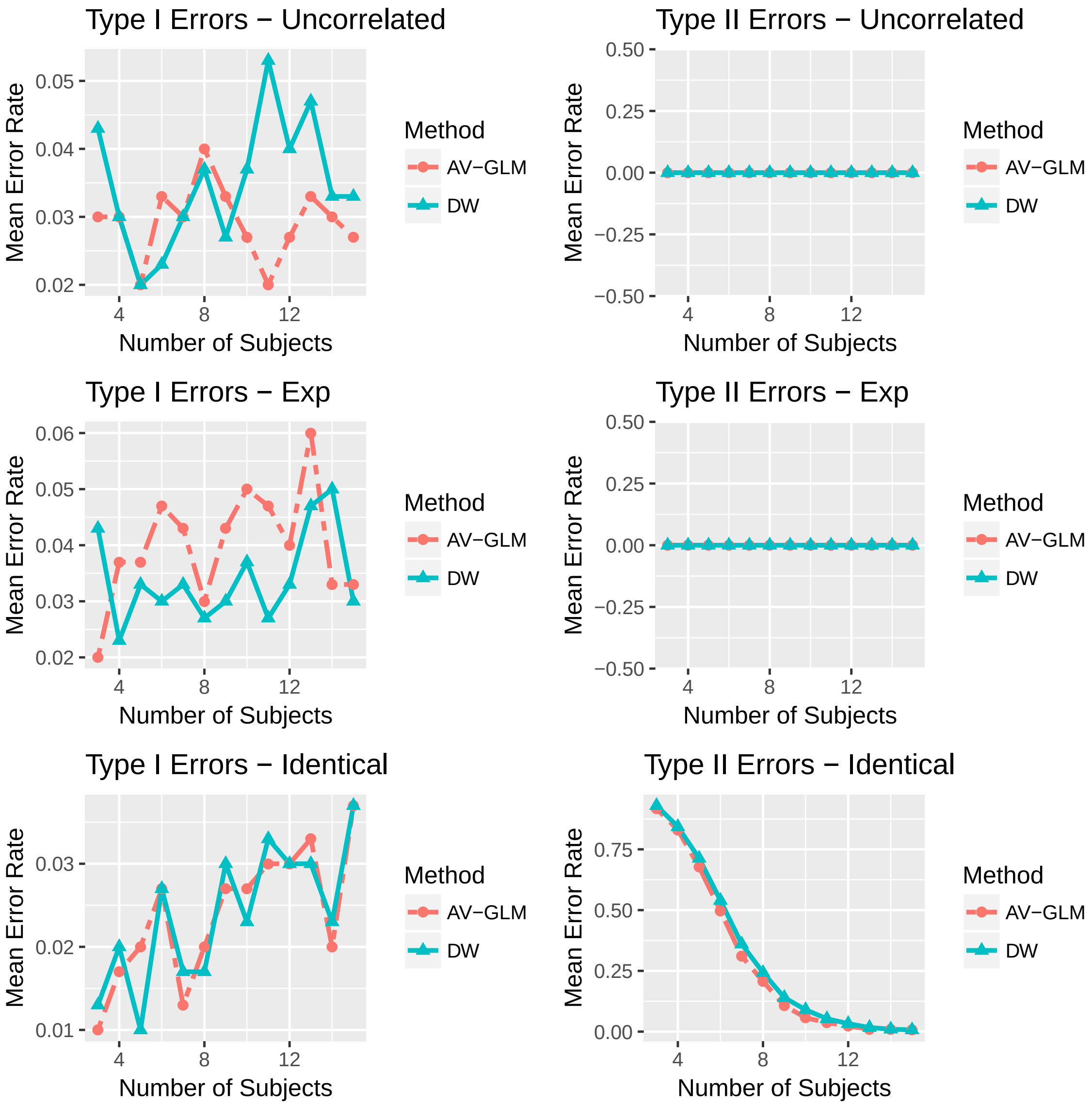


Figure 2.7: Mean Type I errors and mean Type II errors for the DW and AV-GLM approach based on different number of subjects with different spatial correlation structures, when the effect size ($\beta_1 - \beta_2$) was 0.6.

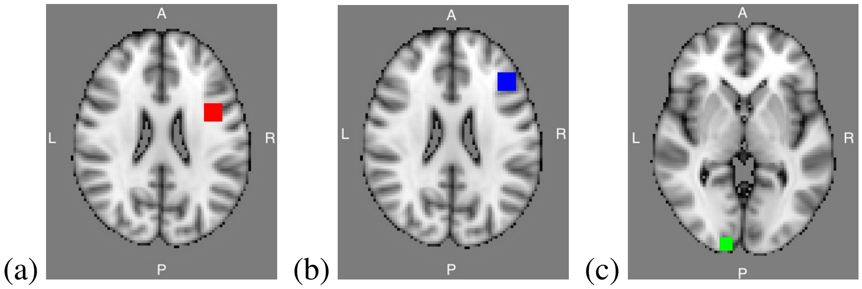


Figure 2.8: Colored boxes on the axial slices of the brain illustrate the location of three ROIs: the red box indicates ROI 1, the blue box indicates ROI 2 and the green box indicates ROI3; their coordinates are also included: (a) ROI 1 (-40,4,3), (b) ROI 2 (-42,28,24), (c) ROI 3 (14,-100,0)

2.5 Data Analysis

We applied our proposed double-wavelet transform approach to a study designed to test cognitive control related activation in the prefrontal cortex (PFC) of the human brain. Here we describe the background, motivation, study design and the data.

In a previous study (Badre & D'Esposito, 2007; Long & Badre, 2009), the anterior premotor cortex (prePMd) was activated in an experimental situation. During the experiment, each participant first selected one of two perceptual dimensions (i.e., shape or texture) of a stimulus and then selected a response. All participants were trained to press one of four buttons while seeing four stimulus shapes and four stimulus textures (e.g., webbed, streaked). Since the response sets associated with shape and texture overlap, the participant needed to first select whether shape or texture was the relevant cue dimension to make a correct response. Participants also needed to link two colors with the texture dimension and two colors with the shape dimension. In the experiment, they would see a shape with a particular texture within a colored box. They would choose one of the four buttons based on the color of the box and the associations they learned for shape or for texture.

For a given block of trials, participants would be cued either one dimension (D1) or two dimensions (D2). On D2 blocks, they were required to select the relevant dimension based on color. On D1 blocks, minimal cognitive control was required because the same dimension was always relevant. Based on prior studies (Badre & D'Esposito, 2007; Badre et al., 2009), prePMd should be activated with the contrast of $D2 > D1$. We defined this as ROI 1. Regions rostral to prePMd (e.g., lateral frontal cortex, Brodmann area 9/46) were suggested to have less activation by this specific control demand by hierarchical theories of rostro-caudal frontal organization. These regions were more involved in more abstract control (Koechlin et al., 2003; Badre, 2008). One region rostral to prePMd in lateral PFC was associated with higher order control (Badre & D'Esposito, 2007). We defined this as ROI 2. We also defined an ROI in the primary visual cortex (ROI 3) whose D2 versus D1 contrast should not be different in low-level perceptual demands. This ROI 3 was not expected to show activation. The three ROIs were illustrated in Figure 2.8, where ROI 1, ROI 2, and ROI 3 were denoted by colored boxes in each axial slice of the brain image. Their x-, y-, and z-coordinates in the brain were also included, where the origin was at the center of the brain.

There were 144 trials (2 sec/trial) for each dimension condition (D1, D2). The inter-trial interval was 0 - 8 sec. Six scanning runs (4 min/run) contained 48 trials each. Each run had 4 blocks (12 trials/block), which followed an ABBA format for dimension type (e.g., D1, D2, D2, D1). The subject's order of dimension conditions were counterbalanced.

Functional images were acquired using a gradient-echo echo-planar sequence (TR = 2 sec; TE = 30 msec; flip angle = 90; 33 axial slices, $3 \times 3 \times 3.5$ mm) on the Siemens 3T TIM Trio MRI system at the Brown University MRI Research Facility. High-resolution T1-weighted (MP-RAGE) anatomical images were collected (TR = 1900 msec; TE = 2.98 sec; flip angle = 9; 160 sagittal slices, $1 \times 1 \times 1$ mm) after each run. Participants would see visual stimuli through a mirror attached to a matrix eight-channel head coil.

Data were preprocessed and analyzed using SPM12 and MATLAB (MathWorks, Natick, MA). To correct for differences in slice acquisition timing, all images of slice in time were re-sampled to match the first slice. Images were then motion corrected and normalized to Montreal Neurological Institute stereotaxic space, and spatially smoothed for the AV-GLM approach (FWHM=4) but not for the double-wavelet approach.

We applied the 3-D wavelet transform using the Db3 wavelet on the volume of the brain at each time point, and then applied the 1-D wavelet transform using the Sym8 wavelet on the time series of each 3-D wavelet coefficient. We had three stimuli: D1, D2, and the instruction period (IP). The three stimuli were convolved with the canonical HRF used in SPM12 and then transformed using the 1-D Sym8 wavelet, which were denoted by $\boldsymbol{V}_{D1}$, $\boldsymbol{V}_{D2}$ and $\boldsymbol{V}_{DIP}$. Then we applied equation (2.5) to the data collected from 15 subjects to see if we could find any activated ROI(s) among the three given ROIs. Each ROI contained 343 $(7 \times 7 \times 7)$ voxels per ROI. The corresponding regression coefficients were λ_c^{D1}, λ_c^{D2} and λ_c^{DIP} in ROI c, where $c \in \{1,2,3\}$. $\lambda_c^{D2} - \lambda_c^{D1}$ was of primary interest to test the hypothesis $H_0 : \lambda_c^{D2} - \lambda_c^{D1} > 0$, which was equivalent to $H_0 : \beta_c^{D2} - \beta_c^{D1} > 0$ in equation (2.1). Given the point estimates of $\lambda_c^{D2} - \lambda_c^{D1}$ at each ROI for each subject, p-values were computed based on a t-test. To manage the multiple comparisons, we controlled the False Discovery Rate (FDR) at 0.05 level.

To see which method was more robust when sample size decreased, we compared the results using a different number of subjects to the result using all 15 subjects for each ROI. A full description of this approach, called data decimation, can be found in Yang et al. (2010). We calculated the rejection rates based on all possible combinations of different subjects. For example, the rejection rate at 6 subjects was based on $C(15,6) = 5005$ tests. The rejection rates are illustrated in Figure 2.9. We defined the rejection rate as the *correct* rejection rate if the test was rejected using all 15 subjects. Otherwise, the rejection rate was called the *incorrect* rejection rate. For ROI 1 and ROI 2, the *correct* rejection rates went up when the number of subjects increased. For ROI 3, the *incorrect* rejection rates went down when the number of subjects increased, as we expected to see no difference between D2 versus D1 contrast. For ROI 1, the DW approach had slightly lower *correct* rejection rates (82.6% and 92.7%) than the AV-GLM approach (85.5% and 94.7%) for 3 and 4 subjects. For ROI 2, the DW approach had slightly lower *correct* rejection rates (94.1% and 99.4%) than the AV-GLM approach (97.4% and 99.9%) for 3 and 4 subjects. For ROI 3, the DW

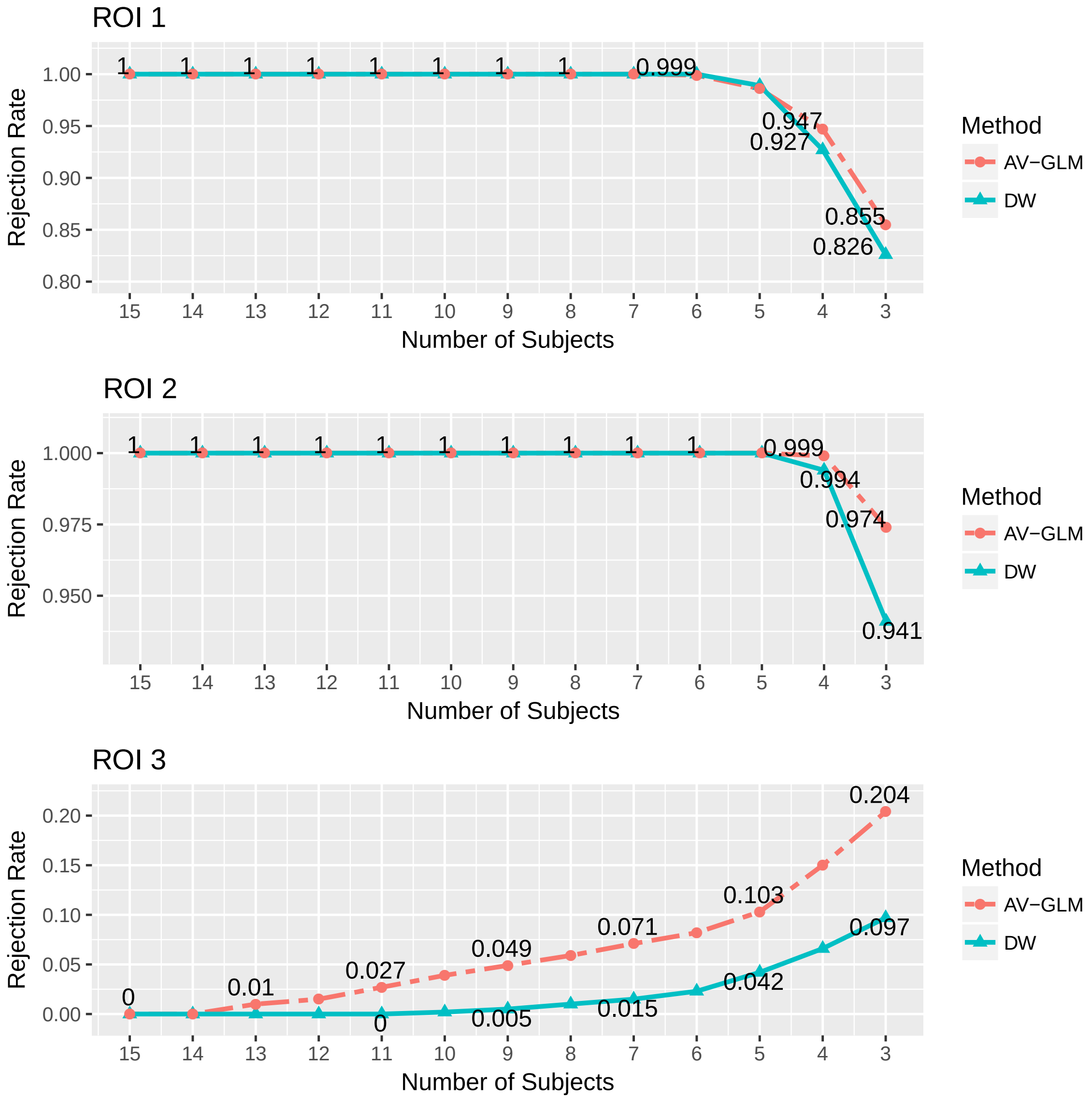


Figure 2.9: Rejection rates for ROI 1, ROI 2 and ROI 3 using different numbers of subjects for the DW and AV-GLM approach

approach had much smaller *incorrect* rejection rates (9.7%, 4.2%, 1.5% and 0.5%) than the AV-GLM approach (20.4%, 10.3%, 7.1% and 4.9%) for 3, 5, 7 and 9 subjects. After 10 subjects, there was no difference between the two methods for all three ROIs.

These results were consistent with our claim supported via simulation studies in Section 4 that the DW approach outperformed the AV-GLM approach in terms of the Type II errors and the Type I errors resulted from the two methods were similar, when the voxel-level spatial correlation was based on the exponential covariance function. We can conclude that ignoring the underlying spatial correlation in fMRI data results in misleading scientific findings for a small sample size by potentially underestimating the variance of estimators of interest as shown in Dubin (1988).

2.6 Conclusion

In this article, we proposed a double-wavelet transform approach, where we transformed the data twice using different wavelet functions and the stimulus functions were also transformed into a wavelet domain. The advantages of our wavelet model over existing models are as follows. First, we took into account the spatial-temporal correlation in fMRI data by transforming the data twice. Second, stationary assumptions on both spatial and temporal data were not required using the double-wavelet transform since the wavelet transform handled it naturally. Third, our approach did not require the estimation and inversion of the covariance matrix because wavelet coefficients were approximately uncorrelated, which significantly reduced the computational burden in the fMRI data analysis. Fourth, the double-wavelet transform converted 4-D data into 2-D data, which simplified the fMRI data structure.

In simulation, we suggested using the Daubechies 3 wavelet on the spatial data and the Symlets 8 wavelet on the temporal data. When there were two ROIs, we found that the DW approach outperformed the AV-GLM approach when all voxels were independent and there existed underlying spatial correlation. The AV-GLM approach only outperformed the DW approach when all voxels were identical, which would be unrealistic in fMRI data analysis. When there were six ROIs, we found that the DW approach outperformed the AV-GLM approach when the spatial correlations among voxels were based on the exponential covariance function.

The data analysis showed that the results using the DW approach were less sensitive to a decrease in sample size, compared to the AV-GLM approach. These results were consistent with our simulation study, meaning that the DW approach made less mistakes than the conventional AV-GLM approach by properly taking into account the spatio-temporal correlation for a small sample size.

One limitation of our approach is the subjective choice of the wavelet functions. Different wavelet functions have various performances. This can be partially solved by using data decimation method to choose wavelet functions, as we did in the data analysis section. The other limitation of our approach is that we may lose some information during the denoising step if there are important signals in either temporal or spatial high frequency band. Our approach can also be extended to the resting state fMRI data to investigate the functional connectivity among ROIs.

