Matlab Tool: Functional Analysis of Diffusion Tensor Tract Statistics

1 Introduction

1.1 FADTTS summary

The aim of this tool is to implement a functional analysis pipeline, called FADTTS, for delineating the structure of the variability of multiple diffusion properties along major white matter fiber bundles and their association with a set of covariates of interest, such as age, diagnostic status and gender, in various diffusion tensor imaging studies. The FADTTS integrates five statistical tools: a multivariate varying coefficient model for allowing the varying coefficient functions to characterize the varying association between fiber bundles diffusion properties and a set of covariates, a weighted least squares estimation to estimate the varying coefficient functions, a functional principal component analysis to delineate the structure of the variability in fiber bundles diffusion properties, a global test statistic to test hypotheses of interest, and a simultaneous confidence band to quantify the uncertainty in the estimated coefficient function. FADTTS can be used to facilitate understanding normal brain development, the neural bases of neuropsychiatric disorders, and the joint effects of environmental and genetic factors on white matter fiber bundles.

1.2 Motivation

Diffusion Tensor Imaging (DTI), which can track the effective diffusion of water in the human brain in vivo, has been widely used to map the structure and orientation of the white matter fiber tracts of the brain (Basser et al., 1994b,a). In the current literature, three major approaches to the group analysis of diffusion imaging data are region-of-interest (ROI) analysis, voxel based analysis, and fiber tract based analysis (Smith et al., 2006; O'Donnell et al., 2009; Snook et al., 2007). The ROI analysis used in some neuroimaging studies (Bonekam et al., 2008; Gilmore et al., 2008) primarily suffers from the difficulty in identifying meaningful ROIs. Voxel based analysis is used more commonly than ROI analysis in neuroimaging studies (Chen et al., 2009; Focke et al., 2008; Camara et al., 2007; Snook et al., 2005). The major drawbacks of voxel based analysis include the issues of alignment quality and the arbitrary choice of smoothing extent (Hecke et al., 2009; Ashburner and Friston, 2000; Smith et al., 2006; Jones et al., 2005). With the drawbacks mentioned of the ROI and voxel based analyses, there is a growing interest in the DTI literature in developing fiber tract based analysis of diffusion properties (Smith et al., 2006; O'Donnell et al., 2009; Yushkevich et al., 2008; Goodlett et al., 2009; Zhu et al., 2010b). Statistically, diffusion properties along fiber bundles are functional data and its analysis requires advanced functional data analysis methods (Li and Hsing, 2010; Yao and Lee, 2006; Hall et al., 2006; Ramsay and Silverman, 2005, 2002).

There are several developments on the use of functional data analysis methods for the statistical analysis of diffusion properties along fiber tracts, which all are "smoothing first, then estimation" procedures. However, their methods are not capable of delineating the structure of the variability in fiber bundles diffusion properties and for quantifying the uncertainty in the estimated coefficient functions. To specifically address the limitations in (Goodlett et al., 2009; Zhu et al., 2010b), FADTTS presents a functional analysis pipeline for delineating the structure of the variability of multiple diffusion properties along major white matter fiber bundles and their association with a set of covariates of interest, such as age, diagnostic status and gender, in various diffusion tensor imaging studies.

1.3 FADTTS description

Compared with (Goodlett et al., 2009; Zhu et al., 2010b) and other existing literature, there are five methodological contributions in FADTTS: first, a multivariate varying coefficient model, second, a weighted least squares estimation, third, a functional principal component analysis, fourth, a global test statistic based on a resampling method and fifth, a simultaneous confidence band based on a resampling method. A schematic overview of FADTTS is given in Fig 1. We describe each of these components briefly below. Detailed description and related theorem proofs can be found in Zhu et al. (2010a).

1. Multivariate Varying Coefficient Model

Let $s \in [0, L]$ be the arc length of any point on a specific fiber bundle relative to a fixed end point of the fiber bundle, where L is the longest arc length on the fiber bundle. For the i-th subject, we consider an $J \times 1$ vector of diffusion properties, denoted by $\mathbf{y}_i(s_m) = (y_{i,1}(s_m), \cdots, y_{i,J}(s_m))^T$, and its associated arc length s_m for the m-th location grid point on the fiber bundle for $m = 1, \cdots, M$ and $i = 1, \cdots, n$, where M and n denote the numbers of grid points and subjects, respectively. We consider a multivariate varying coefficient model (Fan and Zhang, 1999; Wu and Chiang, 2000; Fan et al., 2003; Fan and Zhang, 2008; Wang et al., 2008) , which assumes that for $k = 1, \cdots, m$ and $i = 1, \cdots, n$,

$$y_{i,j}(s) = \mathbf{x}_i^T B_j(s) + \eta_{i,j}(s) + \epsilon_{i,j}(s) = \mathbf{x}_i^T B_j(s) + \sum_{l=1}^{\infty} \xi_{ij,l}\psi_{j,l}(s) + \epsilon_{i,j}(s), \tag{1}$$

where $B_j(s) = (\beta_{j1}(s), \cdots, \beta_{jp}(s))^T$ is a $p \times 1$ vector of coefficient functions of s and $\mathbf{x}_i$ is a $p \times 1$ vector of covariates of interest with $x_{i,1} = 1$, $\epsilon_{i,j}(s)$ are measurement errors. Moreover, $\eta_{i,j}(s) = \sum_{l=1}^{\infty} \xi_{ij,l}\psi_{j,l}(s)$ characterize individual curve variations from $\mathbf{x}_i^T B_j(s)$

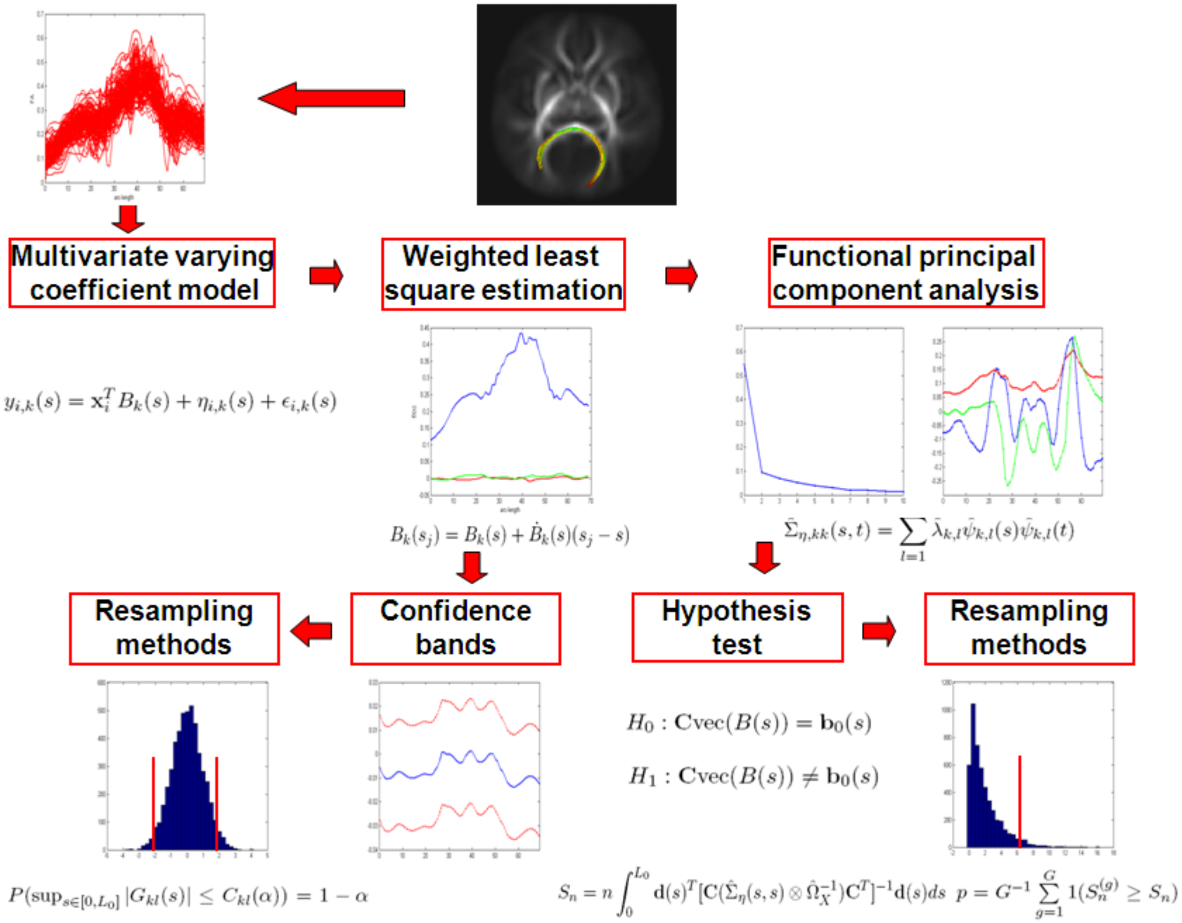
Multivariate varying coefficient model
Weighted least square estimation
Functional principal component analysis
$y_{i,k}(s) = \mathbf{x}_i^T B_k(s) + \eta_{i,k}(s) + \epsilon_{i,k}(s)$
$B_k(s_j) = B_k(s) + \dot{B}_k(s)(s_j - s)$
$\hat{\Sigma}_{\eta,kk}(s,t) = \sum_{l=1} \hat{\lambda}_{k,l}\hat{\psi}_{k,l}(s)\hat{\psi}_{k,l}(t)$
Resampling methods
Confidence bands
Hypothesis test
Resampling methods
$H_0 : \mathbf{C}\mathrm{vec}(B(s)) = \mathbf{b}_0(s)$
$H_1 : \mathbf{C}\mathrm{vec}(B(s)) \neq \mathbf{b}_0(s)$
$P(\sup_{s\in[0,L_0]} |G_{kl}(s)| \leq C_{kl}(\alpha)) = 1 - \alpha$
$S_n = n\int_0^{L_0} \mathbf{d}(s)^T[\mathbf{C}(\hat{\Sigma}_\eta(s,s)\otimes\hat{\Omega}_X^{-1})\mathbf{C}^T]^{-1}\mathbf{d}(s)ds$
$p = G^{-1}\sum_{g=1}^{G} 1(S_n^{(g)} \geq S_n)$