Appendix A

Matlab Code for simulation and data analysis

```
% This file is for data analysis part for paper:
% "Double-wavelet Transform for Multi-subject Task-induced Functional
% Magnetic Resonance Imaging Data"

% Package needed: SPM, NIFTI
% All functions needed are in "function" folder

% Part:
% 1. Before analysis:  Extract ROIs from each subject
% 2. AV-GLM approach
% 3. Double wavelet approach
% 4. result combination using R

% Last update: Jul.18, 2017

% add pato of spm, NIFTI, and functions
% addpath '/Users/minchunzhou/Box Sync/matlab/NIfTI_20140122/';
% addpath '/Users/minchunzhou/Box Sync/matlab/spm12/';
% addpath '~/Desktop/JASA_submit/submit_code/functions/'

```

```matlab
%% Parameters

% the size of each 3D image
dim = [79 95 69]   ;

% number of subject needed
subj_num = [106:107,111:112,116:126];

% images per run
num_images = 128;

% number of ROIs
num_ROI = 3;

% number of runs
num_run = 6;


%% 1. Before analysis:  Extract ROIs from each subject
% load five binary ROI files separately to see which coordinates are corresponding

cd('clust_binary');

% load ROI1
temproi1 = load_nii('con2_clust84_bin.nii');
clust84 = temproi1.img;
```

```
rclust1 = findn(clust84);

% load ROI2
temproi2 = load_nii('con2_clust66_bin.nii');
clust66 = temproi2.img;
rclust2 = findn(clust66);

% load ROI3
temproi3 = load_nii('con2_clust16_bin.nii');
clust16 = temproi3.img;
rclust3 = findn(clust16);

rclust = cell(num_ROI,1);

% resize ROIs into box shape
for clust = 1: num_ROI
    temp = eval(['rclust' num2str(clust)]);
    rclust{clust} = reshapeROI(temp, -3,3);
end


% import data
for subj = 1:length(subj_num)

    ROI_timeseries = cell(num_run,num_ROI);
    ROI_timeseries_smooth = cell(num_run,num_ROI);
    subname = subj_num(subj);
    temp0 = sprintf('Importing Subject %d', subj_num(subj));
    disp(temp0)

    if (subname == 105) || (subname == 111) || (subname ==112) || (subname == 113)
        startv = 14;
    else if (subname == 106) || (subname == 107) || (subname ==117)...
                || (subname == 118) || (subname ==127)
            startv = 13;
        else
            startv = 12;
        end
    end

    if subname == 105
        prefix = 1052;
    else if subname == 116
            prefix = 115;
        else
            prefix = subname;
        end
    end

    for session = 1:num_run
        cd(['/data/kangh1/Collaborations/DavidBadre/connectivity/',num2str(subname),'/',...
            num2str(session,'%03d'),'/']);


        temp0 = sprintf('Importing Subject %d %dth run',
           subj_num(subj), session);
        disp(temp0)


         tempimg = zeros(79,95,69,num_images);
         tempimg_smooth = zeros(79,95,69,num_images);

        for j = 1:128
            imgname=['wraf',num2str(prefix),'-',num2str(
               startv,'%04d'),'-',num2str(j,'%05d'),'-',
               num2str(j,'%06d')];
            tempcel= load_nii(imgname);

            tempimg(:,:,:,j) = tempcel.img; ;
            tempimg_smooth(:,:,:,j) = imgaussfilt3(tempcel.
               img, 1.5); % spatial-smooth, sigma = 1.5
        end

       % extract each ROI time series out of whole brain
     for clust = 1:num_ROI

                    tempclust = rclust{clust};
                    seriesfolder1 = [];
```

```
            seriesfolder1_smooth = [];

            for i=1:size(tempclust,1)
                temp1 = tempclust(i,:) ;
                tempx1 = temp1(1);
                tempy1 = temp1(2);
                tempz1 = temp1(3);
                tempseri1 = tempimg(tempx1,tempy1,tempz1,:);
                tempseri1 = tempseri1(:)';
                seriesfolder1 = [seriesfolder1; tempseri1];

                tempseri1_smooth = tempimg_smooth(tempx1,tempy1,tempz1,:);
                tempseri1_smooth = tempseri1_smooth(:)';
                seriesfolder1_smooth = [seriesfolder1_smooth; tempseri1_smooth];

            end
            seriesfolder2 = seriesfolder1;
        ROI_timeseries{session,clust} = seriesfolder2;
        ROI_timeseries_smooth = seriesfolder1_smooth;
```

```
                end


    cd(['/data/kangh1/Collaborations/DavidBadre/connectivity/', num2str(subname),'/' ])

    startv = startv + 1;

    end

    ROI_timeseriesN = cell(num_run, num_ROI);


for k = 1:num_run
    for clust = 1:num_ROI
        series1 = ROI_timeseries{k,clust};
        ROI_timeseriesN{k,clust}  = normseries(series1);

        series1_smooth = ROI_timeseries_smooth{k,clust};
        ROI_timeseriesN_smooth{k,clust}  = normseries(series1_smooth);

    end
end

    cd('/gpfs20/data/kangh1/Collaborations/DavidBadre/connectivity/data_ROI_box')
```

```
            save(sprintf('%d_data_ROI_box', subname)        ,'ROI_timeseriesN')

        cd('/gpfs20/data/kangh1/Collaborations/DavidBadre/connectivity/data_ROI_box_smoothed')
            save(sprintf('%d_data_ROI_box', subname)        ,'ROI_timeseriesN_smooth')


end



%% 2. AV-GLM approach
% conventional analysis for each ROI
num_images = 128; % images per run
t = num_images; % time in TR unit
num_covN = 3;

% Need to create X = S * HRF (convolution)
num_subj = length(subj_num);

TR=2;
hrf = spm_hrf(TR);

stimu = cell(num_subj, num_run);

```

```
% load onset and convolve with HRF
for j = 1:num_subj
    subj = subj_num(j);

    cd(['/Users/minchunzhou/Desktop/Block_behavior/twodhi-', num2str(subj),'/behavioral/']);

    % load the onset for each subject
    dat = load('onsets.mat');
    onsetinfo = dat.neworder1;
    if onsetinfo(1,4) == 0
        onsetinfo(1,4) =1;
    end

    % deal with the on set for each run
    for k =1:num_run

        run = find(onsetinfo(:,1) == k);
        run_info = onsetinfo(run,:);
        run_D1 = find(run_info(:,3) == 1);
        run_D2 = find(run_info(:,3) == 2);
        run_inst = find(run_info(:,3) == 0); % for instrunction part

        onset1 = zeros(1,num_images);
        onset1(run_info(run_D1,4)) = 1; % for D1

        onset2 = zeros(1, num_images);
        onset2(run_info(run_D2,4)) = 1; % for D2

        onset3 = zeros(1, num_images);
        onset3(run_info(run_inst,4)) = 1; % for instruction

         % convolve onset with HRF for each run
        s1 = conv(onset1, hrf); s1 = s1(1:num_images);
        s2 = conv(onset2, hrf); s2 = s2(1:num_images);
        s0 = conv(onset3, hrf); s0 = s0(1:num_images);
        if k >=2
            s0(5) = 0;
        end

        stimu{j,k} = [s1' s2' s0'];
    end
end


beta_diff_glm = zeros(num_subj,num_ROI);

% estimation
for j = 1:num_subj
    subj = subj_num(j);

    % load the smoothed data
    cd(['/Users/minchunzhou/otherfile/Study/Research/
        Dissertation/block_dissertation/block_data_analysis/
        ROIdata/data_ROI_box_smoothed/']);

    load( sprintf('%d_data_ROI_box.mat',subj ))
    row=1;

    for clust = 1:num_ROI

        First_part = zeros(3);
        Second_part = zeros(3,1);

        for run = 1:num_run

            datarun = ROI_timeseriesN{run,clust};
            X = stimu{j,run};

            [First_temp, Second_temp] = conven_run(datarun,
                X, t);

            First_part = First_part+First_temp;
            Second_part = Second_part+Second_temp;

        end


        allinv = inv(First_part);

```

```
        % estimate the beta
        beta_hat_glm = allinv*Second_part;
        beta_diff_glm(j,clust) = beta_hat_glm(2) -
           beta_hat_glm(1);
    end

end

% delete nan value
beta_diff_glmN = beta_diff_glm( mean(isnan(beta_diff_glm),
   2) ==0, : ) ;

count=1;
pval_glm = zeros(num_subj-2,num_ROI);
result = cell(num_ROI*(15-2),6);

% data decimation
k = length(subj_num);

for calsub = 3:k
    allthree = nchoosek(subj_num,calsub);
    sall = size(allthree);

    result = cell(num_ROI*sall(1),9);
    count=1;

    for clust = 1:num_ROI

    for i = 1: sall(1)

                three = find(ismember(subj_num, allthree(i,:))==1);

                C = num2cell(num2str(three,'%1d'));

                beta_diff_glm_R = mean( beta_diff_glmN( three ,clust) );
                SE_beta_diff_glm_R = std( beta_diff_glmN( three,clust))/sqrt(calsub);
                tstat_glm = beta_diff_glm_R/SE_beta_diff_glm_R;
                pval_glm=2*(1-normcdf(abs(tstat_glm),0,1));

                result{count,1} = clust;
                result{count,2} = calsub;
                result{count,3} = beta_diff_glm_R;
                result{count,4} = SE_beta_diff_glm_R;
                result{count,5} = tstat_glm;
                result{count,6} = pval_glm;
                result{count,7} = strjoin(C,{'_'});
                result{count,8} = 'mean';
```

```
                        result{count,9} = 'mean';

                        count=count+1;

        end


    cd(['/Users/minchunzhou/Desktop/ROIdata/data_combination_roi']);
    cell2csv( sprintf('mean_%d_%d.csv',calsub, clust),result)

    end
end


%% 3. Double wavelet approach
% Double wavelet transform

% all candidate wavelet
allwave = {'haar','db1','db2','db3','db4','db5','db6','db7','db8','db9',...
    'db10','sym2','sym3','sym4','sym5','sym6','sym7','sym8','coif1',...
    'coif2','coif3','coif4','coif5','bior1.1','bior1.3','bior1.5',...
    'bior2.2','bior2.4','bior2.6','bior2.8','bior3.1','bior3
.3',...
    'bior3.5','bior3.7','bior3.9','bior4.4','bior5.5','bior6.8',...
    'rbio1.1','rbio1.3','rbio1.5','rbio2.2','rbio2.4','rbio2.6',...
    'rbio2.8','rbio3.1','rbio3.3','rbio3.5','rbio3.7','rbio3.9',...
    'rbio4.4','rbio5.5','rbio6.8','dmey' };


waveTnumber=18; % sym8
wavePnumber=4; % db3

T = num_images; % time in TR unit
dim = [7 7 7];
waveTname = allwave{waveTnumber};
wavePname = allwave{wavePnumber};
N.level = 1;
Nsubj = length(subj_num);
num_subj = Nsubj;
roisize = 1:7;

WT = wavedec3(ones(dim(1), dim(2),dim(3)), N.level ,
    wavePname);
sWT = size(WT.dec{1});
coefband = prod(sWT);
N.wavecoef = prod(sWT);
```

```

usecoefband = 1:coefband;
[TC,TS] = wavedec(zeros(1,T),1,waveTname);
 Ntime = length(TC);
waveband = 1:(Ntime/2) ;

waveT = zeros(N.wavecoef, Ntime);

wavedata =  zeros(num_run, coefband, Ntime/2);

% onset
TR=2;
hrf = spm_hrf(TR);

stimu = cell(num_subj, num_run);

% double wavelet transform the data
for sub =  1:num_subj
    subj = subj_num(sub);

    cd(['/Users/minchunzhou/otherfile/Study/Research/Dissertation/block_dissertation/block_data_analysis/ROIdata/data_ROI_box/']);

    load( sprintf('%d_data_ROI_box.mat',subj ))
    row=1;

    temp0 = sprintf('Importing Subject %d', subj);
    disp(temp0)

    data = zeros(7,7,7,128);

    for roi = 1:num_ROI
        for run =1:num_run

        temp0 = sprintf('Importing Subject %d %dth run %d roi', subj, run,roi);
        disp(temp0)

            data_temp = ROI_timeseriesN{run,roi};

            % reashap dat aback to coordinator format
            count=1;
            for i=roisize
                for j = roisize
                    for k = roisize
                        data(i,j,k,:) = data_temp(count,:);
                        count= count+1;
                    end
                end
            end


            % 3d wavelet transform
            wavedata(run, :,:) = DW3d(data,wavePname, waveTname);

        end
        cd('/Users/minchunzhou/Desktop/ROIdata/data_combination_wave')
        save(sprintf('wavedata_%d_%d_%d_%d', wavePnumber, waveTnumber, subj, roi) ,'wavedata')

    end
end


% use temporal wavelet transform transfer the stimulus
for j = 1:num_subj
    subj = subj_num(j);

    cd(['/Users/minchunzhou/Desktop/Block_behavior/twodhi-', num2str(subj),'/behavioral/']);

    dat = load('onsets.mat');
    onsetinfo = dat.neworder1;
    if onsetinfo(1,4) == 0
        onsetinfo(1,4) =1;
    end

    for k =1:num_run
```

```

        run = find(onsetinfo(:,1) == k);
        run_info = onsetinfo(run,:);
        run_D1 = find(run_info(:,3) == 1);
        run_D2 = find(run_info(:,3) == 2);
        run_inst = find(run_info(:,3) == 0); % for instrunction part

        onset1 = zeros(1,num_images);
        onset1(run_info(run_D1,4)) = 1; % for D1

        onset2 = zeros(1, num_images);
        onset2(run_info(run_D2,4)) = 1; % for D2

        onset3 = zeros(1, num_images);
        onset3(run_info(run_inst,4)) = 1; % for instruction

        s1 = conv(onset1, hrf); s1 = s1(1:num_images);
        s2 = conv(onset2, hrf); s2 = s2(1:num_images);
        s0 = conv(onset3, hrf); s0 = s0(1:num_images);
        %       s00 = ones(1,num_images);
        if k >=2
            s0(5) = 0;
        end

        % wavelet transform the onset
        [stan1,TS1] = wavedec(s1,1,waveTname);
        [stan2,TS2] = wavedec(s2,1,waveTname);
        [stan0,TS1] = wavedec(s0,1,waveTname);



         stan = [stan1(waveband) stan2(waveband) stan0(waveband)];

        stimu{j,k} = [stan1(waveband)' stan2(waveband)' stan0(waveband)'];
    end
end

% estimation

con = [-1 1 0];
wavecon = zeros(num_run);
wavecon_sub = zeros(num_subj,num_ROI);

for j = 1:num_subj
    subj = subj_num(j);

    for clust = 1:num_ROI

    cd(['/Users/minchunzhou/Desktop/ROIdata/data_combination_wave']);
        load(sprintf('wavedata_%d_%d_%d_%d',wavePnumber,waveTnumber, subj, clust))
        First_inv_temp = zeros(3);
        Second_part = zeros(3,1);

        for run = 1:num_run

            waveX = stimu{j,run};

            wavedata_use = reshape(wavedata(run, :,:),
               coefband, Ntime/2);
            wavedata_use = wavedata_use(usecoefband,waveband
               );

            wavedata_use1 = mean(wavedata_use,1 );%/coefband
                /8*  prod(dim) ;

           First_inv_temp = First_inv_temp+  waveX'*waveX;

           Second_part= waveX'*wavedata_use1'+ Second_part;

        end

        allinv = inv(First_inv_temp);
        beta_hat_glm = allinv*Second_part;

        wavecon_sub(j, clust) = beta_hat_glm(2) -
           beta_hat_glm(1);

```

```
    end

end

beta_diff_glmN = wavecon_sub( mean(isnan(wavecon_sub), 2)==0, : ) ;

k = length(subj_num);

for calsub = 3:k

    allthree = nchoosek(subj_num,calsub);
    sall = size(allthree);

        result = cell(num_ROI*sall(1),9);
        count=1;

    for clust = 1:num_ROI

        for i = 1: sall(1)

                                three = find(ismember(subj_num,allthree(i,:))==1);

                                C = num2cell(num2str(three,'%1d'));


```

```
                        beta_diff_glm_R = mean(beta_diff_glmN( three ,clust) );
                        SE_beta_diff_glm_R = std(beta_diff_glmN( three ,clust))/sqrt(calsub);
                        tstat_glm = beta_diff_glm_R/SE_beta_diff_glm_R;
                        pval_glm=2*(1-normcdf(abs(tstat_glm),0,1));

                        result{count,1} = clust;
                        result{count,2} = calsub;
                        result{count,3} = beta_diff_glm_R;
                        result{count,4} = SE_beta_diff_glm_R;
                        result{count,5} = tstat_glm;
                        result{count,6} = pval_glm;
                        result{count,7} = strjoin(C,{'_'});
                        result{count,8} = wavePname;
                        result{count,9} = waveTname;

                        count=count+1;

         end


        cd(['/Users/minchunzhou/Desktop/ROIdata/
            data_combination_roi']);
        cell2csv( sprintf('dw_%d_%d_%d_%d.csv',wavePnumber,
            waveTnumber,calsub, clust),result)

    end
end
```

Chapter 3

DOUBLE-WAVELET TRANSFORM FOR MULTI-SUBJECT RESTING STATE FUNCTIONAL MAGNETIC RESONANCE IMAGING DATA

The goal of this article is to model multi-subject resting-state fMRI (functional magnetic resonance imaging) response among predefined regions of interest (ROIs) of the human brain. Conventional approaches to resting-state fMRI analysis does not rigorously model the underlying spatial correlation due to the complexity of estimating and inverting the high dimensional spatio-temporal covariance matrix. Other spatio-temporal model approaches estimate the covariance matrix with the assumption of stationary time series. To address these limitations, we propose a double-wavelet approach for modeling the spatio-temporal brain process. Working with wavelet coefficients simplifies temporal and spatial covariance structure, because under regularity conditions, wavelet coefficients are approximately uncorrelated. Different wavelet functions were used to capture different correlation structures in the spatio-temporal model. The advantages of the wavelet approach are that it is scalable and that it captures non-stationarity in brain signals. Simulation studies showed that our method can reduce false positive and false negative rate by taking into account spatial and temporal correlation simultaneously. We also applied our method to resting-state fMRI data to study the difference between heath subjects and major depressive disorder (MDD) patients.