Figure 1: A schematic overview of FADTTS: a functional multivariate varying coefficient model for the diffusion properties of a tract, a weighted least square estimation method for estimating the coefficient functions, a functional principal component analysis model for analyzing the covariance structure, a hypothesis test for coefficient functions using both local and global test statistics, a resampling method for estimating the p-value of the global test statistics, a method of calculating the confidence bands of the coefficient functions based on a resampling method.

and $\xi_{ij,l}$ are uncorrelated random variables with $E(\xi_{ij,l}) = 0$ and $E(\xi_{ij,l}^2) = \lambda_{j,l}$ such that $\lambda_{j,1} \geq \lambda_{j,2} \geq \cdots \geq 0$ with $\sum_{l=1}^{\infty} \lambda_{j,l} < \infty$ and $\int_0^L \psi_{j,l}(t)\psi_{j,l'}(t)dt = \mathbf{1}(l = l')$, where $\mathbf{1}(\cdot)$ is an indicator function. There could be also some constraints on $\boldsymbol{\epsilon}_i(s) = (\epsilon_{i,1}(s), \cdots, \epsilon_{i,J}(s))^T$ and $\boldsymbol{\eta}_i(s) = (\eta_{i,1}(s), \cdots, \eta_{i,J}(s))^T$; see (Zhu et al., 2010a).

2. Weighted Least Squares Estimation

To estimate the coefficient functions in $\mathrm{B}(s) = [B_1(s), \cdots, B_J(s)]$, we develop a weighted least squares estimation method based on an adaptive local polynomial kernel (LPK) smoothing technique (Fan and Gijbels, 1996; Wand and Jones, 1995; Wu and Zhang, 2006; Ramsay and Silverman, 2005; Welsh and Yee, 2006; Zhang and Chen, 2007). Specifically, using Taylor's expansion, we can expand $B_j(s_m)$ at s to obtain

$$B_j(s_m) = B_k(s) + \dot{B}_j(s)(s_m - s) = A_j(h, s)\mathbf{z}(h, s_m - s), \tag{2}$$

where $\mathbf{z}(h, s_m - s) = (1, (s_m - s)/h)^T$ and $A_j(s) = [B_j(s), h\dot{B}_j(s)]$ is a $p \times 2$ matrix, in which $\dot{B}_j(s) = (\dot{\beta}_{j1}(s), \cdots, \dot{\beta}_{jp}(s))^T$ is a $p \times 1$ vector and $\dot{\beta}_{jl}(s) = d\beta_{jl}(s)/ds$ for $l = 1, \cdots, p$. We calculate a weighted least squares estimate of $A_j(s)$ as follows. Let $K(\cdot)$ be a kernel function, such as the Gaussian and uniform kernels (Fan and Gijbels, 1996; Wand and Jones, 1995). For a fixed bandwidth h and each j, we estimate $A_j(s)$ by minimizing an objective function given by

$$\sum_{i=1}^{n}\sum_{m=1}^{M}[y_{i,j}(s_m) - \mathbf{x}_i^T A_j(h, s)\mathbf{z}(h, s_m - s)]^2 K(h, s_m - s), \tag{3}$$

where $K(h, \cdot) = K(\cdot/h)/h$ is a rescaled kernel function. For each j, we pool the data from all n subjects and select an optimal bandwidth $h_j^{(1)}$, denoted by $\hat{h}_j^{(1)}$, and we can obtain $\hat{B}_j(s)$. Combing all $\hat{B}_j(s)$ leads to $\hat{\mathrm{B}}(s) = [\hat{B}_1(s), \cdots, \hat{B}_J(s)]$.

3. Functional Principal Component Analysis

To simultaneously construct all individual functions $\eta_{i,k}(s)$, we also employ the local polynomial kernel smoothing technique (Fan and Gijbels, 1996; Wand and Jones, 1995; Wu and Zhang, 2006; Ramsay and Silverman, 2005; Welsh and Yee, 2006; Zhang and Chen, 2007). Specifically, using Taylor's expansion, we can expand $\eta_{i,j}(s_m)$ at s to obtain

$$\eta_{i,j}(s_m) = \mathbf{d}_{i,j}(s)^T\mathbf{z}(h, s_j - s), \tag{4}$$

where $\mathbf{d}_{i,j}(s) = (\eta_{i,j}(s), h\dot{\eta}_{i,j}(s))^T$ is a 2×1 vector. For each j and a fixed bandwidth $h_j^{(2)}$, we estimate $\mathbf{d}_{i,j}(s)$ by minimizing an objective function given by

$$\sum_{m=1}^{M}[y_{i,j}(s_m) - \mathbf{x}_i^T\hat{B}_j(s_m) - \mathbf{d}_{i,j}(s)^T\mathbf{z}(h, s_m - s)]^2 K(h, s_m - s). \tag{5}$$

For each j, we pool the data from all n subjects and select the optimal bandwidth $h^{(2)}$, denoted by $\hat{h}^{(2)}$, and estimate $\eta_{i,j}(s)$ and $\boldsymbol{\eta}_i(s)$, denoted by $\hat{\eta}_{i,j}(s)$ and $\hat{\boldsymbol{\eta}}_i(s)$, respectively.

We estimate $\boldsymbol{\eta}(s)$ and $\Sigma_\eta(s,t)$ by using their empirical counterparts of the estimated $\hat{\boldsymbol{\eta}}_i(s)$ as follows:

$$\hat{\boldsymbol{\eta}}(s) = n^{-1}\sum_{i=1}^{n}\hat{\boldsymbol{\eta}}_{i,}(s) \quad \text{and} \quad \hat{\Sigma}_\eta(s,t) = (n-J)^{-1}\sum_{i=1}^{n}\hat{\boldsymbol{\eta}}_i(s)\hat{\boldsymbol{\eta}}_i(t)^T.$$

Let $\hat{\mathrm{B}}(s) = [\hat{B}_1(s), \cdots, \hat{B}_J(s)]$ and $\hat{\boldsymbol{\epsilon}}_i(s_m) = \mathbf{y}_i(s_m) - \hat{B}(s_m)^T\mathbf{x}_i - \hat{\boldsymbol{\eta}}_i(s_m)$ be estimated residuals for $i = 1, \cdots, n$ and $m = 1, \cdots, M$. We consider the kernel estimate of $\Sigma_\epsilon(s, s)$ given by

$$\hat{\Sigma}_\epsilon(s, s) = (n - J)^{-1} \sum_{i=1}^{n} \sum_{m=1}^{M} \frac{K(h, s_m - s)[\hat{\boldsymbol{\epsilon}}_i(s_m)]^{\otimes 2}}{\sum_{m=1}^{M} K(h, s_m - s)}. \tag{6}$$

Let $\tilde{\Sigma}_\epsilon(s_m, s_m) = (n - J)^{-1} \sum_{i=1}^{n} [\hat{\boldsymbol{\epsilon}}_i(s_m)]^{\otimes 2}$ for $m = 1, \cdots, M$. To select the optimal bandwidth $h^{(3)}$, denoted by $\hat{h}^{(3)}$, we use a cross-validation score method. Following Rice and Silverman (1991), we calculate the spectral decomposition of $\hat{\Sigma}_{\eta,jj}(s, t)$ for each j as follows:

$$\hat{\Sigma}_{\eta,jj}(s, t) = \sum_{l=1}^{\infty} \hat{\lambda}_{j,l} \hat{\psi}_{j,l}(s) \hat{\psi}_{j,l}(t), \tag{7}$$

where $\hat{\lambda}_{j,1} \geq \hat{\lambda}_{j,2} \geq \cdots \geq 0$ are estimated eigenvalues and the $\hat{\psi}_{j,l}(t)$'s are the corresponding estimated principal components.