3.1 Introduction

One of the most challenging problem in contemporary science is to understand the human brain and its activity pattern. To acquire good quality data of the human brain function, functional magnetic resonance imaging (fMRI) becomes a powerful tool in the last twenty years. Compared to other methods like PET (Positron Emission Tomography) and MEG (magnetoencephalography), fMRI has reasonable temporal resolution ($1 \sim 2$ seconds) and excellent spatial resolution (in mm^3). Typical fMRI data are collected by scanning the brain every few seconds and indirectly measure brain activity by detecting changes in oxygen level associated with blood flow. There are about 100,000 three-dimensional volume elements called voxels in each fMRI scan.

Resting state fMRI (rs-fMRI) is commonly used to estimate the functional connectivity (FC), which is defined as the temporal dependency between spatially distinct brain regions (Aertsen et al., 1989; Friston et al., 1993). Biswal et al. (1995) first detected strong temporal correlations between the left and right sensorimotor cortices. Several regions were reported to have strong correlations similar to functional networks like visual areas, auditory cortices, and other higher order cognitive networks (Biswal et al., 1997; Lowe et al., 1998; Cordes et al., 2000, 2001; Kiviniemi et al., 2004; Luca et al., 2005; den Heuvel et al., 2008). It is now widely acknowledged that there exists functional connectivity between brain regions, measured by using spontaneous fMRI time-series, while subjects do not think of something in particular during the scan.

Multiple preprocessing steps are usually taken before the rs-fMRI data analysis, such as motion correction, temporal filtering and spatial smoothing. The most common approach to estimate the functional connectivity between a pair of regions of interest (ROIs) is to simply calculate the correlation between the mean time series of two ROIs (Biswal et al., 1995; Cordes et al., 2001; Raichle et al., 2001; Fox et al., 2006; Vincent et al., 2008). We refer to this approach as “AVG-FC”. This approach does not rigorously take into account the underlying spatial correlation within an ROI. Dubin (1988) argued that ignoring spatial correlation in data led to biased standard errors.

Several methods have been developed to take into account the spatial correlation in fMRI data. Worsley et al. (1996) proposed spatial smoothing using a Gaussian kernel, which is now commonly used as a preprocessing step in fMRI study. The goal of spatial smoothing is to increase the signal to noise ratio in fMRI data, however it induces more spatial correlation, which may result in invalid statistical inferences. Spatio-temporal modeling is another approach to not only model the spatial correlation but also consider the temporal correlation in fMRI data. Since Fourier coefficients are approximately uncorrelated across frequencies, Ombao et al. (2008) developed a spatio-spectral model to understand the underlying spatio-temporal processes. Based on that, Kang et al. (2012) proposed a spatio-spectral mixed-effects model to simultaneously estimate spatial correlation within a region and correlation across regions.

Similar to Fourier transform, wavelet transform is a linear transformation. The main difference between two methods are the information trade-off between time and frequency domain. Fourier transform coefficients only contain information about the signal in frequency domain without any information in time domain. The wavelet transform down-sample the signal into different scale and location, which captures the low frequencies in longer time period and high frequencies in short time period. Wavelet coefficients are also approximately uncorrelated across and within scales (Fan, 2003). Brammer (1998) and Ruttimann et al. (1998) first applied wavelet transform on fMRI data. Long et al. (2004) performed wavelet transform on spatial data and calculated temporal noise parameters using iterative methods. Ville et al. (2004) proposed to simultaneously estimated two threshold using the original signal and the wavelet coefficients to denoise the signal. Aston et al. (2005) estimated the spatio-temporal model coefficients using the wavelet coefficients only from spatial data.

The wavelet transform has also been recently adapted to resting state fMRI study. Eryilmaz et al. (2011) separated the signal into four frequency bands and calculated the correlation between regions to investigate the effects of positive and negative emotions on functional connectivity. Patel et al. (2014) add one extra step in the preprocessing step using wavelet transform to remove a range of different motion artifacts and motion-related biases. Meddaa et al. (2011) down-sampled signal into five levels and used a data-driven approach to cluster voxels into functional regions. These methods utilized the multiresolution property of wavelet transform in the time domain. However, they did not consider the spatial correlation in the fMRI data.

In this article, we proposed a single level double wavelet framework that transformed the resting state fMRI data twice using different wavelet functions, and estimated the functional connectivity between specific ROI pairs using weighted Pearson correlation calculated by wavelet coefficients. We transform the fMRI data at each time point using one wavelet function (spatial wavelet function). This is the first wavelet transform and the coefficients are called “spatial wavelet coefficient”. Then we transform the time series of spatial wavelet coefficients using another wavelet function (temporal wavelet function). This is the second wavelet transform and the coefficients are called “double wavelet coefficients”. The single level double wavelet transform separates fMRI data into multiple locations. We estimate the functional connectivity by calculating the mean correlation between wavelet coefficients in different location and weighted by wavelet periodgram. We validate our approach *via* simulation studies with different spatial correlation structures.

3.2 The Discrete Wavelet Transform

The wavelet transform decomposes different dimensional data into multiple scales and locations. Wavelet coefficients are calculated by the inner product of the wavelet functions and the data. The wavelet functions are scaled and shifted from the zero integral “mother” wavelet function and the unit integral “father” wavelet function. We only introduce the one-dimensional (1-D) and two-dimensional (2-D) discrete wavelet transform in this paper. Higher dimensional wavelet transform can be performed similarly.

3.2.1 One-Dimensional Discrete Wavelet Transform

Let $\Psi(x)$ and $\Phi(x)$ be the mother and father wavelet function, where $\int \Psi(x)dx = 0$ and $\int \Phi(x)dx = 1$. We call $\Psi_{j,k}(x)$ and $\Phi_{j,k}(x)$ ($j \in \mathbb{R}\backslash\{0\}, k \in \mathbb{R}$) to be series of wavelet functions that are scaled and shifted from the mother and father wavelet function, respectively.

More details can be found in Vidakovic (1999) and Nason (2008):

$$\Psi_{j,k}(x) = \frac{1}{\sqrt{2^j}}\Psi\left(\frac{x-2^jk}{2^j}\right), \quad \Phi_{j,k}(x) = \frac{1}{\sqrt{2^j}}\Phi\left(\frac{x-2^jk}{2^j}\right).$$

In this paper, we only perform one scale wavelet transform, which we call the single level discrete wavelet transform (SL-DWT), where $j = 1$. To simplify the notation, we combine all scaled and shifted wavelet functions $\Psi_{j,k}(x)$ and $\Phi_{j,k}(x)$ as $\varphi_\omega(x)$, where $\omega = 1, 2, ..., \Omega$, and Ω is the number of the 1-D single level wavelet coefficients. For 1-D time series $g(x)$, the 1-D discrete wavelet transform of $g(x)$ can be expressed as

$$W_\omega = \sum_x g(x)\varphi_\omega(x),$$

where W_ω is one 1-D single level wavelet coefficient. The square of wavelet coefficient is called wavelet periodogram,

$$I_\omega = (W_\omega)^2 = \left(\sum_x g(x)\varphi_\omega(x)\right)^2.$$

The wavelet periodogram I_ω represents the energy content at certain scale and location. We can also define the energy of the wavelet coefficient at scale j by $S_j = \sum_\omega I_\omega$, where S_j is called scalegram at scale j. For further details on the scalegram, see Scargle et al. (1993); Chiann & Morettin (1998).

3.2.2 Two-Dimensional Wavelet Transform

Similar to 1-D wavelet transform, we have 2-D wavelet functions $\Psi_{(j,k_1,k_2)}(x,y)$ and $\Phi_{(j,k_1,k_2)}(x,y)$ ($j \in \mathbb{R}\backslash\{0\}, k_1, k_2 \in \mathbb{R}$) which are scaled and shifted from the 2-D mother wavelet function $\Psi(x,y)$ and father wavelet function $\Phi(x,y)$, where $\int\int \Psi(x,y)dxdy = 0$

and $\int\int \Phi(x,y)dxdy = 1$.

$$\Psi_{(j,k_1,k_2)}(x,y) = \frac{1}{\sqrt{2^j}}\Psi\left(\frac{x-2^jk_1}{2^j}, \frac{y-2^jk_2}{2^j}\right)$$

$$\Phi_{(j,k_1,k_2)}(x,y) = \frac{1}{\sqrt{2^j}}\Phi\left(\frac{x-2^jk_1}{2^j}, \frac{y-2^jk_2}{2^j}\right)$$

For the single level wavelet transform where $j = 1$, all 2-D single level wavelet functions $\Psi_{(j,k_1,k_2)}(x,y)$ and $\Phi_{(j,k_1,k_2)}(x,y)$ can be expressed as $\phi_r(x,y)$, $r = 1,2,...,R$, and R is the number of the 2-D single level wavelet coefficients. Let $\{\zeta\} = \{x,y\}$ represent all pairs of 2-D data coordinates. If we have 2-D data $g(\zeta) = g(x,y)$, the 2-D discrete wavelet transform of $g(\zeta)$ can be expressed as

$$W_r = \sum_{\zeta} g(\zeta)\phi_r(\zeta) \tag{3.1}$$

where W_r is a 2-D single level wavelet coefficient. I_r is the 2-D wavelet periodogram, which is calculated by

$$I_r = (W_r)^2 = \left(\sum_{\zeta} g(\zeta)\phi_r(\zeta)\right)^2 \tag{3.2}$$

3.3 Method

3.3.1 Spatio-temporal Model

Let C be the number of ROIs and V_c be the number of voxels within the c-th ROI. Define the time series at voxel v in ROI c to be $Y_{cv}(t)$, $t = 1,...,T$, where T is the length of time series. We define $\pi_b(\cdot)$ to be a function of the Euclidean distance between voxels within an ROI. Consider the following spatio-temporal model for the fMRI time series:

$$Y_{cv}(t) = d_c(t) + b_{cv}(t) + e_{cv}(t), \tag{3.3}$$

- $b_{cv}(t)$ is a zero-mean voxel-specific random effect that accounts for the spatial covariance between voxels v and v' within ROI c, where

$$cov(b_{cv}(t), b_{cv'}(t)) = \begin{cases} \pi_b(\| v - v' \|), & \text{when } c = c', \\ 0 & \text{otherwise.} \end{cases}$$

- $d_c(t)$ is a zero-mean ROI-specific random effect with a covariance structure $cov(d_c(t), d_{c'}(t))$ that takes into account the correlation across ROIs.

- $e_{cv}(t)$ is the noise that models the temporal correlation for voxel v in ROI c.

3.3.2 The Double Wavelet Transform

We now introduce the double wavelet transform. Suppose we have a 3-D data (2-D spatial data and 1-D temporal data) as in model (3.3). First, we perform 2-D wavelet transform on the 2-D spatial data at each time point. We have

$$U_{cr}(t) = \sum_v Y_{cv}(t)\phi_r(v),$$

where $\phi_r(v)$ is a 2-D wavelet function, $r = 1, 2, ..., R$, and R is the total number of 2-D wavelet coefficients at each time point. Secondly, we apply the 1-D wavelet transform on the time series of each 2-D wavelet coefficients $U_{cr}(t)$. Assume that $\varphi_\omega(t)$ is an 1-D wavelet function, then we have

$$W_{cr\omega} = \sum_t U_{cr}(t)\varphi_\omega(t) \tag{3.4}$$

$\omega = 1, 2, ..., \Omega$, where Ω is the length of 1-D wavelet coefficients.

3.3.3 Coefficients Structure

The wavelet transform separates the data into different scales and locations. In each scale, different locations correspond to different frequency bands. For the single level discrete double wavelet transform (SL-DWT), the data are decomposed into two frequencies in each dimension. For the 3-D data (2-D spatial data and 1-D temporal data), the 2-D spatial data are separated into high and low frequency bands on both vertical and horizontal directions, which includes four parts, low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high(horizontal-vertical) frequency band (HH). The 1-D time series are decomposed into one high frequency band and one low frequency band. Figure 3.1(b) illustrates the structure of double wavelet coefficients $W_{cr\omega}$ in equation (3.4), where S indicates the wavelet coefficients from the 2-D wavelet transform on the spatial data and T indicates the wavelet coefficients from the 1-D wavelet transform on the temporal data.

We define the double wavelet periodogram $I_{cr\omega}$ as the energy of the double wavelet coefficient $W_{cr\omega}$,

$$I_{cr\omega} = (W_{cr\omega})^2$$

We use $\boldsymbol{I}_{cr}$ to represent all wavelet periodogram in each spatial-temporal frequency band, where $p = 1, 2, ..., P$ and P is the total number of spatial-temporal frequency bands that the single level double wavelet transform created. In Figure 3.1(b), $P = 8$ for double wavelet transform that includes 2-D spatial wavelet transform and 1-D temporal transform.

3.3.4 Weighted Correlation

To calculate the correlation between ROI c and c', we first calculate the Pearson correlation ρ_p between two vectors of mean temporal wavelet coefficients in each spatial-temporal frequency band p. We also calculate the variance of wavelet periodogram as the weight

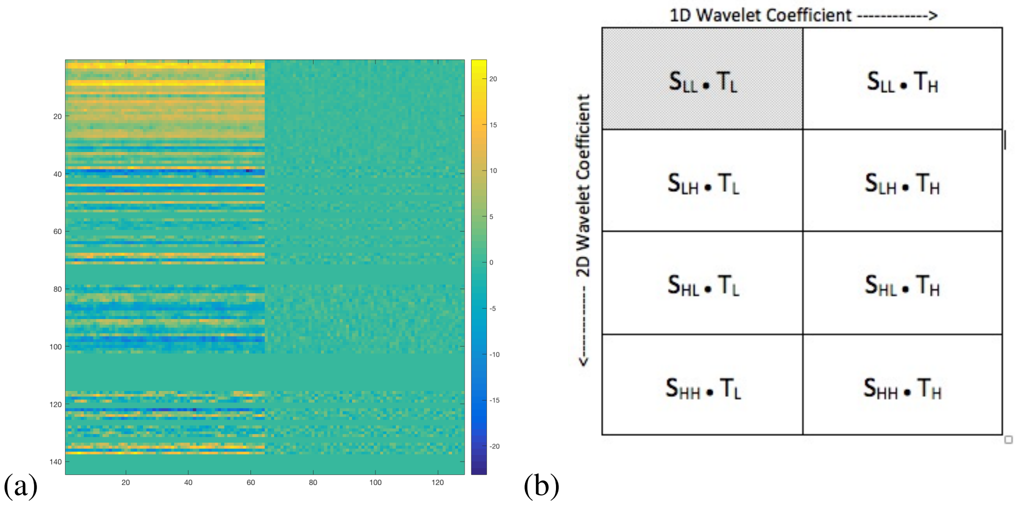


Figure 3.1: Coefficient structure using the double-wavelet transform. (a) an example using Reverse Biorthogonal 3.1 wavelet as spatial wavelet and Haar wavelet as temporal wavelet on a simulated 3-D ROI data (2-D in spatial domain and 1-D in temporal domain). (b) S_{LL}, S_{LH}, S_{HL} and S_{HH} represent the wavelet coefficients from the 2-D wavelet transform on the spatial data in low-low (horizontal-vertical) frequency band (LL), low-high(horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH) respectively; T_L and T_H represent the wavelet coefficients from the 1-D wavelet transform on the temporal data in low and high frequency band respectively.

ϖ_p. Then we have the weighted correlation between ROI c and c' using the double wavelet transform.

To simplify the notation, we use $W^p_{cr\omega}$ to denote the double wavelet coefficient in spatial-temporal frequency band p, then $r = 1,2,...,R/2^j$ and $\omega = 1,2,...,\Omega/2$, where j is the dimension of spatial wavelet. In the example of 3-D data (2-D spatial data and 1-D temporal data), the dimensional of spatial data is two, which means $j = 2$. We can rewrite $W^p_{cr\omega}$ in vector notation,

$$\boldsymbol{W^p_{cr}} = [W^p_{cr1} \quad W^p_{cr2} \quad \dots \quad W^p_{cr(\Omega/2)}].$$

The correlations and weights are calculated by,

$$\rho_p = corr\left(\frac{1}{R/2^j}\sum_{r=1}^{R/2^j}\boldsymbol{W^p_{cr}}, \frac{1}{R/2^j}\sum_{r=1}^{R/2^j}\boldsymbol{W^p_{c'r}}\right)$$

$$\varpi_p \;=\; \frac{var(\boldsymbol{I}_{cp}) + var(\boldsymbol{I}_{c'p})}{\sum_{p=1}^{P}[var(\boldsymbol{I}_{cp}) + var(\boldsymbol{I}_{c'p})]}. \tag{3.5}$$

Finally, we have the weighted correlation $\rho = \sum_{p=1}^{P} \rho_p \varpi_p$. For multiple subjects, we can perform Fisher z-transform on ρ_n and test if the correlation is equal to zero or not, where n indicates the number of subjects.

3.4 Simulation Study

We explored and validated our approach through simulation studies. We generated single subject and multi-subject spatially and temporally correlated data. Then, we compared our double wavelet (DW) approach with the conventional mean voxel approach at each ROI in terms of mean square error (MSE) for single subject analysis and false positive and false negative rates for multi-subject analysis.

3.4.1 Data Generation

We have four simulation studies with different number of ROIs and subjects. In the first simulation, we generated data with two ROIs for single subject. We varied the correlation between ROIs from 0 to 1. At each voxel, we use AR(1) structure to generate temporally correlated time series with length 128. The AR(1) parameter was 0.6. In the second simulation, the total length of each time series was 128 as well, but we use AR(1) and AR(2) structure alternately with length 32 each to generate non-stationary time series data. The AR(1) parameter was 0.6 and AR(2) parameter was 0.3. Other settings were the same as the first simulation. In the third simulation, we generated data with five ROIs for one subject.

These five ROIs have the covariance matrix π_d in Model (3.3) as

$$\begin{pmatrix} 1.0 & 0.6 & 0.0 & 0.5 & 0.0 \\ 0.6 & 1.0 & 0.2 & 0.6 & 0.0 \\ 0.0 & 0.2 & 1.0 & 0.0 & 0.1 \\ 0.5 & 0.6 & 0.0 & 1.0 & 0.2 \\ 0.0 & 0.0 & 0.1 & 0.2 & 1.0 \end{pmatrix}$$

In the fourth simulation, we generated data with two ROIs for multiple subjects. We varied the correlation between ROIs from 0 to 1 and three to ten subjects were used in the analysis. In the third and fourth simulation, we use AR(1) structure with parameter 0.6 to generate temporally correlated time series with length 128.

Three different spatial correlations were used in each simulation study. First we assumed all voxels were independent within an ROI, i.e., no spatial correlation. Secondly, we assumed that the spatial correlation was from an exponential covariance function with the decaying parameter 0.5, i.e., moderate spatial correlation. Thirdly, we assumed that all time series were the same across voxels within an ROI, i.e., extreme spatial correlation. In all three simulation studies, each ROI contained 100 voxels (10×10). Each spatial correlation scenario were based on 500 repetitions in four simulation studies.

3.4.2 Estimation and Results

Different wavelet functions were tested for both spatial wavelet function and temporal wavelet function. Using MATLAB (MathWorks, Natick, MA) wavelet toolbox, we tried 54 different wavelet functions, which resulted in total 2916 (54×54) combinations of the double-wavelet transform. In simulation, we found that using the Reverse Biorthogonal 3.1 wavelet (Rbio3.1) on the spatial data, and using the Haar wavelet (Haar) on the temporal data minimized the MSE. This combination was used for all subsequent simulation studies and data analysis.

3.4.2.1 Single Subject - Two ROIs - Stationary time series

Figure 3.2 shows the MSE, Bias^2 and Variance for the DW and AVG-FC approach when the underlying correlation between two ROIs varied from 0 to 1 for three different types of spatial correlation. The underlying temporal correlation for each voxel was AR(1). When all voxels were uncorrelated, the DW approach had smaller MSE because the gain in variance dominates the loss in bias, compared to AVG-FC. When the spatial correlations among voxels were based on exponential covariance function, the DW approach outperformed the AVG-FC approach in terms of bias^2, variance and MSE. The DW approach had 1/4 of the AVG-FC approach's bias and variance, and 1/8 of the AVG-FC approach's MSE. When all voxels were identical, the AVG-FC approach had smaller MSE because the AVG-FC approach had smaller variance, while there was no discernible difference in bias between the two approach.

3.4.2.2 Single Subject - Two ROIs - Non-stationary time series

Figure 3.3 shows the MSE, Bias^2 and Variance for the DW and AVG-FC approach when the underlying correlation between two ROIs varied from 0 to 1 for three different types of spatial correlation. The underlying temporal correlation for each voxel was a combination of AR(1) and AR(2). The results showed similar pattern as simulation 1. When all voxels were uncorrelated, the DW approach had smaller MSE. When the spatial correlations among voxels were based on exponential covariance function, the DW approach outperformed the AVG-FC approach in terms of bias^2, variance and MSE. The DW approach had 1/3 of the AVG-FC approach's bias^2, variance, and MSE. When all voxels were identical, the AVG-FC approach had smaller MSE.

3.4.2.3 Single Subject - Five ROIs - Stationary time series

For this simulation, there were five correlated ROIs. Figure 3.4 shows the MSE for the DW and AVG-FC approach. The DW approach outperformed the AVG-FC approach in

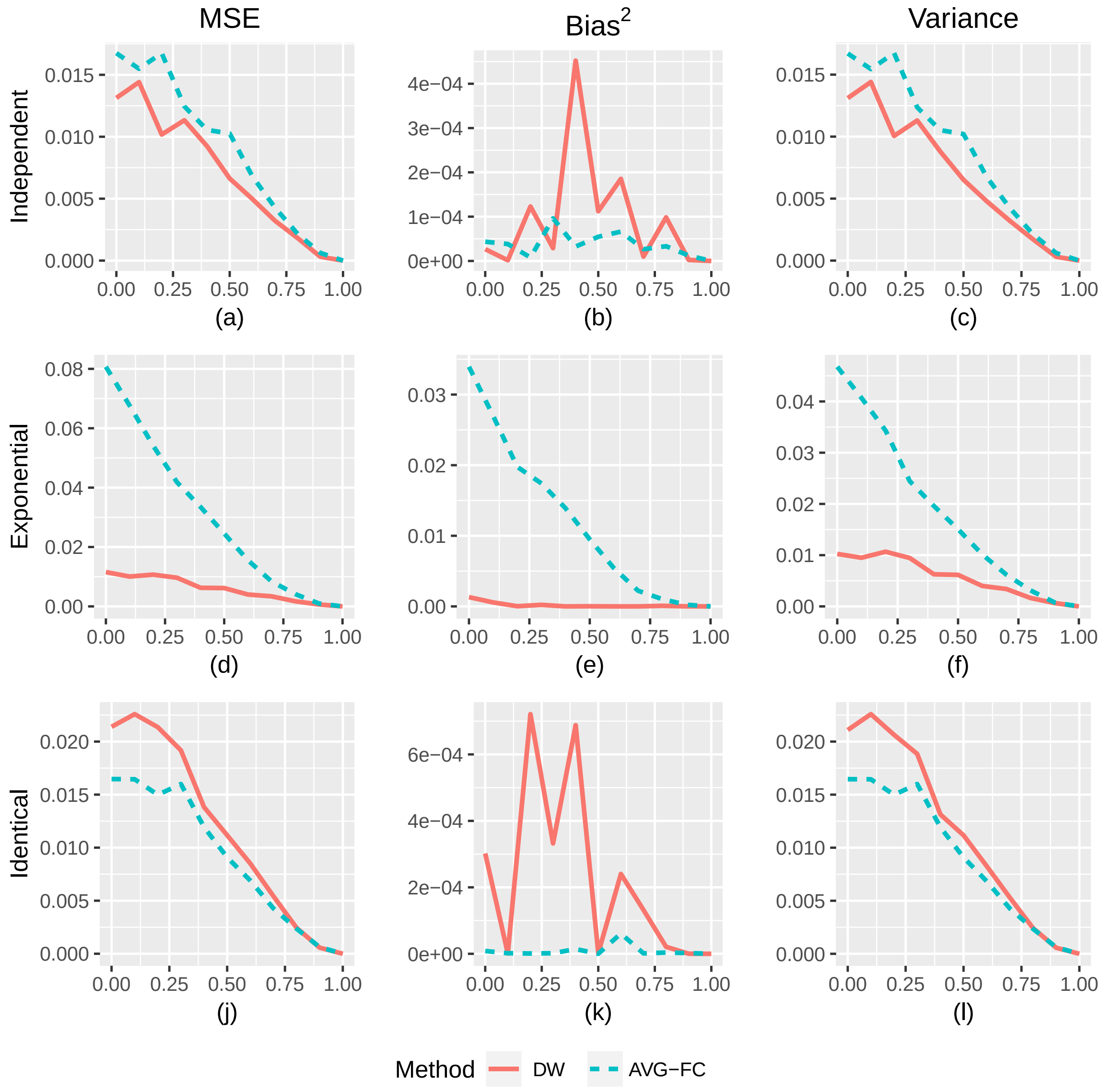


Figure 3.2: MSE, Bias^2 and Variance for the DW and AVG-FC approach based on single subject analysis between two ROIs for stationary time series. X-axis is the underlying correlation.

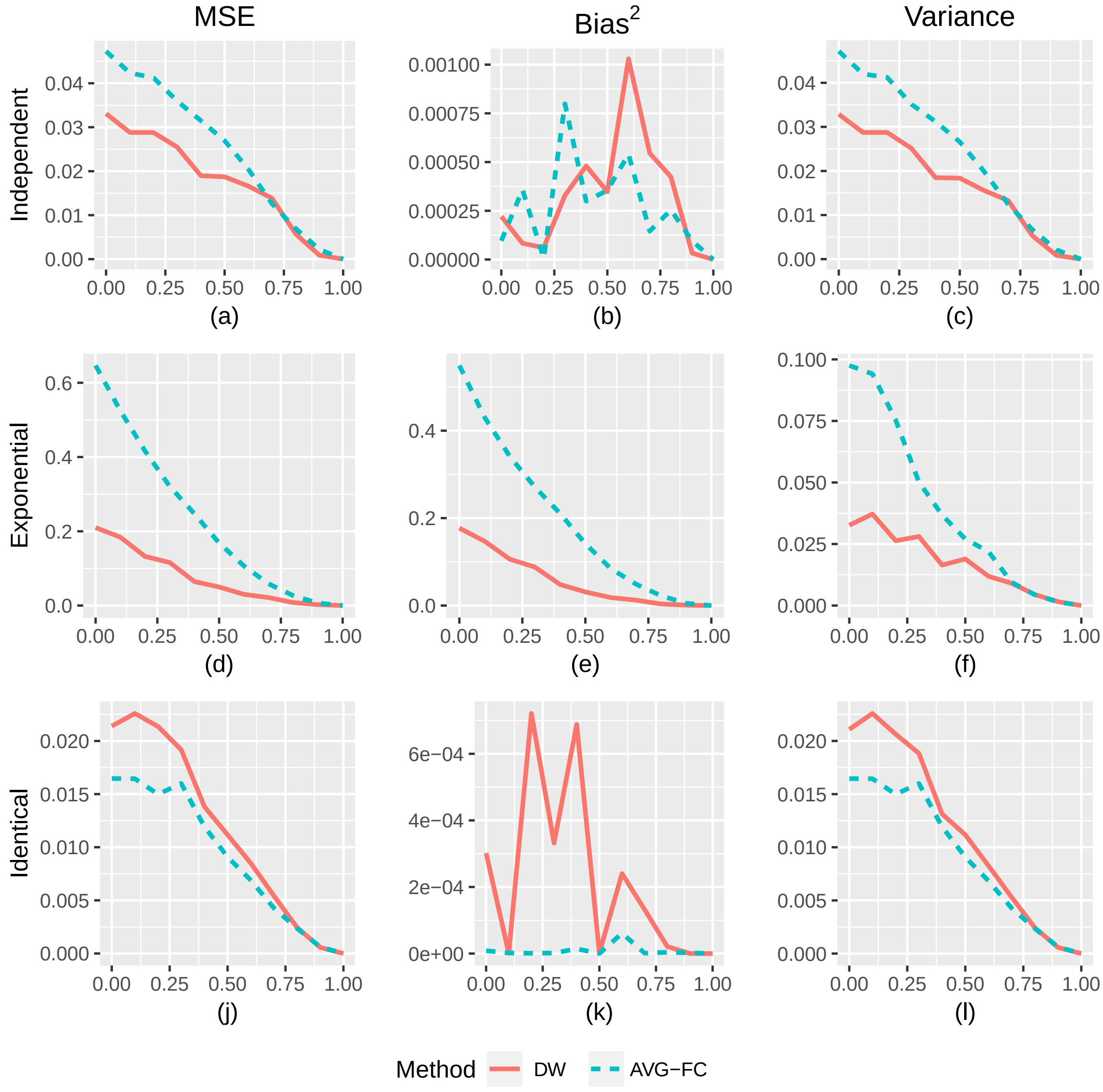


Figure 3.3: MSE, Bias^2 and Variance for the DW and AVG-FC approach based on single subject analysis between two ROIs for non-stationary time series. X-axis is the underlying correlation.

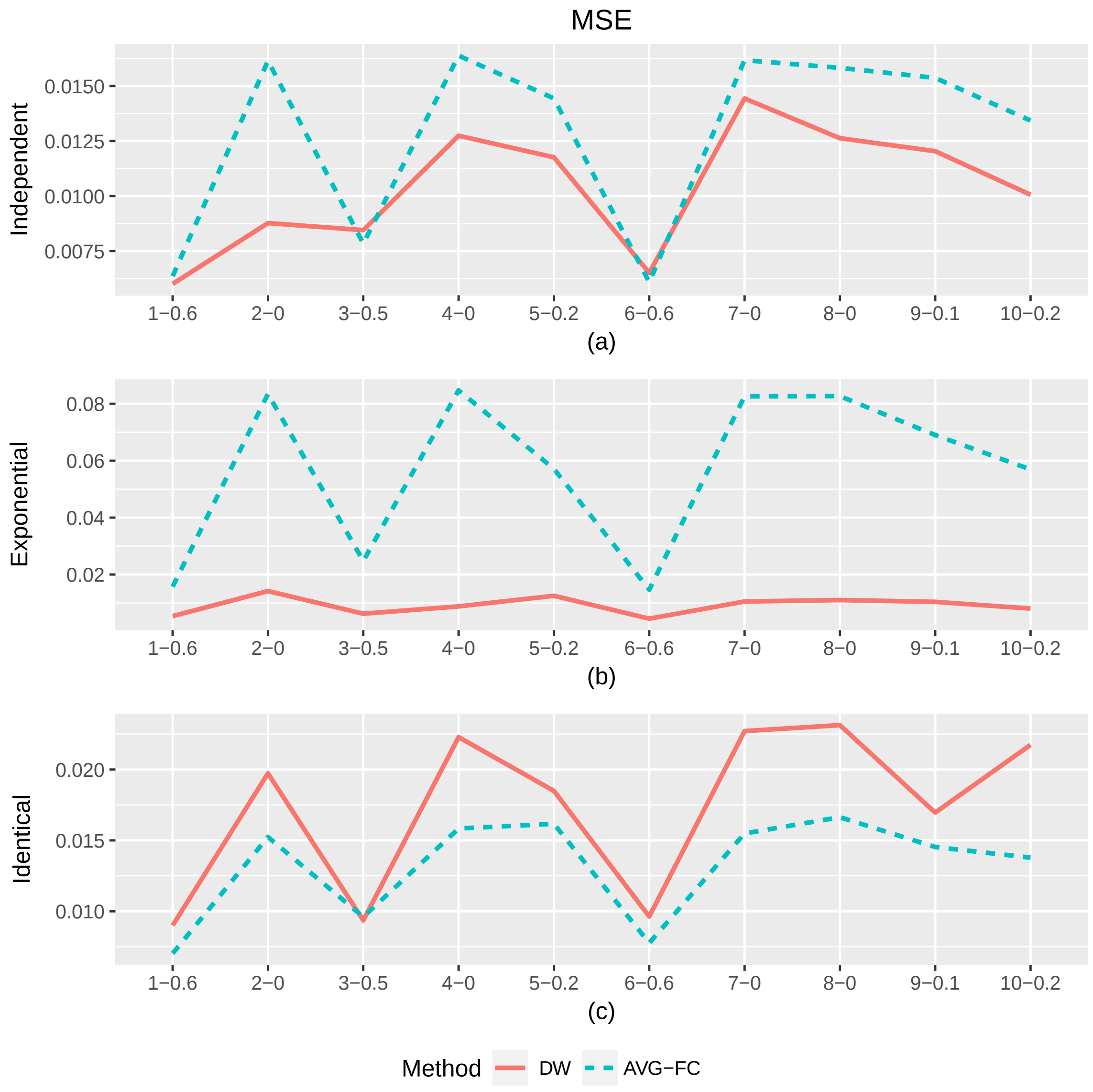


Figure 3.4: MSE for the DW and AVG-FC approach based on single subject analysis among five ROIs. X-axis is the underlying correlation for different pairs, for example, “1-0.6” means the first pair with true correlation 0.6.

terms of MSE when all voxels were uncorrelated and when the spatial correlations among voxels were based on exponential covariance function. When the underlying correlation are 0.5 and 0.6, the difference in MSE between two methods are small. When all voxels were identical, the AVG-FC approach had smaller MSE but this extreme spatial correlation would not be feasible in practice. These results were consistent with the results using two ROIs.

3.4.2.4 Multiple Subjects - Two ROIs - Stationary time series

Figure 3.5 shows rejection rate of the correlation between two ROIs for the DW and AVG-FC approach when the number of subjects varied from 3 to 20 for three different types of spatial correlation and the underlying true correlation were 0, 0.2 and 0.5.

When all voxels were uncorrelated and there were no the underlying correlation between ROIs, the DW approach and the AVG-FC approach had similar rejection rate when there were less than 15 subject, after that, the DW approach had smaller rejection rate. The DW approach had slightly larger rejection rate than the AVG-FC approach when the true underlying correlation was 0.2. There was no difference between two methods when the underlying correlation was 0.5 and when we had more than 4 subjects.