4. Hypothesis Test

In neuroimaging studies, most scientific questions require the comparison of fiber bundle diffusion properties along fiber bundles across two (or more) diagnostic groups and the assessment of the development of fiber bundle diffusion properties along time. Such questions can often be formulated as linear hypotheses of $B(s)$ as follows:

$$H_0 : \mathbf{C}\mathrm{vec}(B(s)) = \mathbf{b}_0(s) \text{ for all } s \text{ vs. } H_1 : \mathbf{C}\mathrm{vec}(B(s)) \neq \mathbf{b}_0(s), \tag{8}$$

where $\mathbf{C}$ is a $r \times Jp$ matrix of full row rank and $\mathbf{b}_0(s)$ is a given $r \times 1$ vector of functions.

We propose both local and global test statistics. At a given grid point s_m on a specific tract, we test the local null hypothesis $H_0(s_m) : \mathbf{C}\mathrm{vec}(B(s_m)) = \mathbf{b}_0(s_m)$ against $H_1(s_m) : \mathbf{C}\mathrm{vec}(B(s_m)) \neq \mathbf{b}_0(s_m)$. Let $\tilde{H} = \mathrm{diag}(h_1^{(1)}, \cdots, h_J^{(1)})$ and $u_2(K) = \int u^2 K(u)du$. We use a local test statistic $S_n(s_m)$ defined by

$$S_n(s_m) = n\mathbf{d}(s_m)^T[\mathbf{C}(\hat{\Sigma}_\eta(s_m, s_m) \otimes \hat{\Omega}_X^{-1})\mathbf{C}^T]^{-1}\mathbf{d}(s_m), \tag{9}$$

where $\hat{\Omega}_X = n^{-1} \sum_{i=1}^{n} \mathbf{x}_i^{\otimes 2}$ and $\mathbf{d}(s) = \mathbf{C}\mathrm{vec}(\hat{B}(s) - \mathrm{bias}(\hat{B}(s))) - \mathbf{b}_0(s)$. We test the null hypothesis $H_0 : \mathbf{C}\mathrm{vec}(B(s)) = \mathbf{b}_0(s)$ for all s using a global test statistic S_n defined by

$$S_n = n \int_0^L \mathbf{d}(s)^T[\mathbf{C}(\hat{\Sigma}_\eta(s, s) \otimes \hat{\Omega}_X^{-1})\mathbf{C}^T]^{-1}\mathbf{d}(s)ds. \tag{10}$$

We develop a wild bootstrap method to approximate the p-value of S_n.

Step (i): Fit model (1) under the null hypothesis H_0, which yields $\hat{B}^*(s_j)$, $\hat{\boldsymbol{\eta}}_i^*(s_m)$ and $\hat{\boldsymbol{\epsilon}}_i^*(s_j)$ for $i = 1, \cdots, n$ and $m = 1, \cdots, M$.

Step (ii): Generate a random sample $\tau_i^{(g)}$ and $\tau_i(s_m)^{(g)}$ from a $N(0, 1)$ generator for $i = 1, \cdots, n$ and $m = 1, \cdots, M$ and then construct

$$\hat{\mathbf{y}}_i(s_m)^{(g)} = \hat{B}^*(s)^T\mathbf{x}_i + \tau_i^{(g)}\hat{\boldsymbol{\eta}}_i^*(s_m) + \tau_i(s_m)^{(g)}\hat{\boldsymbol{\epsilon}}_i^*(s_m).$$

Then, based on $\hat{\mathbf{y}}_i(s_m)^{(g)}$, we recalculate $\tilde{H}^{(1)}$, $\hat{\mathrm{B}}(s)^{(g)}$, and $\mathbf{d}(s)^{(g)} = \mathbf{C}\mathrm{vec}(\hat{\mathrm{B}}(s)^{(g)}) - \mathbf{b}_0(s)$. Subsequently, we compute $S_n^{(g)}$ and $S_n(s_m)^{(g)}$ via (9) and (10), respectively, where $\mathbf{d}(s)$ is replaced by $\mathbf{d}(s)^{(g)}$.

Step (iii): Repeat Step (ii) G times to obtain $\{S_{n,\max}^{(g)} = \max_{1\leq m\leq n_G} S_n(s_m)^{(g)} : g = 1, \cdots, G\}$ and calculate $p(s_m) = G^{-1}\sum_{g=1}^{G} 1(S_{n,\max}^{(g)} \geq S_n(s_m))$ for each s_m. Then, $p(s_m)$ is the corrected p-value at the location s_m.

Step (iv): Repeat Step (ii) G times to obtain $\{S_n^{(g)} : g = 1, \cdots, G\}$ and calculate $p = G^{-1}\sum_{g=1}^{G} 1(S_n^{(g)} \geq S_n)$. If p is smaller than a pre-specified significance level α, say 0.05, then we reject the null hypothesis H_0.

5. Confidence Bands

For a given significance level α, we construct a confidence band for each $\beta_{jl}(s)$ such that

$$P(\hat{\beta}_{jl}^{L,\alpha}(s) < \beta_{jl}(s) < \hat{\beta}_{jl}^{U,\alpha}(s) \ \text{ for all } \ s \in [0, L]) = 1 - \alpha, \tag{11}$$

where $\hat{\beta}_{jl}^{L,\alpha}(s)$ and $\hat{\beta}_{jl}^{U,\alpha}(s)$ are the lower and upper limits of the confidence band. Based on Theorem 1 of (Zhu et al., 2010a), a $1 - \alpha$ simultaneous confidence band for $\beta_{jl}(s)$ is given as follows:

$$\left(\hat{\beta}_{jl}(s) - \frac{C_{jl}(\alpha)}{\sqrt{n}}, \quad \hat{\beta}_{jl}(s) + \frac{C_{jl}(\alpha)}{\sqrt{n}}\right). \tag{12}$$

We develop an efficient resampling method to approximate $C_{jl}(\alpha)$ as follows (Zhu et al., 2007; Kosorok, 2003), see (Zhu et al., 2010a) for details.

2 Matlab functions

We implemented the FADTTS pipeline in Matlab. The following is the description of the functions in FADTTS Matlab tool. We first give an overview of the Matlab function and then explain each of the function in terms of function name, input, output, the function goal, and remarks if desired. Examples and results will be given in the next section.

2.1 Function overview

1. Multivariate Varying Coefficient Model

MVCM_read: read raw data and generate, arc length, standardized design and response matrices, and related dimension parameters.

MVCM_read1: read raw data and standardized generate design and response matrices.

MVCM_read2: read raw response data and generate standardized response matrix.

2. Weighted Least Squares Estimation

MVCM_lpks_wob: read arc length, design and response matrices and generate optimal bandwidth for weighted least squares estimation.

MVCM_lpks_wb1: read arc length, design and response matrices, and optimal bandwidth and generate the estimated coefficient functions, their first derivatives, and fitted responses using weighted least squares estimation.

3. Functional Principal Component Analysis

MVCM_sif: read arc length and the residuals of response and generate the estimations of $\boldsymbol{\eta}$, $\boldsymbol{\varepsilon}$, and $\Sigma_\eta(s,t)$.

MVCM_sif2: read arc length and the residuals of response and generate the estimations of $\boldsymbol{\eta}$, $\boldsymbol{\varepsilon}$, and $\Sigma_\eta(s,s)$.

MVCM_eigen: read the estimation of $\boldsymbol{\eta}$ and generate the individual covariance matrices and the the respective eigenvalues and eigenvectors.