When the spatial correlations among voxels were based on exponential covariance function, the rejection rate using AVG-FC approach increased as number of subjects went up when the underlying correlation between two ROIs was 0, which means the AVG-FC approach had too strong power to detect small difference. However, the reject rate using the DW approach decreased as the number of subjects increased when there were no underlying correlation between two ROIs. When the underlying correlation between two ROIs was 0.2, the DW approach had slightly higher rejection rate than the AVG-FC approach. There were no difference between two methods when the underlying correlation was 0.5.

When all voxels were identical, the DW approach and the AVG-FC approach had similar rejection rate when the underlying correlation between ROIs was 0. The DW approach

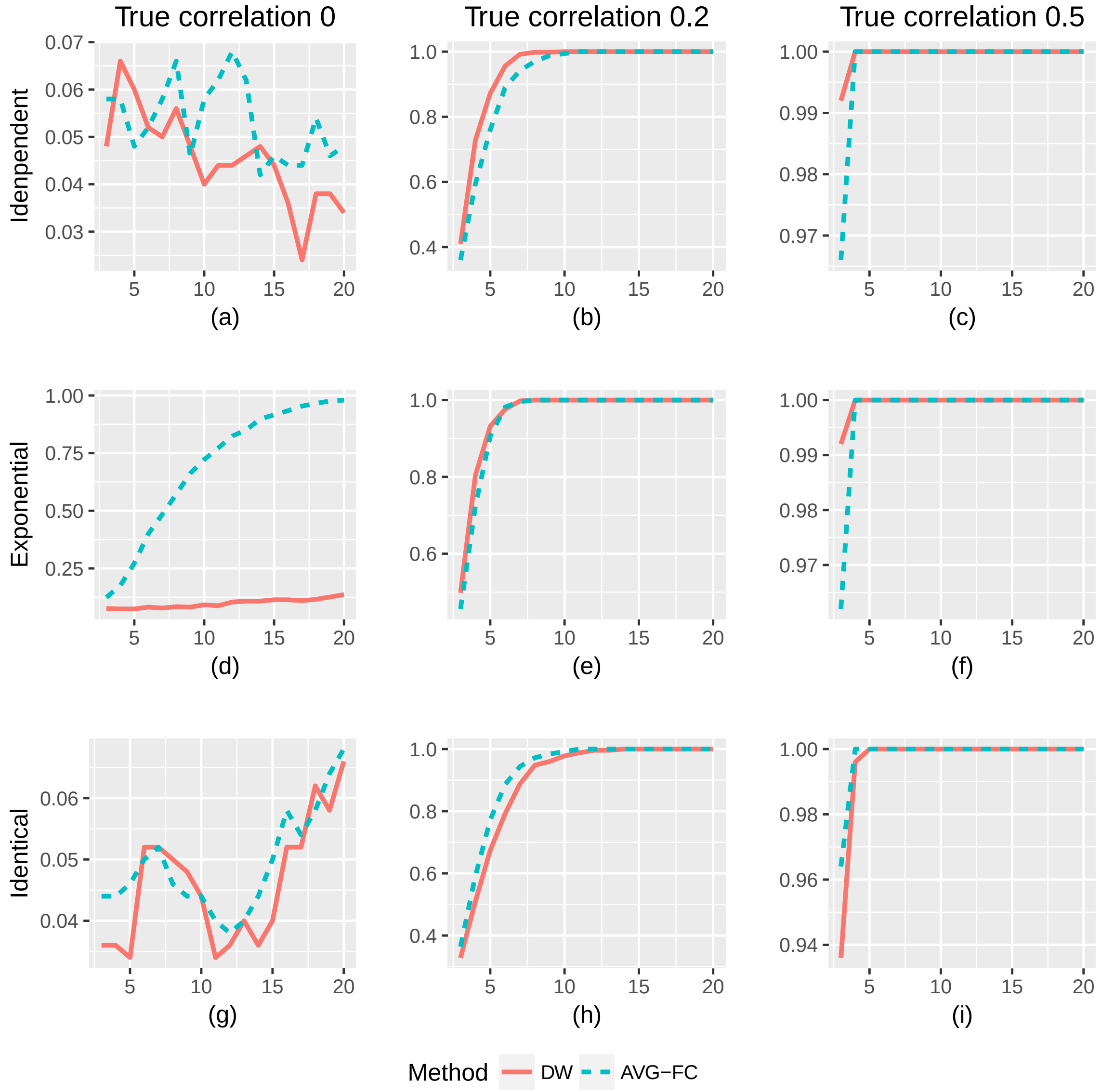


Figure 3.5: Rejection rates for the DW and AVG-FC approach based on multiple subject analysis between two ROIs. X-axis is the number of subjects.

had smaller rejection rate than the AVG-FC approach when the true underlying correlation was 0.2. There was no difference between two methods when the underlying correlation was 0.5 and when we had more than 4 subjects.

The DW approach outperformed the AVG-FC approach in all practical settings, although the AVG-FC approach outperformed the DW approach in some unrealistic settings, i.e., extreme spatial correlation.

3.5 Data Analysis

We applied our proposed double-wavelet transform approach on a study designed to investigate the functional connectivity in major depressive disorder (MDD) patients. Here we describe the background, motivation, study design and the data.

In a previous study (Kang et al., 2016), brain network properties associated with major depressive disorder (MDD) was established using resting-state fMRI data. They found that 35 regions out of 90 automated anatomical labeling (AAL) regions showed significant differences in functional connectivity between the healthy control (HC) subjects and MDD groups. In this paper, we investigated these 35 regions using a data set (Taylor et al., 2016; Saleh et al., 2017), which included 12 subjects (6 HC and 6 MDD patients) matched by their age, race, sex and education level. The average age is 27.7 years for the MDD group, and 27.5 years for the controls; both groups are 50% male.

Functional images were acquired using a gradient-echo echo-planar sequence (TR = 2 sec; TE = 27 msec; flip angle = 77; 32 axial slices, 4 × 4 × 4 mm) on the Siemens 3T TIM Trio MRI system at the Duke University MRI Research Facility. High-resolution T1-weighted (MP-RAGE) anatomical images were collected for visualization (TR = 2300 msec; TE = 3.46 sec; 160 sagittal slices, 0.9 × 0.9 × 1.2 mm).

Data were preprocessed and analyzed using FSL (https://fsl.fmrib.ox.ac.uk) and AFNI (Analysis of Functional NeuroImages, https://afni.nimh.nih.gov). Following quality assurance procedures to assess outliers or artifacts in volume and slice-to-slice variance in the

Table 3.1: List of ROIs showing significant difference between HC and MDD groups using DW approach and AVG-FC approach

Pair	Method	HC	MDD	HC-MDD	P-value
Angular_L - Frontal_Mid_R	DW	0.13	-0.05	0.18	0.04
Frontal_Sup_L - Temporal_Inf_R	DW	0.09	-0.05	0.14	0.01
Frontal_Sup_Orb_R - Rectus_R	DW	0.26	0.39	-0.14	0.02
Parietal_Sup_R - Cingulum_Ant_R	DW	-0.27	-0.10	-0.18	0.01
ParaHippocampal_L - Paracentral_Lobule_R	DW	0.00	-0.11	0.11	0.04
ParaHippocampal_L - Paracentral_Lobule_R	AVG-FC	0.00	-0.18	0.18	0.01

global signal, functional images were corrected for differences in slice acquisition timing by re-sampling all slices in time to match the first slice. Images were then motion corrected. Functional data were then normalized based on Montreal Neurological Institute stereotaxic space, and spatially smoothed for the AVG-FC approach but not for the double wavelet approach.

We applied the 3-D wavelet transform using Rbio3.1 wavelet on the volume data at each time point, and then applied the 1-D wavelet transform using Haar wavelet on the time series of each 3-D wavelet coefficient. The double wavelet weighted correlation as in equation (3.5) were calculated for each subject at each ROI pair. Then p-values were computed based on t-test between HC and MDD groups.

The results showing significant difference between two groups were listed in Table 3.1. Out of 35 significant region pairs in Kang et al. (2016), we found 5 region pairs showed significant results using the DW approach, and only one region pair was found significant using the AVG-FC approach. After controlling False Discovery Rate (FDR) at 0.1 level, all significance disappeared.

There could be several reasons why we were not able replicate the results in Kang's paper. First, the patients in our study were younger than the patients in Kang's study (27.5 years vs 44.1 years). Second, Kang performed binomial tests on the differences between the two groups in the proportion of subjects showing strong functional connections. In our method, we test the difference in the mean correlation between two groups. Although the significance disappeared when we control for FDR at 0.1 level, our method tends to detect more regions pairs than the AVG-FC approach out of 35 significant pairs.

3.6 Conclusion

In this article, we proposed a double-wavelet transform approach, where we transformed the data twice using different wavelet functions, and combined all information from different frequency bands using weighted Pearson correlation. There are several advantages to use double-wavelet transform. First, the spatial-temporal correlation in fMRI data as into consideration by transforming the data twice using different wavelet functions. Second, due to a natural property of wavelet transform, wavelet transform released the stationary assumptions on both spatial and temporal data, which may not be feasible but is necessary for most of available tools in fMRI data analysis. Third, estimation and inversion of the covariance matrix were not required as in spatio-temporal modeling. This significantly reduced computational burden in fMRI data analysis. Fourth, the double-wavelet transform simplified the fMRI data structure by converting 4-D data into 2-D data.

In simulation, we suggested to use the Reverse Biorthogonal 3.1 wavelet on the spatial data and the Haar wavelet on the temporal data. For single subject analysis, we found that the DW approach outperformed the AVG-FC approach in terms of MSE when all voxels were independent and the spatial correlations among voxels were moderate.

For multiple subject analysis, we found that the DW approach had lower false positive rate than the AVG-FC approach as the number of subject increased, when the spatial correlations among voxels were based on exponential covariance function and when all voxels were identical. The DW approach had lower false negative rate than the AVG-FC approach as the number of subject increased, when the spatial correlations among voxels were based on exponential covariance function and when all voxels were independent. Note that the two extreme cases in spatial correlation, i.e., observations are all identical across voxels or are all independent, are not realistic. Therefore, it can be claimed that the DW approach outperformed the AVG-FC approach when the underlying spatial correlation is realistic.

In data analysis, we found five ROI pairs showed significant result using the DW approach, and one ROI pair using the AVG-FC approach. The five ROI pairs had mean correlation difference between 0.1 and 0.2. These results were consistent with our simulation study, meaning that the DW approach made less mistakes than the AVG-FC approach by properly taking into account the spatio-temporal correlation as shown in Figure 3.5. Although the significance disappeared when we control for FDR at 0.1 level, our method would be able to find more significant region pairs with a small sample size than the AVG-FC approach in the 35 region pairs in Kang et al. (2016).

One limitation of our approach is the subjective choice of the wavelet functions. Different wavelet functions have various performance when decorrelating different correlation structure. More research can be done for different levels of wavelet transform in the future.

Appendix B

Matlab Code for simulation

```
%%%%% This file is for analyzing using Double wavelet approach

clear all

local=0

if local==1
    disp('Running on local machine')
    addpath '/Users/Michaelzmc/matlab/NIfTI_20140122/';
    addpath '/Users/Michaelzmc/matlab/matlabFunctions/';
    addpath '/Users/Michaelzmc/Documents/matlab/spm12/';

    else
    disp('Running on Accre')
    addpath '/home/zhoum5/Matlab_function/NIfTI_20140122/';
    addpath '/home/zhoum5/Matlab_function/matlabFunctions/';
    addpath '/home/zhoum5/Matlab_function/spm8/';

end


```

```matlab

Nsimu = 1;
Nsubj = 500;
dim = [10 10];


phi = 0.6;
T = 128; t = 128;
R = 5; num_ROI = 5;
Ncor = nchoosek(R,2);

Voxel = 100;
alpha = 0.1;
N.level = 1;


% for FDR control
increment=alpha/R;
FDR_vec = [increment:increment:alpha];
wave_FDR_result_glm_sim = zeros(Nsimu, num_ROI);

num_covN = 2;

row=1;

SIgma = [1, 0.6, 1e-10, 0.5, 1e-10;
          0.6, 1, 0.2, 0.6, 1e-10;
```

```
         1e-10, 0.2, 1, 1e-10, 0.1;
         0.5, 0.6, 1e-10, 1, 0.2;
         1e-10, 1e-10, 0.1, 0.2, 1];

truthtemp = tril(SIgma, -1);
truth = (truthtemp(truthtemp ~=0))';

truth(truth == 1) = 0;

waveTname = 'haar'
wavePname = 'rbio3.1'


            % get 2d wavelet coefficient size
            [C, S]  = wavedec2(zeros(dim(1), dim(2)), N.level ,wavePname);

            % number of low-low wavelet coefficient
            plot.size = S(1,1)*S(1,2);
            %plot.size = sum(S(:,1) .* S(:,2));
            coefband = 1:length(C);

            % number of 2d wavelet coefficient
            N.wavecoef = length(C);

            % create wavelet coefficient matrix
            waveC = zeros( T, length(C));
```

```
        % 1D wavelet coefficient
        [TC,TS] = wavedec(zeros(1,T),1,waveTname);

        Ntime = length(TC);
        waveband = 1:Ntime ;

        waveT = zeros(length(C), length(TC) );

        wave_pval_glm = zeros(R,Nsimu);

        wavecon = zeros( Nsimu, R,Nsubj, length(coefband));
        wavecon_sub = zeros(Nsubj, R);

wavedata = zeros(R,Nsubj, length(coefband), length(waveband));

origmean = zeros(Nsimu, R, Nsubj, Ntime);
origcor = zeros( Nsubj, Ncor);
origcor1 = zeros( Nsubj, Ncor);


origcormean = zeros(Nsimu, Ncor);
origcormeanpar = zeros(Nsimu, Ncor);

mean_ROI= zeros(R, Nsubj, Ntime);

```

```

count = 0;

result = cell(9, 11);

datatpyeall =  { 'Exp2', 'gau0', 'gau100' };
cortype = {'reg', 'par'}


for dtype = 1:length(datatpyeall)

    datatpye = datatpyeall{dtype}

        sim=1;

cd('/scratch/kangh1/minchun/RestData/cor_simu_fiveroi')

            load(sprintf('%s_Sim%d_five.mat',datatpye, sim))
               ;
                data = data(1:Nsubj,:,:,:,:);

             for sub = 1:Nsubj

                for roi = 1:R

                        for t = 1:T
                            subjbrain = reshape(data(sub,roi
                                ,:,:,t), dim(1), dim(2));
                            [waveC(t,:), S] = wavedec2(
                                subjbrain, N.level ,wavePname
                                );
                        end

                        for coef = 1:N.wavecoef

                            subjT = reshape(waveC(:,coef),1,
                                T);
                            [waveT(coef, :),TS] = wavedec(
                                subjT,1,waveTname);

                         end


                        wavedata( roi,sub, :,:) = waveT(
                            coefband,waveband);



                    end

                end

        sdata = size(wavedata);
        compcor = zeros(100,10);

        wavecor = zeros(100,10);
        wavepar = zeros(100,10);

        spadata1 = (1:(sdata(3)*1/4)) + 0* (sdata(3)*1/4) ;
        spadata2 = (1:(sdata(3)*1/4)) + 1* (sdata(3)*1/4) ;
        spadata3 = (1:(sdata(3)*1/4)) + 2* (sdata(3)*1/4) ;
        spadata4 = (1:(sdata(3)*1/4)) + 3* (sdata(3)*1/4) ;

        spadataAll = [spadata1;spadata2;spadata3;spadata4];

        tempdataA = (1:(sdata(4)/2)) + 0*(sdata(4)/2);
        tempdataB = (1:(sdata(4)/2)) + 1*(sdata(4)/2);

        tempdataAll = [tempdataA; tempdataB];

        indTemp = ['A' ;'B'];
        variance = zeros(4,2,100);

        for spa = 1:4

           spadata = spadataAll(spa,:);

                tempdata = 1:(sdata(4));

             for sim = 1:100

```

```
        d1 = reshape( mean( wavedata(1,sim,spadata,tempdata),3),length(tempdata),1);
        d2 = reshape( mean( wavedata(2,sim,spadata,tempdata),3),length(tempdata),1);
        d3 = reshape( mean( wavedata(3,sim,spadata,tempdata),3),length(tempdata),1);
        d4 = reshape( mean( wavedata(4,sim,spadata,tempdata),3),length(tempdata),1);
        d5 = reshape( mean( wavedata(5,sim,spadata,tempdata),3),length(tempdata),1);


        tempcor = corr([d1 d2 d3 d4 d5]);
        temppar = partialcorr([d1 d2 d3 d4 d5]);

        dia_tempcor = tril(tempcor, -1);
        vec_tempcor = (dia_tempcor(dia_tempcor ~=0))';

        dia_temppar = tril(temppar, -1);
        vec_temppar = (dia_temppar(dia_temppar ~=0))';


        wavecor(sim, :) = vec_tempcor;
        wavepar(sim, :) = vec_temppar;

```

```
         end


        for j = 1:10

                count = count+1;

                correlation = truth(j);

                allsub = wavecor(:,j);

                result{count,1} = correlation;
                result{count,2} = datatpye;
                result{count,3} = wavePname;
                result{count,4} = waveTname;

                result{count,5} = mean(allsub - correlation);
                result{count,6} = var(allsub - correlation);
                result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

                result{count,8} = spa;
                result{count,9} = 'Alltemp';
                result{count,10} = cortype{1};
                result{count,11} = j;

                count = count+1;

                correlation = truth(j);

                allsub = wavepar(:,j);

                result{count,1} = correlation;
                result{count,2} = datatpye;
                result{count,3} = wavePname;
                result{count,4} = waveTname;

                result{count,5} = mean(allsub - correlation);
                result{count,6} = var(allsub - correlation);
                result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

                result{count,8} = spa;
                result{count,9} = 'Alltemp';
                result{count,10} = cortype{2};
                result{count,11} = j;