MVCM_ecm: read arc length and estimation of $\boldsymbol{\varepsilon}$ and generate the covariance matrix σ_ε of $\boldsymbol{\varepsilon}$.

4. Hypothesis Test

MVCM_lpks_wb2: read arc length, design and response matrices, and optimal bandwidth and generate the estimated coefficient functions, their up to third derivatives, and fitted responses using weighted least squares estimation.

MVCM_bias read arc length, design matrix, response matrix, optimal bandwidth, and so on; and generate the bias of the estimation of $B(s)$ using its up to the third order derivative.

MVCM_ht_stat: read arc length, design matrix, the estimations of $B(s)$, $\boldsymbol{\eta}$, and $\boldsymbol{\epsilon}$, hypothesis design matrix, and so on; and generate the global and local statistics.

MVCM_grs: read design matrix, the estimations of $B(s)$, $\boldsymbol{\eta}$, and $\boldsymbol{\epsilon}$, and generate random sample of response.

MVCM_bstrp_stat: read arc length, design matrix, the estimations of $B(s)$, $\boldsymbol{\eta}$, and $\boldsymbol{\epsilon}$, hypothesis design matrix, and so on; and generate the global and local statistics without bias adjustment.

MVCM_bstrp_pvalue3: read design matrix, response matrix, the estimations of $B(s)$ and its derivatives, global test statistics, hypothesis design matrix, number of bootstrap, and so on; and generate the global p-value based on a resampling method.

5. Confidence Bands

MVCM_cb_Gval: read arc length, design and response residual matrices and so on; and generate C_{kl} for confidence bands.

MVCM_CBands: read Gvalue, efitBetas and ebiasBetas and generate the simulated confidence bands.

2.2 Function description

MVCM_read

Function [NoSetup, arclength, Xdesign, Ydesign, scalediffusion]=MVCM_read(tractdata, designdata, diffusionFiles, nofeatures)

Input `tractData:` the text file containing (x, y, z) coordinates of all locations on a given fiber tract. The data set should start from one end to the other end.
tractData is a $L \times 3$ matrix, where L_0 denotes the number of locations. 3 denotes the three coordinates.

`designData:` the text file containing covariates of interest. Please always include the intercept in the first column. designData is a $n \times p$ matrix, where n denotes the number of subjects and p denotes the number of covariates.

`diffusionFiles:` a $m \times 1$ cell containing the names of all fiber diffusion properties files. Each fiber bundle diffusion properties should contain a $L_0 \times n$ matrix. Rows correspond to the columns in tractData, while columns correspond to the columns in designData.

`nofeatures:` the number of diffusion properties, denoted by m.

`featurenames:` a $m \times 1$ cell of names.
properties.

Output `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p - 1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`scalediffusion:` a $m \times 1$ vector of scales for each properties.

Remark to avoid unnecessary errors, please use this function to preprocess the raw data before you go to the weighted least squares estimation.

MVCM_read1

Function [Xdesign, Ydesign, scalediffusion]=MVCM_read1(designdata, respdata)

Input `designData:` the text file containing covariates of interest. Please always include the intercept in the first column. designData is a $n \times p$ matrix, where n denotes the number of subjects and p denotes the number of covariates.

`respdata:` a $n \times L_0 \times m$ matrix of response variables.

Output `Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`scalediffusion:` a $m \times 1$ vector of scales for each properties.

Remark this function will save some time when just need to normalize design and response matrices.

MVCM_read2

Function [Ydesign, scalediffusion]=MVCM_read2(respdata)

Input `respdata:` a $n \times L_0 \times m$ matrix of response variables.

Output `Ydesign:` a $n \times L0 \times m$ matrix.

`scalediffusion:` a $m \times 1$ vector of scales for each properties.

Remark this function will save some time when just need to normalize response matrix.

MVCM_lpks_wob

Function [mh, GCVs, vh]=MVCM_lpks_wob(NoSetup, arclength, Xdesign, Ydesign, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `mh:` a $1 \times m$ vector of optimal bandwidth.

`GCVs:` a $nh \times m$ matrix of GCVs, where nh is the number of candidate bandwidth for each property.

`vh:` a $1 \times nh$ vector of candidate bandwidth.

Remark you need to use function MVCM_read to preprocess the data.

MVCM_lpks_wb1

Function [efitBetas, efitBetas1, InvSigmats, efitYdesign]=MVCM_lpks_wb1(NoSetup, arclength, Xdesign, Ydesign, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`mh:` a $1 \times m$ vector of optimal bandwidth.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `efitBetas:` a $p \times L_0 \times m$ matrix of estimated functional coefficients.

`efitBetas1:` a $2p \times L_0 \times m$ matrix of estimated functional coefficients and their first derivatives next to them.

`InvSigmats:` a $2p \times 2p \times L_0 \times m$ matrix; see (Zhu et al., 2010a).

`efitYdesign:` a $n \times L_0 \times m$ matrix of fitted responses.

Remark you need to use function MVCM_read to preprocess the data and function MVCM_lpks_wob to obtain the optimal bandwidth, respectively.

MVCM_sif

Function [ResEtas, efitEtas, eSigEta]=MVCM_sif(arclength, ResYdesign, kstr)

Input `arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`efitYdesign:` a $n \times L_0 \times m$ matrix of fitted responses.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `ResEtas:` a $n \times L_0 \times m$ matrix of difference between ResYdesign and fitted eta .

`efitEtas:` a $n \times L_0 \times m$ matrix of estimated etas.

`eSigEta:` a $m \times m \times L_0 \times L_0$ matrix of covariance matrix of etas.

Remark you need to run MVCM_wb1 and MVCM_read before use this function.

MVCM_sif2

Function [ResEtas, efitEtas, eSigEta]=MVCM_sif(arclength, ResYdesign, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`efitYdesign:` a $n \times L_0 \times m$ matrix of fitted responses.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `ResEtas:` a $n \times L_0 \times m$ matrix of difference between ResYdesign and fitted eta .

`efitEtas:` a $n \times L_0 \times m$ matrix of estimated etas.

`eSigEta:` a $m \times m \times L_0 \times L_0$ matrix of covariance matrix of etas. Only the diagonal covariance matrices are true and nonzero.

Remark you need to run MVCM_wb1 and MVCM_read before use this function. This function will save a lot time comparing with MVCM_sif as it only calculates the matrices when the grid points are equal, namely, $\Sigma_\eta(s, s)$, which is used in the later resampling procedure.

MVCM_eigen

Function [mSigEtaEig, mSigEta]=MVCM_eigen(efitEtas)

Input `efitEtas:` a $n \times L_0 \times m$ matrix of estimated etas.

Output `mSigEta:` a $L_0 \times L_0 \times m$ matrix of covariance matrix of etas for each individual measures

`mSigEtaEig:` a $L_0 \times (L_0 + 1) \times m$ matrix of eigenvalues and eigenvectors of individual covariance matrix of etas. The first column is eigenvalues and the rest are the respective eigenvectors.

Remark you need to either run MVCM_sif or MVCM_sif2 before use this function.

MVCM_ecm

Function [eSigE]=MVCM_ecm(arclength, ResEtas, kstr)

Input `arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`ResEtas:` a $n \times L_0 \times m$ matrix of difference between ResYdesign and fitted eta.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `eSigE:` a $L_0 \times m \times m$ matrix of the covariance of ε.

Remark you need to either run MVCM_sif or MVCM_sif2 before use this function.

MVCM_lpks_wb2

Function [efitBetas3] = MVCM_lpks_wb2(NoSetup, arclength, Xdesign,Y design, mh, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`mh:` a $1 \times m$ vector of optimal bandwidth.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `efitBetas1:` a $4p \times L_0 \times m$ matrix of estimated functional coefficients and their up to third derivatives next to them.

Remark you need to use function MVCM_read to preprocess the data and function MVCM_lpks_wob to obtain the optimal bandwidth, respectively. This function is used mainly to estimate the bias of the estimation $B(s)$; see function MVCM_bias.

MVCM_bias

Function [ebiasBetas] = MVCM_bias(NoSetup, arclength, Xdesign, Ydesign, InvSigmats, mh, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`InvSigmats:` a $2p \times 2p \times L_0 \times m$ matrix; see (Zhu et al., 2010a).

`mh:` a $1 \times m$ vector of optimal bandwidth.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `ebiasBetas:` a $p \times L_0 \times m$ matrix of the bias of the estimated betas.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1 and and MVCM_sif before use this function. You also need function MVCM_lpks_wb2 in this function.

MVCM_ht_stat

Function [Gstat, Lstat]=MVCM_ht_stat(NoSetup, arclength, Xdesign, efitBetas, eSigEta, Cdesign, B0vector, ebiasBetas)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`efitBetas:` a $p \times L_0 \times m$ matrix of estimated functional coefficients.

`eSigEta:` a $m \times m \times L_0 \times L_0$ matrix of covariance matrix of etas.

`efitBetas1:` a $p \times L_0 \times m$ matrix of the bias of the estimated betas.

`Cdesign:` a $r \times mp$ matrix for characterizing the r linear constraints among mp parameters.

`B0vector:` a $r \times L_0$ vector for hypothesis testing.

`ebiasBetas:` a $p \times L_0 \times m$ matrix of the bias of the estimated betas.

Output `Gstat:` a global test statistics.

`Lstat:` a $L_0 \times 1$ column vector of test statistics for each location.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1 and and MVCM_sif before use this function. You also need function MVCM_lpks_wb2 in this function. The global and local tets statistics here are bias adjusted.

MVCM_grs

Function [SimYdesign]=MVCM_grs(efitBetas, efitEtas, ResEtas, Xdesign)

Input `efitBetas:` a $p \times L_0 \times m$ matrix of estimated functional coefficients.

`Xdesign:` a $n \times p$ normalized design matrix.

`efitEtas:` a $n \times L_0 \times m$ matrix of estimated etas.

`ResEtas:` a $n \times L_0 \times m$ matrix of difference between ResYdesign and fitted eta .

Output `SimYdesign:` a $n \times L0 \times m$ simulated response matrix.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1 and and MVCM_sif before use this function.

MVCM_bstrp_stat

Function [Gstat, Lstat]=MVCM_bstrp_stat(arclength, Xdesign, efitBetas, eSigEta, Cdesign,
B0vector)

Input `arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`efitBetas:` a $p \times L_0 \times m$ matrix of estimated functional coefficients.

`eSigEta:` a $m \times m \times L_0 \times L_0$ matrix of covariance matrix of etas.

`Cdesign:` a $r \times mp$ matrix for characterizing the r linear constraints among mp parameters.

`B0vector:` a $r \times L_0$ vector for hypothesis testing.

Output `Gstat:` a global test statistics.

`Lstat:` a $L_0 \times 1$ column vector of test statistics for each location.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1 and and MVCM_sif before use this function. The global and local tets statistics here are without bias adjusted.

MVCM_bstrp_pvalue3

Function [Gpval]=MVCM_bstrp_pvalue3(NoSetup, arclength, Xdesign, Ydesign, efitBetas1, InvSigmats, mh, Cdesign, B0vector, Gstat, GG, kstr)

Input `NoSetup:` a column vector of $[n, L_0, p, m]$, where n is sample size, L_0 is the number of grid points, $p-1$ is the number of covariates and m is the number of features.

`arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`Ydesign:` a $n \times L0 \times m$ matrix.

`efitBetas1:` a $2p \times L_0 \times m$ matrix of estimated functional coefficients and their first derivatives next to them.

`InvSigmats:` a $2p \times 2p \times L_0 \times m$ matrix; see (Zhu et al., 2010a).

`mh:` a $1 \times m$ vector of optimal bandwidth.

`Cdesign:` a $r \times mp$ matrix for characterizing the r linear constraints among mp parameters.

`B0vector:` a $r \times L_0$ vector for hypothesis testing.

`Gstat:` a global test statistics.

`GG:` a resampling number.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `Gpval:` a p-value for the global test statistics based on a resampling method.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1, MVCM_sif, MVCM_lpks_wb2, MVCM_bias, MVCM_ht_stat before use this function. In this function, we need functions MVCM_bstrp_stat and MVCM_sif2.

MVCM_cb_Gval

Function [Gvalue]=MVCM_cb_Gval(arclength, Xdesign, ResYdesign, InvSigmats, mh, GG, kstr)

Input `arclength:` a $L_0 \times 1$ column vector of the arclength from one end to the other end.

`Xdesign:` a $n \times p$ normalized design matrix.

`ResYdesign:` a $n \times L0 \times m$ matrix of difference between fiber bundle diffusion properties and fitted fiber bundle diffusion properties .

`InvSigmats:` a $2p \times 2p \times L_0 \times m$ matrix; see (Zhu et al., 2010a).

`mh:` a $1 \times m$ vector of optimal bandwidth.

`GG:` a resampling number.

`kstr:` kernel function. The default one is $\exp(-.5t^2)$.

Output `Gvalue:` a $m \times p \times L_0 \times GG$ matrix of simulated G value matrix based on a resampling method.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, and MVCM_lpks_wb1 before use this function.

MVCM_CBands

Function [CBands] = MVCM_CBands(n, alpha, Gvalue, efitBetas, ebiasBetas)

Input `n:` number of subjects.

`alpha:` a preselected significance level.

`Gvalue:` a $m \times p \times L_0 \times GG$ matrix of simulated G value matrix based on a resampling method.

`efitBetas:` a $p \times L_0 \times m$ matrix of estimated functional coefficients.

`ebiasBetas:` a $p \times L_0 \times m$ matrix of the bias of the estimated betas.

Output `CBands:` a $2p \times L_0 \times m$ matrix of estimated confidence bands.

Remark you need to run functions MVCM_read, MVCM_lpks_wob, MVCM_lpks_wb1 and MVCM_cb_Gval before use this function.

3 Examples and Results

The example we illustrate here is using the two diffusion properties including factional anistropy (FA), mean diffusivity (MD) along the genu tract (Fig. 2). FA and MD, respectively, measure the inhomogeneous extent of local barriers to water diffusion and the averaged magnitude of local water diffusion. In the example $n = 23$, $L0 = 22$, $p = 4$ and $m = 2$. There are two groups: one is the Flu and another is Ket-Control. In this analysis, One of our interests is to know whether there is a difference between these two groups. Also we try to understand the difference between the genders, male and female, and assess the developement of diffusion properties with time. For more examples on simulation and real data, see (Zhu et al., 2010a).

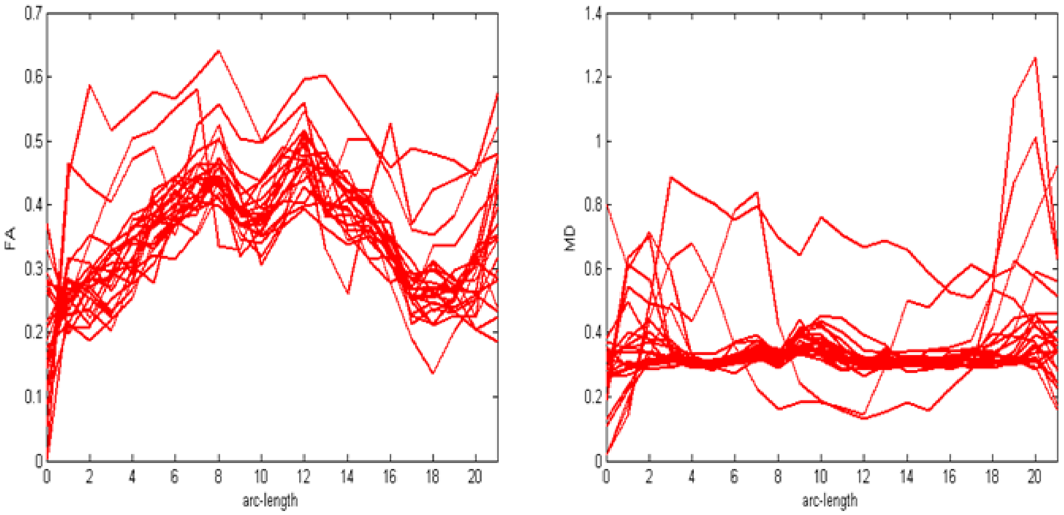

Figure 2: FA and MD along the genu tract

1. Multivariate Varying Coefficient Model

The FADTTS model we considered is

$$\begin{aligned}
(\mathrm{FA}_i(s_j), \mathrm{MD}_i(s_j))^T &= (\mathbf{x}_i^T B_1(s_j), \mathbf{x}_i^T B_2(s_j))^T + \boldsymbol{\eta}_i(s_j) + \boldsymbol{\epsilon}_i(s_j), \\
\mathbf{x}_i^T B_1(s) &= \beta_{11}(s) + \beta_{12}(s) \times \mathrm{G}_i + \beta_{13}(s) \times \mathrm{Flu}_i + \beta_{14}(s) \times \mathrm{Age}_i, \\
\mathbf{x}_i^T B_2(s) &= \beta_{21}(s) + \beta_{22}(s) \times \mathrm{G}_i + \beta_{23}(s) \times \mathrm{Flu}_i + \beta_{24}(s) \times \mathrm{Age}_i, \qquad (13)
\end{aligned}$$

where G_i, Flu_i and Age_i, respectively, denote gender, Flu (0 for Flu and 1 for Ket-control), and the gestational age at the scan time of the i-th infant, $\boldsymbol{\eta}_i(s) = (\eta_{i1}(s), \eta_{i2}(s))^T$ is a Gaussian process with zero mean and covariance matrix $\Sigma_\eta(s,t)$ and $\boldsymbol{\epsilon}_i(s) = (\epsilon_{i1}(s), \epsilon_{i2}(s))^T$ is a Gaussian random vector with zero mean and covariance matrix $\Sigma_\epsilon(s,s)\mathbf{1}(s=t)$. As stated, $i = 1, 2, \cdots, 23$ and $j = 1, 2, \cdots, 22$.