```

```
        end


        for temp = 1:2

            tempdata = tempdataAll(temp,:);

         for sim = 1:100

            d1 = reshape( mean( wavedata(1,sim,spadata,tempdata),3),length(tempdata),1);
            d2 = reshape( mean( wavedata(2,sim,spadata,tempdata),3),length(tempdata),1);
            d3 = reshape( mean( wavedata(3,sim,spadata,tempdata),3),length(tempdata),1);
            d4 = reshape( mean( wavedata(4,sim,spadata,tempdata),3),length(tempdata),1);
            d5 = reshape( mean( wavedata(5,sim,spadata,tempdata),3),length(tempdata),1);

            tempcor = corr([d1 d2 d3 d4 d5]);
            temppar = partialcorr([d1 d2 d3 d4 d5]);

            dia_tempcor = tril(tempcor, -1);
            vec_tempcor = (dia_tempcor(dia_tempcor ~=0))';

            dia_temppar = tril(temppar, -1);
            vec_temppar = (dia_temppar(dia_temppar ~=0))';


            wavecor(sim, :) = vec_tempcor;
            wavepar(sim, :) = vec_temppar;

         end

        for j = 1:10

                count = count+1;

                correlation = truth(j);

                allsub = wavecor(:,j);

                result{count,1} = correlation;
                result{count,2} = datatpye;
                result{count,3} = wavePname;
                result{count,4} = waveTname;

                result{count,5} = mean(allsub - correlation);
                result{count,6} = var(allsub - correlation);
                result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

                result{count,8} = spa;
                result{count,9} = indTemp(temp);
                result{count,10} = cortype{1};
                result{count,11} = j;



                count = count+1;

                correlation = truth(j);

                allsub = wavepar(:,j);

                result{count,1} = correlation;
                result{count,2} = datatpye;
                result{count,3} = wavePname;
                result{count,4} = waveTname;

                result{count,5} = mean(allsub - correlation);
                result{count,6} = var(allsub - correlation);
                result{count,7} = (mean(allsub -
                   correlation))^2 + var(allsub -
                   correlation);

                result{count,8} = spa;
                result{count,9} = indTemp(temp);
                result{count,10} = cortype{2};
                result{count,11} = j;


        end
            end
        end



 % here

wavecor = zeros(4,2, 100,10);
wavepar = zeros(4,2, 100,10);

compcor = zeros(100,10);

for sim = 1:100

    for spa = 1:4

        for temp = 1:2

        spadata = spadataAll(spa,:);
        tempdata = tempdataAll(temp,:);

            d1 = reshape( mean( wavedata(1,sim,spadata,tempdata),3),length(tempdata),1);
            d2 = reshape( mean( wavedata(2,sim,spadata,tempdata),3),length(tempdata),1);
            d3 = reshape( mean( wavedata(3,sim,spadata,tempdata),3),length(tempdata),1);
            d4 = reshape( mean( wavedata(4,sim,spadata,tempdata),3),length(tempdata),1);
            d5 = reshape( mean( wavedata(5,sim,spadata,tempdata),3),length(tempdata),1);

            tempcor = corr([d1 d2 d3 d4 d5]);
            temppar = partialcorr([d1 d2 d3 d4 d5]);

            dia_tempcor = tril(tempcor, -1);
            vec_tempcor = (dia_tempcor(dia_tempcor ~=0))';

            dia_temppar = tril(temppar, -1);
            vec_temppar = (dia_temppar(dia_temppar ~=0))';

            wavecor(spa, temp, sim,:) = vec_tempcor;
```

```
      wavepar(spa, temp, sim,:) = vec_temppar;

    var1 = var(reshape(wavedata(1,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1));
    var2 = var(reshape(wavedata(2,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1));
    var3 = var(reshape(wavedata(3,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1));
    var4 = var(reshape(wavedata(4,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1));
    var5 = var(reshape(wavedata(5,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1));

    variance(spa, temp, sim) = mean([var1 var2 var3
      var4 var5])   ;



    vector1 = reshape(wavedata(1,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
      ,1);
    vector2 = reshape(wavedata(2,sim,spadata,
      tempdata), length(spadata)*length(tempdata)
```

```
,1);
        vector3 = reshape(wavedata(3,sim,spadata,tempdata), length(spadata)*length(tempdata),1);
        vector4 = reshape(wavedata(4,sim,spadata,tempdata), length(spadata)*length(tempdata),1);
        vector5 = reshape(wavedata(5,sim,spadata,tempdata), length(spadata)*length(tempdata),1);

        var1 = var(vector1.* vector1);
        var2 = var(vector2.* vector2);
        var3 = var(vector3.* vector3);
        var4 = var(vector4.* vector4);
        var5 = var(vector5.* vector5);


        variance_waveform(spa, temp, sim) = mean([var1 var2 var3 var4 var5])  ;


     end

    end

    tempvar = sum( reshape(variance(:,:,sim), 1,8) );
```

```
        tempvar_waveform = sum( reshape(variance_waveform(:,:,sim), 1,8) );

        for j = 1:10

            compcor(sim, j) = sum(sum(reshape(variance(:, :, sim), 4,2) .* reshape(wavecor(:, :, sim,j), 4,2)))/tempvar;
            comppar(sim, j) = sum(sum(reshape(variance(:, :, sim), 4,2) .* reshape(wavepar(:, :, sim,j), 4,2)))/tempvar;

            compcor_waveform(sim, j) = sum(sum(reshape(variance_waveform(:, :, sim), 4,2) .* reshape(wavecor(:, :, sim,j), 4,2)))/tempvar_waveform;
            comppar_waveform(sim, j) = sum(sum(reshape(variance_waveform(:, :, sim), 4,2) .* reshape(wavepar(:, :, sim,j), 4,2)))/tempvar_waveform;

        end

    end


                for j = 1:10


                correlation = truth(j);

                count = count+1;

                allsub = compcor(:,j);

                result{count,1} = correlation;
                result{count,2} = datatpye;
                result{count,3} = wavePname;
                result{count,4} = waveTname;

                result{count,5} = mean(allsub - correlation);
                result{count,6} = var(allsub - correlation);
                result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

                result{count,8} = 'comp_variance';
                result{count,9} = 'comp_variance';
                result{count,10} = cortype{1};
                result{count,11} = j;

                count = count+1;
```

```

            allsub = comppar(:,j);

            result{count,1} = correlation;
            result{count,2} = datatpye;
            result{count,3} = wavePname;
            result{count,4} = waveTname;

            result{count,5} = mean(allsub - correlation);
            result{count,6} = var(allsub - correlation);
            result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

            result{count,8} = 'comp_variance';
            result{count,9} = 'comp_variance';
            result{count,10} = cortype{2};
            result{count,11} = j;



            correlation = truth(j);

            count = count+1;

            allsub = compcor_waveform(:,j);

            result{count,1} = correlation;
            result{count,2} = datatpye;
            result{count,3} = wavePname;
            result{count,4} = waveTname;

            result{count,5} = mean(allsub - correlation);
            result{count,6} = var(allsub - correlation);
            result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

            result{count,8} = 'comp_waveform';
            result{count,9} = 'comp_waveform';
            result{count,10} = cortype{1};
            result{count,11} = j;

            count = count+1;

            allsub = comppar_waveform(:,j);

            result{count,1} = correlation;
            result{count,2} = datatpye;
            result{count,3} = wavePname;
```

```
                    result{count,4} = waveTname;

                    result{count,5} = mean(allsub - correlation);
                    result{count,6} = var(allsub - correlation);
                    result{count,7} = (mean(allsub - correlation))^2 + var(allsub - correlation);

                    result{count,8} = 'comp_waveform';
                    result{count,9} = 'comp_waveform';
                    result{count,10} = cortype{2};
                    result{count,11} = j;


                end

end


    cell2csv('/scratch/kangh1/minchun/RestResult/dw_five_0325_waveform.csv',result)
```

Chapter 4

MATLAB GUI FOR MULTI-SUBJECT TASK-INDUCED FUNCTIONAL MAGNETIC RESONANCE IMAGING DATA USING DOUBLE-WAVELET TRANSFORM

One of the main challenge in functional magnetic resonance imaging (fMRI) data analysis is the high computational cost due to the complex spatio-temporal correlation. Using wavelet transform can help reduce the spatio-temporal correlation, as well as save the computation time. In this paper, we develop a MATLAB graphical user interface (GUI) for multi-subject task-induced fMRI data using double-wavelet transform, which can estimate the effect of user-specified stimulus functions in pre-specified region of interests (ROI).

4.1 Introduction

Task-induced functional magnetic resonance imaging (fMRI) is one of the technologies for studying human brain activity in response to external stimuli. The conventional analysis of fMRI data only takes into account the temporal correlation, e.g., auto-regressive order one (AR(1)) structure, and estimates the parameters from the mean time series of each region of interest (ROI). For multiple subjects, a simple t-test is used on the parameters (Worsley & Friston, 1995; Weiskopf et al., 2003; Huettel et al., 2004). This approach does not rigorously model the underlying spatial correlation., which led to smaller standard errors and higher Type I errors Dubin (1988).

Spatial smoothing is common approaches that take into account spatial correlation in fMRI data analysis. Spatial smoothing using a Gaussian kernel was proposed by Worsley et al. (1996). This method was implemented in statistical parametric mapping (SPM) by Frackowiak et al. (1997). Although spatial smoothing increased the signal to noise ratio (SNR) in fMRI data, it actually induce more spatial correlation, which may cause a higher error rate.

The wavelet transform is a linear transformation, and the wavelet coefficients we get are approximately uncorrelated Fan (2003). The wavelet transform was first introduced to fMRI analysis by Brammer (1998) and Ruttimann et al. (1998). Aston et al. (2005) proposed estimating the model coefficients in the wavelet domain by applying one wavelet transform on the spatial data at each time point.

We develop a novel single level double-wavelet framework that estimating the model coefficients in the wavelet domain by applying the wavelet transform twice on both spatial data and temporal data. This approach takes into account the spatial and temporal correlation for estimating the ROI-level activation patterns in multi-subject fMRI data analysis.

This paper provides a MATLAB graphical user interface (GUI) using the single level double-wavelet transform for multiple subject fMRI data analysis. The GUI aims to be useful for both beginners and advanced user of MATLAB. The GUI incorporates a user-friendly interface that helps to control the whole fMRI analysis by (a) importing data, (b) reading stimulus functions (c) selecting region of interest (ROI), (d) single subject analysis, (e) group analysis.

The remaining sections are organized as follows. In section 2, the spatio-temporal model for task-induced fMRI data is presented, and then double-wavelet transform is introduced. In section 3, we illustrate the design of GUI, and using example data to show how to use the GUI step by step. The summary and future work are presented in section 4.

4.2 Method

4.2.1 Model

In fMRI study, suppose we have N subjects, P external stimuli, C ROIs and V_c voxels within the c-th ROI. For each voxel v in ROI c for subject n, we define the time series as $Y_{ncv}(t), t = 1, ..., T$, where T is the length of time series. There are two different correlations we need to consider: the spatial correlation between voxels within an ROI, and the temporal correlation within a voxel. Here is the model to describe these two different correlation in fMRI data:

$$Y_{ncv}(t) \quad = \quad \sum_{p=1}^{P} [\beta_{ncv}^{p} X^{p}(t)] + \varepsilon_{ncv}(t), \ \ \text{where} \tag{4.1}$$

$$\beta_{ncv}^{p} = \beta_{c}^{p} + b_{ncv}^{p}$$

- $X^{p}(t)$ is the expected BOLD response corresponding to the p^{th} stimulus which is formally the convolution between the HRF and the p^{th} impulse function. The HRF is the expected neuronal activation function given a stimuli. .
- β_{c}^{p} is the ROI-specific activation level fixed effect due to stimulus p;
- b_{ncv}^{p} is a zero-mean voxel-specific random effect that accounts for the spatial covariance between voxels v and v' within ROI c for subject n.
- $\varepsilon_{ncv}(t)$ is the noise that takes into account the voxel-specific temporal correlation, which is assumed to follow an AR(1) process.

To test whether ROI c is activated when the p^{th} stimulus is presented, we are interested in the hypothesis:

$$H_0 : \beta_{c}^{p} - \beta_{c}^{1} = 0 \tag{4.2}$$

where β_{c}^{1} indicates the baseline condition at ROI c.

4.2.2 Wavelet Transform

Similar to Fourier transform, wavelet transform is a linear transform. Wavelet coefficients are obtained by the inner product of the observed data and wavelet functions, which is similar to the sine and cousin functions in Fourier transform. There are two types of wavelet transform, zero integral mother wavelet function and the unit integral father wavelet function. Wavelet transform can be performed into different levels, which corresponding to different time interval length. In this paper, we only discuss single level wavelet transform, which decomposes data into two frequency bands at each dimension.

4.2.3 Double-Wavelet Transform

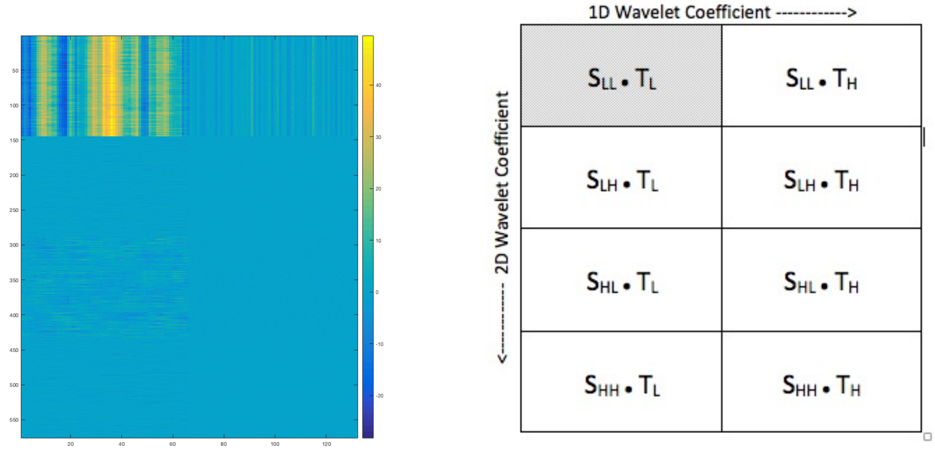
The main idea of the double-wavelet transform is to apply different wavelet functions to different dimensional data in fMRI data analysis. The order of two wavelet transform is interchangeable since wavelet transform is a linear transform. Suppose we have 4-D fMRI data (1-D time series and 3-D volume). First we can obtain 3-D wavelet coefficients by applying 3-D wavelet transform on 3-D volume data at each time point. For each 3-D wavelet transform, we have a time series of it. Second we apply 1-D wavelet transform on the time series of each 3-D wavelet coefficient. We also apply the 1-D wavelet transform on the stimulus function. We will introduce double-wavelet transform in theory step by step with Matlab code.

We first apply the 3-D discrete wavelet transform on the data at each time point in equation (4.1). The 3-D wavelet function we use is $\phi_r(v)$ for ROI c, then we have

$$U_{ncr}(t) = \sum_v Y_{ncv}(t)\phi_r(v) = \sum_{p=1}^{P} \lambda_{ncr}^{p} X^{p}(t) + \varepsilon_{ncv}(t) \text{ where}$$

$r = 1, 2, ..., R$, and R is the total number of 3-D wavelet coefficients at each time point t. $\lambda_{ncr}^{p} = \sum_v \beta_{ncv}^{p}\phi_r(v)$ is the 3-D wavelet coefficient, which are obtained by convolving the 3-D wavelet function and the spatially dependent parameter β_{ncv}^{p} in ROI c for subject n.

Next, for each time series of 3-D wavelet coefficient $U_{ncr}(t)$, we apply the 1-D wavelet transform. The 1-D wavelet function we use is $\varphi_\omega(t)$, then we have



(a) (b)

Figure 4.1: The double-wavelet coefficient structure. (a) is an example using the Daubechies 3 wavelet (spatial wavelet function) and the Symlet 8 wavelet (temporal wavelet function) on an activated 3-D ROI data ($\beta_1 - \beta_2 > 0$) in simulation (2-D in spatial domain and 1-D in temporal domain). (b): S_{LL}, S_{LH}, S_{HL} and S_{HH} represent the wavelet coefficients from the 2-D wavelet transform on the spatial data in low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH) respectively; T_L and T_H represent the wavelet coefficients from the 1-D wavelet transform on the temporal data in low and high frequency band respectively.

$$W_{ncr\omega} = \sum_t U_{ncr}(t)\varphi_\omega(t) = \sum_{p=1}^{P} \lambda_{ncr}^p V_\omega^p + \delta_{ncr\omega} \;\text{ where} \tag{4.3}$$

$\omega = 1, 2, ..., \Omega$, and Ω is the total number of 1-D wavelet coefficients for the time series of each 3-D wavelet coefficient. $V_\omega^p = \sum_t X^p(t)\varphi_\omega(t)$ is the 1-D wavelet coefficients by convolving the 1-D wavelet function with the stimulus function $X^p(t)$. $\delta_{ncr\omega} = \sum_t \varepsilon_{ncr}(t)\varphi_\omega(t)$ is the 1-D wavelet coefficients by convolving with 1-D wavelet function and the time dependent error term $\varepsilon_{ncr}(t)$ at the 3-D wavelet coefficient r in ROI c for subject n. $\delta_{ncr\omega}$ is the double-wavelet coefficient, which is approximately uncorrelated after double-wavelet transformation.

The single level discrete wavelet transform (SL-DWT) decomposes data into two scales on each dimension. For example, a 1-D signal would be transformed into wavelet coefficients indicating information from one high frequency band and one low frequency band.