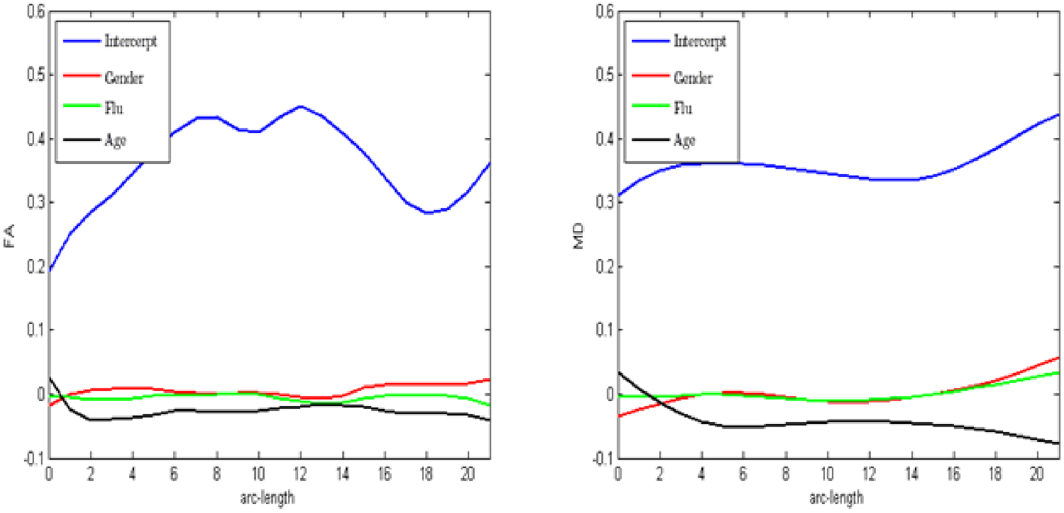

Figure 3: estimated functional coefficients for FA and MD along the genu tract

2. Weighted Least Squares Estimation

There are three data sets we need to import, namely, `tractdata, designdata, diffusionFiles`. The data set `tractdata` contains (x, y, z) coordinates of all locations on a given fiber tract. The data set should start from one end to the other end. `tractData` is a $L_0 \times 3$ matrix, where $L_0 = 22$ denotes the number of locations. 3 denotes the three coordinates. The following shows the first 5 rows of `tractData`,

```
-11 0 0
-10 0 0
-9 0 0
-8 0 0
-7 0 0
...
```

the data set `designdata` contains covariates of interest. We always need to 1include the intercept in the first column. `designData` is a $n \times p$ matrix, where $n = 23$ denotes the number of subjects and $p = 4$ denotes the number of covariates, as we have intercept, gender, age and gage. Each covariate is listed in one column. The following shows the first 5 rows, where the columns are, respectively, intercept, gender, flu and age.

```
1 1 0 375
1 0 0 381
1 1 0 378
1 1 1 366
1 1 1 377
...
```

The data set `diffusionFiles` is a $m(= 2) \times 1$ cell containing the names of all fiber diffusion properties files. Each fiber bundle diffusion properties should contain a $L_0(=22) \times n(=23)$ matrix. Rows correspond to the columns in `tractData`, while columns correspond to the columns in `designData`. In particular, we use

```
diffusionFiles=cell(2,1);
```

to define the cell structure. We then specify the first cell `diffusionFiles{1}` as FA values and the second `diffusionFiles{2}` as MD values. Both are $n \times L_0$ matrices. The following are the first 5 rows of `diffusionFiles{1}`.

```
0.3286 0.2782 0.2172 0.2095 0.2939 0.1450 ···
0.2342 0.2104 0.2800 0.2621 0.2256 0.2578 ···
0.2954 0.2066 0.2514 0.3040 0.2378 0.2897 ···
0.3411 0.2922 0.3336 0.2956 0.2952 0.3268 ···
0.3609 0.2759 0.2904 0.3043 0.3684 0.3482 ···
···
```

After load covariates, response and arc length data, we use MVCM_read to transfer data into

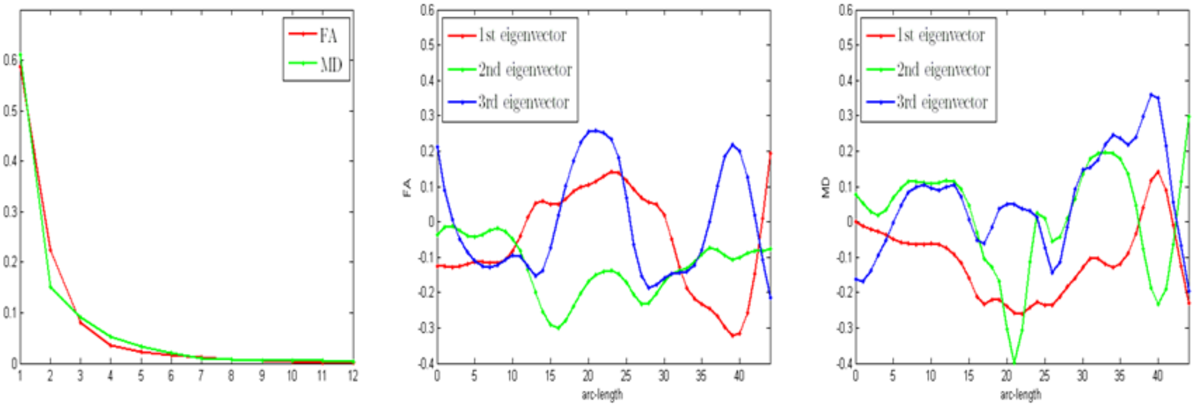

Figure 4: estimated first 12 relative eigenvalues and first 3 eigenfunctions for FA and MD along the genu tract

the format we want.

```
[NoSetup, arclength, Xdesign, Ydesign]
      =MVCM_read(tractdata, designdata, diffusionFiles, nofeatures);
```

We use function MVCM_lpks_wob to find the optimal bandwidth.

```
[mh]
      =MVCM_lpks_wob(NoSetup, arclength, Xdesign, Ydesign);
mh=1.3978 3.0318
```

After we got the optimal bandwidth, we are able to estimate the coefficients using function MVCM_lpks_wb1.

```
[efitBetas, efitBetas1, InvSigmats, efitYdesign]
      =MVCM_lpks_wb1(NoSetup, arclength, Xdesign, Ydesign, mh);
```

The estimated coefficients are shown in Fig. 3.