A 2-D image would be decomposed into high and low frequency bands on both vertical and horizontal directions by SL-DWT, which results into four parts, low-low (horizontal-vertical) frequency band (LL), low-high (horizontal-vertical) frequency band (LH), high-low (horizontal-vertical) frequency band (HL), and high-high (horizontal-vertical) frequency band (HH). For simplicity, we only illustrate 3-D data (1-D time series and 2-D spatial data) in Figure 4.1(b), where S indicates the wavelet coefficients from the 2-D wavelet transform on the spatial data and T indicates the wavelet coefficients from the 1-D wavelet transform on the temporal data.

We found that more than 95% of the wavelet periodograms were contained in the $S_{LL} \cdot T_L$ part for simulated task-induced fMRI data. Hence we only used the double-wavelet coefficients in the $S_{LL} \cdot T_L$ part, where $r = 1, 2, ..., R/4$ and $\omega = 1, 2, ..., \Omega/2$, since other parts of the double wavelet coefficient can be consider as noise in the data.

We can rewrite equation (4.3) in matrix notation. We define $N_q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as multi-variate normal distribution with $q \times 1$ mean vector $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$, and denote the $q \times q$ identity matrix by $\boldsymbol{I_q}$. Then we have

$$\boldsymbol{W}_{ncr} \quad = \quad \boldsymbol{V}^T \boldsymbol{\lambda}_{ncr} + \boldsymbol{\delta}_{ncr}, \quad \text{where} \tag{4.4}$$

- $\boldsymbol{W}_{ncr} = [W_{ncr1}, W_{ncr2}, ..., W_{ncr(\Omega/2)}]^T$ is a $\Omega/2 \times 1$ vector, which are the double-wavelet coefficients in $S_{LL} \cdot T_L$ part by performing double-wavelet transform on data Y_{ncv} for subject n in ROI c. We assume that $\boldsymbol{W}_{ncr} \sim N_{\Omega/2}(\boldsymbol{V}^T \boldsymbol{\lambda}_{ncr}, cov(\boldsymbol{\delta}_{ncr}))$

- $$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V^1} \\ \boldsymbol{V^2} \\ .. \\ \boldsymbol{V^P} \end{bmatrix} = \begin{bmatrix} V_1^1 & V_2^1 & V_3^1 & \cdots & V_{\Omega/2}^1 \\ V_1^2 & V_2^2 & V_3^2 & \cdots & V_{\Omega/2}^2 \\ & & \cdots\cdots\cdots & & \\ V_1^P & V_2^P & V_3^P & \cdots & V_{\Omega/2}^P \end{bmatrix}_{P \times \Omega/2}$$

  where $\boldsymbol{V}$ is a $P \times \Omega/2$ matrix, in which elements are the 1-D wavelet coefficients by performing 1-D wavelet transform on $X^p(t)$ corresponding the p^{th} stimulus function;

- $\boldsymbol{\lambda}_{ncr} = [\lambda_{ncr}^1, \lambda_{ncr}^2, ..., \lambda_{ncr}^P]^T$ is a $P \times 1$ vector, consisting of the 2-D wavelet coefficients by performing 2-D wavelet transform on the β_{ncv}^p;
- $\boldsymbol{\delta}_{ncr} = [\delta_{ncr1}, \delta_{ncr2}, ..., \delta_{ncr(\Omega/2)}]^T$ is a $\Omega/2 \times 1$ vector, consisting of the 1-D wavelet coefficients by performing 1-D wavelet transform on the error term $\boldsymbol{\epsilon}_{ncv}$, and $\boldsymbol{\delta}_{ncr} \sim N_{\Omega/2}(\underline{0}, \boldsymbol{\sigma}^2 \boldsymbol{I}_{\Omega/2})$, where $\boldsymbol{\sigma}^2$ is the variance of the wavelet coefficients $\boldsymbol{\delta}_{ncr}$.

Then we estimate $\hat{\boldsymbol{\lambda}}_{ncr}$ by using the ordinary least squares estimator

$$\hat{\boldsymbol{\lambda}}_{ncr} = (\boldsymbol{V}\boldsymbol{V}^T)^{-1}\boldsymbol{V}\boldsymbol{W}_{ncr}$$

The boxcar stimuli are orthogonal to each other in task-induced fMRI data, which means $\boldsymbol{X}^{p'} \cdot (\boldsymbol{X}^{\boldsymbol{p}})^T = 0$ when $p \neq p'$, then

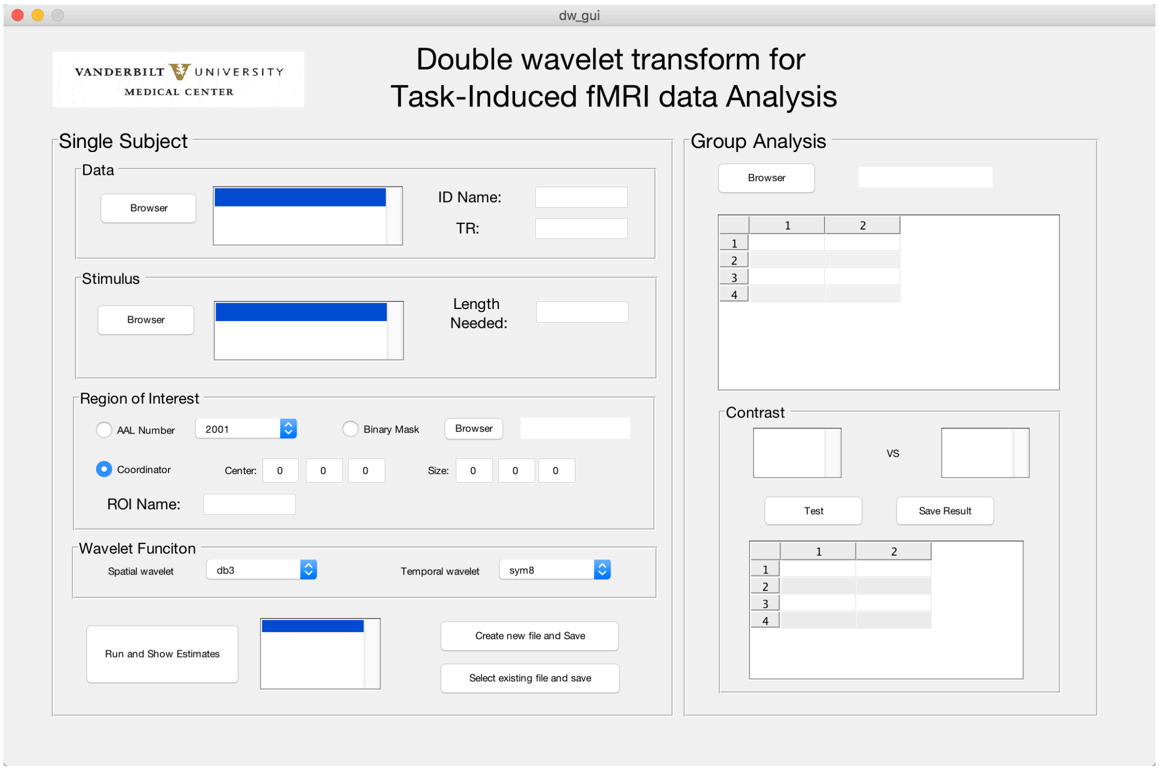
$$\boldsymbol{V}^{\boldsymbol{p}'} \cdot (\boldsymbol{V}^{\boldsymbol{p}})^T = (\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{\boldsymbol{p}'})(\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{\boldsymbol{p}})^T = \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{X}^{\boldsymbol{p}'}(\boldsymbol{X}^{\boldsymbol{p}})^T\boldsymbol{\phi}(\boldsymbol{x})^T$$

where $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(x) \quad \phi_2(x) \quad ... \quad \phi_{\Omega/2}(x)]$ and $\phi_k(x), k = 1, 2, ..., \Omega/2$ are wavelet functions. Then we have $\boldsymbol{V}^{\boldsymbol{p}'} \cdot (\boldsymbol{V}^{\boldsymbol{p}})^T = 0$ and $(\boldsymbol{V}\boldsymbol{V}^T)^{-1}$ is a diagonal matrix.

A simple t-test is used on a linear contrast of $\boldsymbol{\lambda}_{ncr}$ based on multi-subject data. Since there is a one to one relationship between β_{ncv}^p and λ_{ncr}^p, the hypothesis in equation (4.2) is equivalent to

$$H_0 : \lambda_c^p - \lambda_c^1 = 0$$

where λ_c^1 is the mean of the estimator λ_{ncr}^1 and λ_c^p is the mean of the estimator λ_{ncr}^p across subjects for ROI c using double-wavelet coefficients in $S_{LL} \cdot T_L$ part.



Double wavelet transform for
Task-Induced fMRI data Analysis

○ AAL Number 2001
● Coordinator Center: 0 0 0 Size: 0 0 0

Spatial wavelet db3
Temporal wavelet sym8

Figure 4.2: The interface of Matlab GUI

4.3 Graphical User Interface Design

4.3.1 GUI Design Process

This Matlab GUI for multi-subject task-induced fMRI data using double-wavelet transform is developed in Matlab (Mathwork, Inc) as an open source software. It is designed to analyze single subject fMRI data using double-wavelet approach, and then perform group analysis using T-test for different contrast in stimulus. The interface is shown in Fig 4.2.

The GUI includes two main parts. The left part is for signal subject analysis. There are five chunks in this part, which will guide the user step by step to perform single subject analysis. In the “Data” chunk, the user can browse in the computer and select multiple sessions’ data for single subject. The format of data is required to be in NIFTI format. The ID name and repetition time (TR) need to be entered here.

In the “Stimulus” chunk, the user can browse and select multiple sessions’ stimuli for the data in the “Data” chunk. The stimulus is a box-car function at each time point. Different stimulus needs to be put in different column without header and save as csv (comma separated values) file. The user also need to input the length of time series needed.

In the “Region of Interest” chunk, the user can specify the region of interest in three ways, using prespecified AAL (Automated Anatomical Labeling) number, user’s own binary mask or cube centered at a certain coordinator.

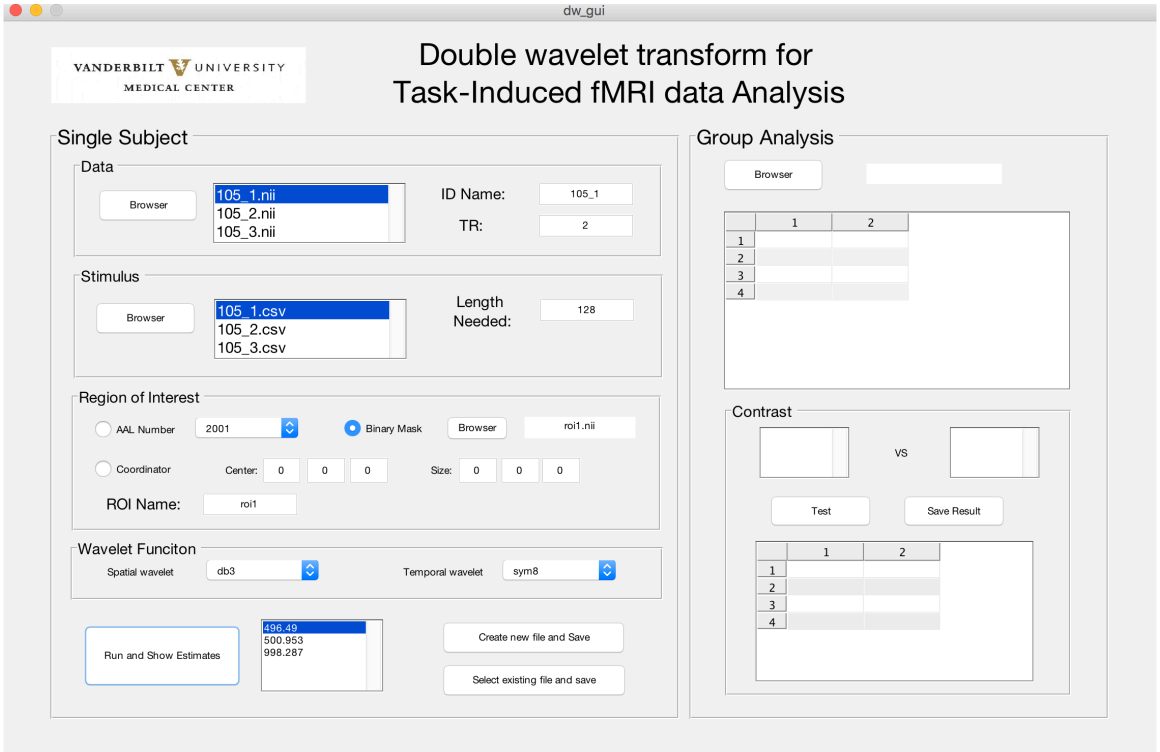
In the “Wavelet Function” chunk, the user can choose different wavelet functions for volume data and time series data. In the last chunk, the user can run the single subject analysis given input data, create a new csv file and save the result. The user can also save result to a current csv file, where multiple subjects’ result can be saved in one file for later use.

The group analysis function is on the right side of the GUI, There are two chunks in this part. In the first chunk, the user need to load the csv file containing single subjects’ analysis. The content in the file will be shown in the table. In the “Contrast” chunk, the user can choose different contrast to test. A student t-test will be performed here. The test result is shown below and can be saved in to a csv file. We will use an example to illustrate all the steps in more detail.

4.3.2 Illustrative examples

We provide sample data for users to download. The sample data include three subjects’ data, stimulus functions, and one binary mask file. We will use these files to illustrate how to use the GUI.

The sample data were from a study designed to test cognitive control related activation in the prefrontal cortex (PFC) of the human brain. The binary mask is from the anterior premotor cortex (prePMd). During the study, each participant would be given two stimulus (D1 and D2). Based on prior studies (Badre & D’Esposito, 2007; Badre et al., 2009),



dw_gui

VANDERBILT UNIVERSITY MEDICAL CENTER

Double wavelet transform for
Task-Induced fMRI data Analysis

Single Subject

Data

105_1.nii
105_2.nii
105_3.nii

ID Name: 105_1
TR: 2

Stimulus

105_1.csv
105_2.csv
105_3.csv

Length Needed: 128

Region of Interest

○ AAL Number 2001 ● Binary Mask roi1.nii

○ Coordinator Center: 0 0 0 Size: 0 0 0

ROI Name: roi1

Wavelet Funciton

Spatial wavelet db3 Temporal wavelet sym8

496.49
500.953
998.287

Group Analysis

	1	2
1		
2		
3		
4		

Contrast

VS

	1	2
1		
2		
3		
4		

Figure 4.3: Illustrating the setting and result for subject 105

prePMd should be activated with the contrast of $D2 > D1$. More details about the study can be found in Kang et al. (2012).

In “Data” section, we select “105_1.nii”, “105_2.nii” and “105_3.nii” using browser. The ID Name will be set to default as the first file name “105_1”. Users can change the ID name. The repetition time (TR) is 2 seconds in this study. In “Stimulus” section, we select “105_1.csv”, “105_2.csv” and “105_3.csv” using browser. We can set the length of time series that we want to use. Here we set the length as 128. In “Region of Interest” section, we select “Binary Mask”, and select “roi1.nii” using browser. In the “Wavelet Function” section, we use the default setting: “db3” as spatial wavelet and “sym8” as temporal wavelet.

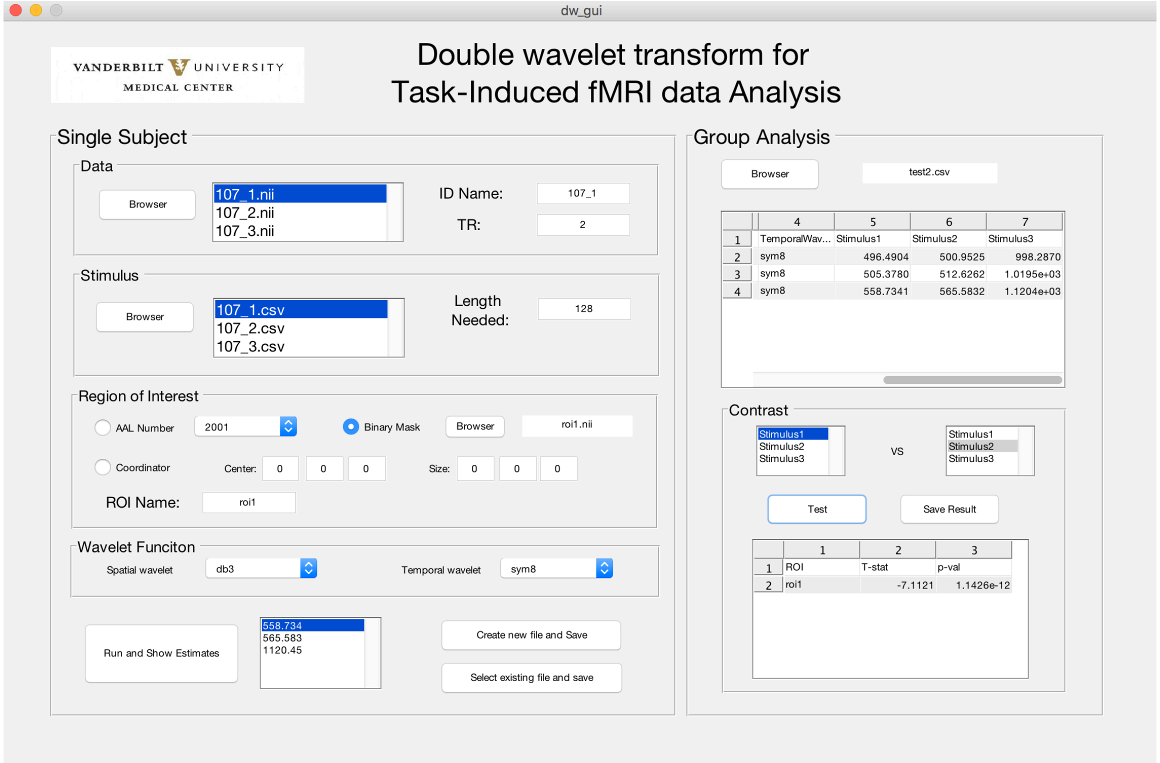
After selecting all files and settings, we click “Run and Show Estimate”, the result of subject 105 will be calculated and shown in the window as in Figure 4.3. Since there are three stimulus in the stimulus files, three double-wavelet coefficients corresponding to each stimulus were shown in the box. Then we click “Create new file and Save” to save the result of subject 105. The subject ID, ROI name, and estimate of each stimulus function will be saved to a csv file. We save it as “test.csv”.