3. Functional Principal Component Analysis

Next we find the residual of response.

```
ResYdesign=Ydesign-efitYdesign;
```

Then we use function MVCM_sif to estimate $\boldsymbol{\eta}$ and MVCM_eigen to estimate the eigenvalues and eigenfunctions of $\hat{\Sigma}_{\eta,kk}$.

```
[ResEtas, efitEtas, eSigEta]
      =MVCM_sif(arclength, ResYdesign);
[mSigEtaEig, mSigEta]
      =MVCM_eigen(efitEtas);
```

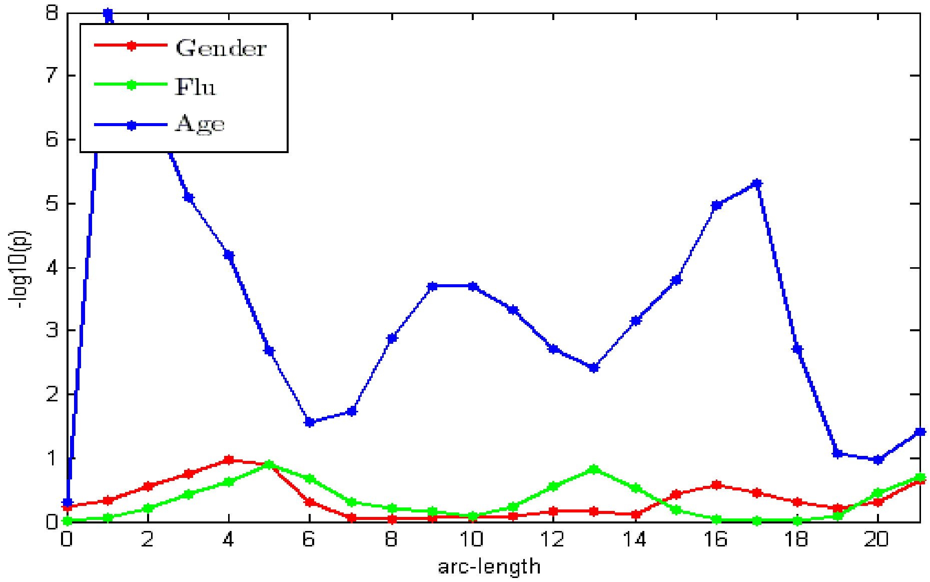

Figure 5: $-\log_{10}$ local p-values for testing Gender, Flu and Age effects for FA and MD together along the genu tract

Fig. 4 shows the estimated first 12 relative eigenvalues and first 3 eigenfunctions for FA and MD along the genu tract. The relative eigenvalues of $\hat{\Sigma}_{\eta,kk}$ are defined as the ratios of the eigenvalues of $\hat{\Sigma}_{\eta,kk}$ over their sum. To estimate the covariance matrix of $\boldsymbol{\varepsilon}$, we use function MVCM_ecm.

```
[eSigE]
      =MVCM_ecm(arclength, ResEtas);
```

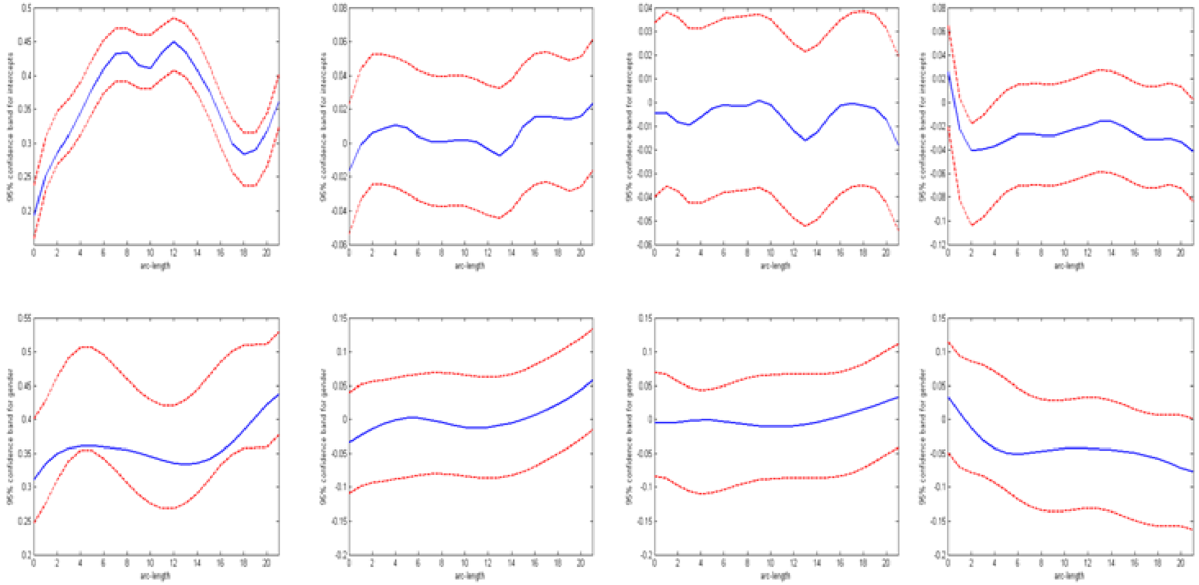

Figure 6: 95% confidence bands of Gender, Flu and Age effects for FA and MD together along the genu tract: red dashed lines are confidence bands; blue lines are estimated $B(s)$; the first row is for FA and the second row is for MD.

4. Hypothesis Test

Before we test the significance of the covariates, we first need to estimate the bias of $\hat{B}(s)$ using function MVCM_bias.

```
[ebiasBetas]=
        MVCM_bias(NoSetup, arclength, Xdesign,Ydesign, InvSigmats, mh);
```

To use hypothesis test, we need to specify the matrix for the coefficient functions and their corresponding values under the null hypothesis. For example, to test the significance of over all effect of Age, the matrix $\mathbf{C}$ and $\mathbf{b}_0(s)$ are, respectively,

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{b}_0(s) \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{for all} \quad s.$$

Using function MVCM_ht_stat, we found the global test statistics S_n for testing the effects of gender, flu and age are, respectively, 35.3904, 33.4591, and 332.7554.

```
[Gstat, Lstat]=
        MVCM_ht_stat(NoSetup, arclength, Xdesign, efitBetas, eSigEta, Cdesign,
        B0vector, ebiasBetas);
```

The corresponding p-values are 0.658, 0.617, and 0.007 calculated by function MVCM_bstrp_pvalue3 based on a resampling method with $G = 1000$ replication.

```
[Gpval]=
       MVCM_bstrp_pvalue3(NoSetup, arclength, Xdesign, Ydesign, efitBetas1,
        InvSigmats, mh, Cdesign, B0vector, Gstat, GG);
```

We can also find the local p-values by

```
Lpvals=1-chi2cdf(Lstats,m);
```

Fig. 5 shows the $-\log_{10}$ local p-values for testing Gender, Flu and Age effects for FA and MD together along the genu tract.

5. Confidence Bands

Let $\alpha = 0.05$ (you chose other values if desired), we use function MVCM_cb_Gval to estimate $\mathbf{C_{kl}}$ based on a resampling method.

```
[Gvalue]=
        MVCM_cb_Gval(arclength, Xdesign, ResYdesign, InvSigmats, mh, GG);
```

The the simultaneous confidence bands were found by function MVCM_CBands.

```
[CBands]=
        MVCM_CBands(n, alpha, Gvalue, efitBetas, ebiasBetas);
```

Fig. 6 shows 95% confidence bands of Gender, Age and Gage effects for FA and MD together along the genu tract (red dashed lines).