Repeating the step above for subject 106 and subject 107. The only difference would be the last step. Rather than clicking “Create new file and Save”, we need to click “Selecting existing file and Save”, where you can save the result in the “test.csv”, the file we created for subject 105. Then the estimates for all subjects will be saved in to one file and ready for group analysis.

To perform group analysis, we click the “Browser” in the “Group Analysis” section on the right hand side of the GUI. We then select the file “test.csv”, which contains the result for three subjects. The file name will be displayed next to the “Browser” button, and the content of the file will be shown in the box below the “Browser” button. Since we are interested in the contrast between first two stimulus functions, we select “Stimulus1” and “Stimulus2” in the “Contrast” Section, and then click “Test”. The T statistics and p-value of the test will be calculated and shown in the box as in Figure 4.4.

4.4 Summary

In this paper, we presented the MATLAB GUI for multi-subject task-induced fMRI data using double-wavelet transform. The main advantage of using double-wavelet transform is that it takes account both spatial and temporal correlation, which reduce type I and type II errors. This GUI provides features from importing data and stimulus functions, specifying region of interest, and performing analysis from single subject analysis to group analysis. We believe that this GUI will make double-wavelet transform to a wider range of researchers.



Double wavelet transform for
Task-Induced fMRI data Analysis

Single Subject

Data

107_1.nii
107_2.nii
107_3.nii

ID Name: 107_1

TR: 2

Stimulus

107_1.csv
107_2.csv
107_3.csv

Length Needed: 128

Region of Interest

○ AAL Number 2001

● Binary Mask roi1.nii

○ Coordinator Center: 0 0 0 Size: 0 0 0

ROI Name: roi1

Wavelet Funciton

Spatial wavelet db3

Temporal wavelet sym8

558.734
565.583
1120.45

Group Analysis

test2.csv

	4	5	6	7
1	TemporalWav...	Stimulus1	Stimulus2	Stimulus3
2	sym8	496.4904	500.9525	998.2870
3	sym8	505.3780	512.6262	1.0195e+03
4	sym8	558.7341	565.5832	1.1204e+03

Contrast

Stimulus1
Stimulus2
Stimulus3

VS

Stimulus1
Stimulus2
Stimulus3

	1	2	3
1	ROI	T-stat	p-val
2	roi1	-7.1121	1.1426e-12

Figure 4.4: Illustrating the setting and result group analysis.

Appendix C

Main Matlab Code for GUI

```
%  double wavelet use level 1
N.level = 1;
% extract the size of 4d data
 Sdata = size(data);
% 3D size
dim = [Sdata(1) Sdata(2) Sdata(3)];
% Time length
T = Sdata(4);
% get the size of 3d wavelet coefficient
WT = wavedec3(zeros(dim(1), dim(2), dim(3)), N.level ,wave3d
    );
% extract the low−low part
sWT = size(WT.dec{1});
% number of 2d wavelet coefficient
 coef3d = prod(sWT);
% 1D wavelet coefficient
 [TC,TS] = wavedec(zeros(1,T), N.level ,wave1d);
% length 1d wavelet transform
Ntime = length(TC);
% number of 1d low frequency wavelet coefficient
 waveband = 1:(Ntime/2) ;
% size of double wavelet transform data
```

```
DWdata = zeros(coef3d, Ntime/2);
% create 3d wavelet coefficient matrix
waveC = zeros( T, coef3d);

% wavelet transform on each 3D data at each time point
for t = 1:T
    data_waiting = data(:,:,:,t);
    % reshape data into 3d
    subjbrain = reshape(data_waiting, dim(1), dim(2),dim(3))
        ;
    % 3d Wavelet transform
    WT_temp = wavedec3(subjbrain, N.level ,wave3d);
    % reshape transformed data into 2d
    waveC(t,:) = reshape(WT_temp.dec{1},coef3d,1 );
end

% wavelet transform on timeseries of each 3d wavelet
    coefficient
for coef = 1:coef3d
    % reshape to 1d time series
    subjT = reshape(waveC(:,coef),1,T);
    % 1d wavelet transform
    [waveT(coef, :),TS] = wavedec(subjT,N.level,wave1d);
end
% extract low-low low data
DWdata = waveT(:, waveband);
```

Chapter 5

CONCLUSION

5.1 Summary

In this dissertation, we proposed a double-wavelet transform approach, where we transformed the data twice using different wavelet functions. We applied this approach to task-induced fMRI data analysis in Chapter 2, and to resting state fMRI data analysis in Chapter 3. The advantages of our wavelet model over existing models are as follows. First, we took into account the spatial-temporal correlation in fMRI data by transforming the data twice. Second, stationary assumptions on both spatial and temporal data were not required using the double-wavelet transform since the wavelet transform handled it naturally. Third, our approach did not require the estimation and inversion of the covariance matrix because wavelet coefficients were approximately uncorrelated, which significantly reduced the computational burden in the fMRI data analysis. Fourth, the double-wavelet transform converted 4-D data into 2-D data, which simplified the fMRI data structure. In Chapter 4, we presented the MATLAB GUI for multi-subject task-induced fMRI data using double-wavelet transform.

5.2 Future Research

Though double-wavelet transform has many advantage over existing model, there is much work to be done in the future. Here we list some directions.

First, more simulation and research need to be done on the subjective choice of the wavelet functions. This can be partially solved by using data decimation method to choose wavelet functions. Second, the wavelet transform naturally decomposes data into different scales. The single level discrete wavelet transform (SL-DWT) decomposes data into two frequency bands at each dimension. There are more ways to combine these information in different frequency bands. Third, we only investigate single level discrete wavelet transform (SL-DWT). More levels can be used to further improve the signal-to-noise ratio and model performance. Fourth, the MATLAB GUI can be attached to SPM to make it more accessible to researchers. Fifth, R shinyapp and package for double wavelet transform can be developed for R users.

REFERENCES

Aertsen, A., Gerstein, G., Habib, M. & Palm, G. (1989), Dynamics of neuronal firing correlation: modulation of effective connectivity, *J. Neurophysiol* **61**(5), 900 – 917.

Aston, J., Gunn, R., Hinz, R. & Turkheimerc, F. (2005), Wavelet variance components in image space for spatiotemporal neuroimaging data, *NeuroImage* **25**, 159 – 168.

Badre, D. (2008), Cognitive control, hierarchy, and the rostro-caudal axis of the prefrontal cortex, *Trends in Cognitive Science* **12**, 193–200.

Badre, D. & D'Esposito, M. (2007), Fmri evidence for a hierarchical organization of the prefrontal cortex, *Journal of Cognitive Neuroscience* **19**, 2082–2099.

Badre, D., Hoffman, J., Cooney, J. & D'Esposito, M. (2009), Hierarchical cognitive control deficits following damage to the human frontal lobe, *Nature Neuroscience* **12**, 515–522.

Biswal, B., Kylen, J. & Hyde, J. (1997), Simultaneous assessment of flow and bold signals in resting-state functional connectivity maps, *NMR Biomed* **10**(4-5), 165 – 170.

Biswal, B., Yetkin, F., Haughton, V. & Hyde, J. (1995), Functional connectivity in the motor cortex of resting human brain using echo-planar mri, *Magnetic Resonance in Medicine* **34**(4), 537 – 541.

Brammer, M. (1998), Multidimensional wavelet analysis of functional magnetic resonance images, *Human Brain Mapping* (6), 378–382.

Bullmore, E., Fadili, J., Breakspear, M., Salvador, R., Suckling, J. & Brammer, M. (2003), Wavelets and statistical analysis of functional magnetic resonance images of the human brain, *Statistical Methods in Medical Research* (12), 375–399.

Chiann, C. & Morettin, P. (1998), A wavelet analysis for time series, *Journal of Nonparametric Statistics* **10**(1), 1 – 46.

Cordes, D., Haughton, V., Arfanakis, K., Carew, J., Turski, P., Moritz, C., Quigley, M. & Meyerand, M. (2001), Frequencies contributing to functional connectivity in the cerebral cortex in resting-state data, *AJNR Am. J. Neuroradiol* **22**, 1326 – 1333.

Cordes, D., Haughton, V., Arfanakis, K., Wendt, G., Turski, P., Moritz, C., Quigley, M. & Meyerand, M. (2000), Mapping functionally related regions of brain with functional connectivity mr imaging, *AJNR Am. J. Neuroradiol* **21**, 1636 – 1644.

Degras, D. & Lindquist, M. (2014), A hierarchical model for simultaneous detection and estimation in multi-subject fmri studies, *NeuroImage* **98**, 61 – 72.

den Heuvel, M. V., Mandl, R. & Hulshoff Pol, H. (2008), Normalized group clustering of resting-state fmri data, *PLoS ONE* (e2001).

Dubin, R. (1988), Estimation of regression coefficients in the presence of spatially autocorrelated error terms, *Review of Economics and Statistics* **70**, 466–474.

Eryilmaz, H., Ville, D., Schwartz, S. & Vuilleumier, P. (2011), Impact of transient emotions on functional connectivity during subsequent resting state: A wavelet correlation approach, *NeuroImage* **54**, 2841 – 2491.

Fan, Y. (2003), On the approximate decorrelation property of the discrete wavelet transform for fractionally differenced processes, *IEEE Transactions on Information Theory* **49**(2), 516–521.

Fox, M., Maurizio, C., Abraham, S., Justin, V., & Marcus, R. (2006), Spontaneous neuronal activity distinguishes human dorsal and ventral attention systems, *Proc Natl Acad Sci* **103**, 10046 – 10051.

Frackowiak, R., Friston, K., Frith, C., Dolan, R. & Mazziotta, J. (1997), *Human Brain Function*, Academic Press USA.

Friston, K., Frith, C., Liddle, P. & Frackowiak, R. (1993), Functional connectivity: the principal-component analysis of large (pet) data sets, *J. Cereb. Blood Flow Metab* **13**(1), 5 – 14.

Huettel, S., Song, A. & McCarthy, G. (2004), *Functional Magnetic Resonance Imaging*, Sinauer Associates, Sunderland, MA, chapter 10.

Hyun, J. W., Li, Y., Gilmore, J., Lu, Z., Styner, M. & Zhu, H. (2014), Sgpp: spatial gaussian predictive process models for neuroimaging data, *NeuroImage* **89**, 70 – 80.

Hyun, J.-W., Li, Y., Huang, C., Styner, M., Lin, W. & Zhu, H. (2016), Stgp: Spatio-temporal gaussian process models for longitudinal neuroimaging data, *NeuroImage* **134**, 550 – 562.

Kang, H., Ombao, H., Linkletter, C., Long, N. & Badre, D. (2012), Spatio-spectral mixed effects model for functional magnetic resonance imaging data, *Journal of the American Statistical Association* **107**, 568–577.

Kang, J., Bowman, F., Mayberg, H. & Liu, H. (2016), A depression network of functionally connected regions discovered via multi-attribute canonical correlation graphs, *NeuroImage* **141**, 431 – 441.

Karaman, M., Nencka, A., Bruce, P. & Rowe, D. (2014), Quantification of the statistical effects of spatiotemporal processing of nontask fmri data, *Brain Connectivity* **4**(9), 649 – 661.

Katanoda, K., Matsuda, Y. & Sugishita, M. (2002), A spatio-temporal regression model for the analysis of functional mri data, *NeuroImage* **17**, 1415–1428.

Kiviniemi, V., Kantola, J., Jauhiainen, J. & Tervonen, O. (2004), Comparison of methods for detecting nondeterministic bold fluctuation in fmri, *Magn. Reson. Imaging* **22**.

Koechlin, E., Ody, C. & Kouneiher, F. (2003), The architecture of cognitive control in the human prefrontal cortex, *Science* **302**, 1181–1185.

Lindquist, M. A., Loh, J., Atlas, Y. & Wager, T. (2009), Modeling the hemodynamic response function in fmri: Efficiency, bias and mis-modeling, *NeuroImage* **45**, S187 – S198.

Long, C., Brown, E., Manoach, D. & Soloa, V. (2004), Spatiotemporal wavelet resampling for functional neuroimaging data, *NeuroImage* (23), 500–516.

Long, N. & Badre, D. (2009), 'Testing hierarchical interactions in frontal cortex during cognitive control', Poster presented at the 16th Cognitive Neuroscience Society meeting.

Lowe, M., Mock, B. & Sorenson, J. (1998), Functional connectivity in single and multislice echoplanar imaging using resting-state fluctuations, *NeuroImage* **7**, 119 – 132.

Luca, M. D., Smith, S., Stefano, N., Federico, A. & Matthews, P. (2005), Blood oxygenation level dependent contrast resting state networks are relevant to functional activity in the neocortical sensorimotor system, *Exp. Brain Res* **167**(4), 587 – 594.

Meddaa, A., Hoffmannb, L., Magnusonc, M., Thompsonc, G., Pan, W. & Keilholzc, S. (2011), Wavelet-based clustering of resting state mri data in the rat, *Magnetic Resonance Imaging* **34**(1), 35 – 43.

Nason, G. (2008), *Wavelet Methods in Statistics with R*, Springer.

Ombao, H., Shao, X., Rykhlevskaia, E., Fabiani, M. & Gratton, G. (2008), Spatio-spectral analysis of brain signals, *Statistica Sinica* **18**, 1465–1482.

Patel, A., Kundu, P., Rubinov, M., Jones, S., Vrtes, P., Ersche, K., Suckling, J. & Bullmore, E. (2014), A wavelet method for modeling and despiking motion artifacts from resting-state fmri time series, *NeuroImage* **95**, 287 – 304.

Raichle, M., MacLeod, A., Snyder, A., Powers, W., Gusnard, D. & Shulman, G. (2001), A default mode of brain function., *Proc Natl Acad Sci* **98**(2), 676 – 682.

Ruttimann, U., Unser, M., Rawlings, R., Rio, D., Ramsey, N., Mattay, V., Hommer, D., Frank, J. & Weinberger, D. (1998), Statistical analysis of functional mri data in the wavelet domain, *IEEE Transactions on Medical Imaging* **17**(2), 142–154.

Saleh, A., Potter, G. G., McQuoid, D. R., Boyd, B., Turner, R., MacFall, J. R. & Taylor, W. D. (2017), Effects of early life stress on depression, cognitive performance and brain morphology, *Psychological Medicine* **47**(1), 171 – 181.

Scargle, J., Steinman-Cameron, T., Young, K., Donoho, D. & Crutch-field, J. (1993), *Imamura J.* **411**, L91.

Taylor, W., Boyd, B., Turner, R., McQuoid, D., Ashley-Koch, A., MacFall, J., Saleh, A. & Potter, G. (2016), Apoe 4 associated with preserved executive function performance and maintenance of temporal and cingulate brain volumes in younger adults, *Brain Imaging and Behavior* 1 – 11.

Vidakovic, B. (1999), *Statistical Modeling by Wavelets*, Wiley.

Ville, D., Blu, T. & Unser, M. (2004), Integrated wavelet processing and spatial statistical testing of fmri data, *NeuroImage* (23), 1472–1485.

Vincent, J., Kahn, I., Snyder, A., Raichle, M. & Buckner, R. (2008), Evidence for a frontoparietal control system revealed by intrinsic functional connectivity, *J Neurophysiol* **100**(6), 3328 – 3342.

Weiskopf, N., Veit, R., Erb, M., Mathiak, K., Grodd, W., Goebel, R. & Birbaumera, N. (2003), Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fmri): methodology and exemplary data, *NeuroImage* **19**, 577 – 586.

Worsley, K. & Friston, K. (1995), Analysis of fmri time-series revisited again, *NeuroImage* **2**, 173 – 181.

Worsley, K., Marrett, S., Neelin, P., Vandal, A., Friston, K. & Evans, A. (1996), A unified statistical approach for detecting significant signals in images of cerebral activation, *Human Brain Mapping* **4**, 58–73.

Yang, X., Kang, H., Newton, A. & Landman, B. (2010), Evaluation of statistical inference on empirical resting state fmri, *IEEE Transacion On Biomedical Engineering* **61**(4), 1091 – 1099.

Zhang, T., Li, F., Beckes, L., Brown, C. & Coan, J. (2012), Nonparametric inference of hemodynamic response for multi-subject fmri data, *NeuroImage* **63**, 1754 – 1765.

Zhang, T., Li, F., Beckes, L. & Coan, J. (2013), A semi-parametric model of the hemodynamic response for multi-subject fmri data, *NeuroImage* **75**, 136 – 145.

Zhang, T., Li, F., Gonzalez, M., Maresh, E. & Coan, J. (2014), A semi-parametric nonlinear model for event-related fmri, *NeuroImage* **97**, 178 – 187.