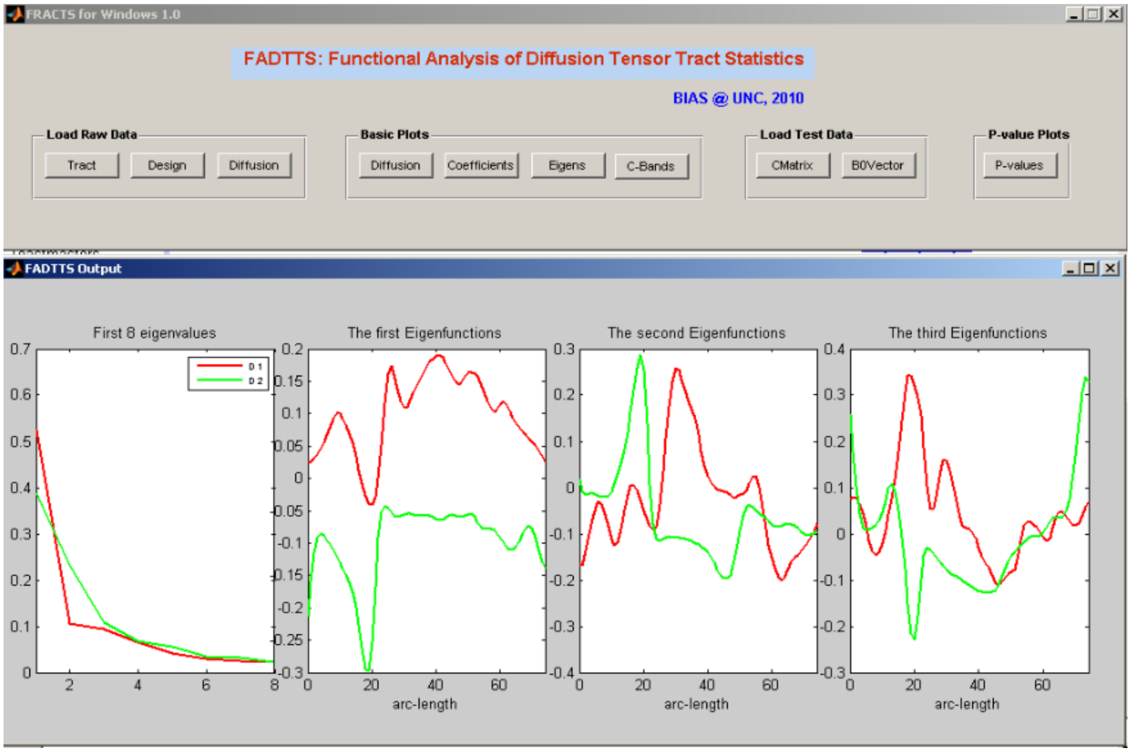

Figure 7: FRATS GUI

4 FADTTS: graphical user interface (GUI)

To make it easily accessible, we developed a Graphical User Interface (GUI) to pack the code. As shown in Figure 7, there are 4 button groups, which are supposed to be executed in order. The 4 groups are s Load Raw Data, *Basic Plots*, *Load Test Data*, and *P-value Plots*. There are 3 raw data sets, namely, tract data, design data and diffusion data. The test data sets include test design matrix and null hypothesis vector. All data sets must be in `.mat`. The package includes a sample matlab code `pre_address_data.m` on how to set up data. After loading all raw data, GUI will transfer the raw data, estimate the coefficients, do spectral decomposition and estimate confidence bands. Then you can plot the raw tract data, the coefficient functions, spectral decomposition and confidence bands by pushing the corresponding buttons. If you want to do a test, you need to load the test design data. There are two types of test. One is to test individually and the other one is test all the diffusion properties together. Once you loaded the test design data, GUI will display what test type you requested. The test calculation may take a while. After matlab finishes the computation, GUI will report the global test statistics and p-values. You also have the option to plot the local p-values.

References

Ashburner, J. and Friston, K. J. (2000). Voxel-based morphometry: the methods. *Neuroimage*, 11:805–821.

Basser, P. J., Mattiello, J., and LeBihan, D. (1994a). Estimation of the effective self- diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance Ser. B*, 103:247–254.

Basser, P. J., Mattiello, J., and LeBihan, D. (1994b). Mr diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267.

Bonekam, D., Nagae, L. M., Degaonkar, M., Matson, M., Abdalla, W. M., Barker, P. B., Mori, S., and Horská, A. (2008). Diffusion tensor imaging in children and adolescents: Reproducibility, hemispheric, and age-related differences. *NeuroImage*, 34:733–742.

Camara, E., Bodammer, N., Rodriguez-Fornells, A., and Tempelmann, C. (2007). Age-related water diffusion changes in human brain: A voxel-based approach. *NeuroImage*, 34:1588–1599.

Chen, Y. S., An, H. Y., Zhu, H. T., Stone, T., Smith, J. K., Hall, C., Bullitt, E., Shen, D. G., and Lin, W. L. (2009). White matter abnormalities revealed by diffusion tensor imaging in non-demented and demented hiv+ patients. *NeuroImage*, 47:1154–1162.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. Chapman and Hall, London.

Fan, J., Yao, Q., and Cai, Z. (2003). Adaptive varying-coefficient linear models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 65(1):57–80.

Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *Ann. Statist.*, 27(5):1491–1518.

Fan, J. and Zhang, W. (2008). Statistical methods with varying coefficient models. *Stat. Interface*, 1(1):179–195.

Focke, N. K., Yogarajah, M., Bonelli, S. B., Bartlett, P. A., Symms, M. R., and Duncan, J. S. (2008). Voxel-based diffusion tensor imaging in patients with mesial temporal lobe epilepsy and hippocampal sclerosis. *NeuroImage*, 40:728–737.

Gilmore, J. H., Smith, L. C., Wolfe, H. M., Hertzberg, B. S., Smith, J. K., Chescheir, N. C., Evans, D. D., Kang, C., Hamer, R. M., Lin, W., and Gerig, G. (2008). Prenatal mild ventriculomegaly predicts abnormal development of the neonatal brain. *Biol Psychiatry*, 64:1069–1076.

Goodlett, C. B., Fletcher, P. T., Gilmore, J. H., and Gerig, G. (2009). Group analysis of dti fiber tract statistics with application to neurodevelopment. *NeuroImage*, 45:S133–S142.

Hall, P., Müller, H.-G., and Wang, J.-L. (2006). Properties of principal component methods for functional and longitudinal data analysis. *Ann. Statist.*, 34(3):1493–1517.

Hecke, W. V., Sijbers, J., Backer, S. D., Poot, D., Parizel, P. M., and Leemans, A. (2009). On the construction of a ground truth framework for evaluating voxel-based diffusion tensor mri analysis methods. *NeuroImage*, 46:692–707.

Jones, D. K., Symms, M. R., Cercignani, M., and Howard, R. J. (2005). The effect of filter size on vbm analyses of dt-mri data. *NeuroImage*, 26:546–554.

Kosorok, M. R. (2003). Bootstraps of sums of independent but not identically distributed stochastic processes. *J. Multivariate Anal.*, 84:299–318.

Li, Y. and Hsing, T. (2010). Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics*, page in press.

O'Donnell, L., Westin, C.-F., and Golby, A. (2009). Tract-based morphometry for white matter group analysis. *Neuroimage*, 45:832–844.

Ramsay, J. O. and Silverman, B. W. (2002). *Applied functional data analysis.* Springer Series in Statistics. Springer-Verlag, New York. Methods and case studies.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis.* Springer-Verlag, New York.

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *J. Roy. Statist. Soc. Ser. B*, 53(1):233–243.

Smith, S. M., Jenkinson, M., Johansen-Berg, H., Rueckert, D., Nichols, T. E., Mackay, C. E., Watkins, K. E., Ciccarelli, O., Cader, M., Matthews, P., and Behrens, T. E. (2006). Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data. *NeuroImage*, 31:1487–1505.

Snook, L., Paulson, L. A., Roy, D., Phillips, L., and Beaulieu, C. (2005). Diffusion tensor imaging of neurodevelopment in children and young adults. *NeuroImage*, 26:1164–1173.

Snook, L., Plewes, C., and Beaulieu, C. (2007). Voxel based versus region of interest analysis in diffusion tensor imaging of neurodevelopment. *NeuroImage*, 34:243–252.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing.* Chapman and Hall, London.

Wang, L., Li, H., and Huang, J. Z. (2008). Variable selection in nonparametric varying-coefficient models for analysis of repeated measurements. *J. Amer. Statist. Assoc.*, 103(484):1556–1569.

Welsh, A. H. and Yee, T. W. (2006). Local regression for vector responses. *Journal of Statistical Planning and Inference*, 136:3007–3031.

Wu, C. O. and Chiang, C.-T. (2000). Kernel smoothing on varying coefficient models with longitudinal dependent variable. *Statist. Sinica*, 10(2):433–456.

Wu, H. L. and Zhang, J. T. (2006). *Nonparametric Regression Methods for Longitudinal Data Analysis.* John Wiley & Sons, Inc., Hoboken, New Jersey.

Yao, F. and Lee, T. C. M. (2006). Penalized spline models for functional principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):3–25.

Yushkevich, P. A., Zhang, H., Simon, T., and Gee, J. C. (2008). Structure-specific statistical mapping ofwhite matter tracts. *Neuroimage*, 41:448–461.

Zhang, J. and Chen, J. (2007). Statistical inference for functional data. *The Annals of Statistics*, 35:1052–1079.

Zhu, H. T., Ibrahim, J. G., Tang, N., Rowe, D., Hao, X., Bansal, R., and Peterson, B. S. (2007). A statistical analysis of brain morphology using wild bootstrapping. *IEEE Trans Med Imaging*, 26:954–966.

Zhu, H. T., Li, R. Z., and Kong, L. L. (2010a). Multivariate varying coefficient models for functional responses. Technical report, University of North Carolina at Chapel Hill.

Zhu, H. T., Styner, M., Tang, N. S., Liu, Z. X., Lin, W. L., and Gilmore, J. (2010b). Frats: Functional regression analysis of dti tract statistics. *IEEE Transactions on Medical Imaging*, 29:1039–1049.