Functional data analytics for wearable device and neuroscience data

Julia Wrobel

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2019

ABSTRACT

Functional data analytics for wearable device and neuroscience data

Julia Wrobel

This thesis uses methods from functional data analysis (FDA) to solve problems from three scientific areas of study. While the areas of application are quite distinct, the common thread of functional data analysis ties them together. The first chapter describes interactive open-source software for explaining and disseminating results of functional analyses. Chapters two and three use curve alignment, or registration, to solve common problems in accelerometry and neuroimaging, respectively. The final chapter introduces a novel regression method for modeling functional outcomes that are trajectories over time.

The first chapter of this thesis details a software package for interactively visualizing functional data analyses. The software is designed to work for a wide range of datasets and several types of analyses. This chapter describes that software and provides an overview of FDA in different contexts. The second chapter introduces a framework for curve alignment, or registration, of exponential family functional data. The approach distinguishes itself from previous registration methods in its ability to handle dense binary observations with computational efficiency. Motivation comes from the Baltimore Longitudinal Study on Aging, in which accelerometer data provides valuable insights into the timing of sedentary behavior. The third chapter takes lessons learned about curve registration from the second chapter and use them to develop methods in an entirely new context: large multisite brain imaging studies. Scanner effects in multisite imaging studies are non-biological variability due to technical differences across sites and scanner hardware. This method identifies and removes scanner effects by registering cumulative distribution functions of image intensities values. In the final chapter the focus shifts from curve registration to regression. Described within this chapter is an entirely new nonlinear regression framework that draws from both functional data analysis and systems of ordinary equations. This model is motivated by the

neurobiology of skilled movement, and was developed to capture the relationship between neural activity and arm movement in mice.

Table of Contents

List of Figures

Acknowledgments

I would like to thank my committee members Yifei Sun, Gen Li, and Joseph E. Schwartz for their time, as well as their thoughtful questions during the defense of this dissertation.

I feel gratitude towards the Department of Biostatistics at Columbia for supporting me financially, as well as to Georgia Andre, Katy Hardy, and Justine Herrera for helping me navigate my way through pizza-order woes, intricacies of the payroll bureaucracy, and pretty much everything else. I am grateful to my fellow students for attending my GSRS talks and sharing the graduate school experience.

Christine Mauro, I am thankful for your mentorship, patience, and friendship. Todd Ogden, thank you for your insightful comments and critiques in FDAWG, which have allowed me to confidently give talks at conferences knowing the hardest questions about my presentation have already been asked. Never kill the DAWG!

Vadim Zipunnikov, thank you for teaching me much about wearable devices. To Taki Shinohara- thank you for sharing your expertise in imaging statistics, your data, and your collaborators. Thanks, too, for being the nice one.

Jeff Goldsmith, I am thankful for your mentorship over the last 6 years. Your guidance and belief in me have been crucial for my growth as a statistician and as a person. I am grateful for your thorough feedback, even if I sometimes tell you otherwise. I know I said Taki is the nice one but you're the nice one too.

Thanks to Mom, Dad, and Greg. In particular, thanks to a living room full of Wrobels for, during Christmas, hearing my early stage ideas on what has become the final chapter of this work. Mom, thank you for listening to my Bayesian statistics lecture while we made Thanksgiving dinner. I thank both the Wrobel and Kronik sides for displaying unwavering enthusiasm for knowledge throughout my life. Collectively you have created an environment where giving statistics lectures is a perfectly acceptable thing to do at holiday get-togethers.

Chapter 1

Introduction

Functional data analysis (FDA) has become a popular and useful framework for applications in which the unit of measurement is a function, curve or image. The basic unit of observation is the curve $Y_i(t)$ for $i \in \ldots, I$, where i denotes the index for a particular curve, and t is the domain of the function Y. This thesis comes in four parts, each with different objectives and motivations. The common theme is that some part of the problem can be conceptualized using a functional data framework, and this work leverages that structure to answer scientific questions.

The focus of the first chapter is an open-source software package, `refund.shiny`, that enables interactive visualization of the results of functional data analyses. Although there are established graphics that accompany the most common functional data analyses, generating these graphics for each dataset and analysis can be cumbersome and time consuming. Often, the barriers to visualization inhibit useful exploratory data analyses and prevent the development of intuition for a method and its application to a particular dataset. The `refund.shiny` package was developed to address these issues for several of the most common functional data analyses. After conducting an analysis, the `plot_shiny()` function is used to generate an interactive visualization environment that contains several distinct graphics, many of which are updated in response to user input.

These visualizations reduce the burden of exploratory analyses and can serve as a useful tool for the communication of results to non-statisticians.

The second chapter introduces a novel method for registration, or separating amplitude and phase variability in exponential family functional data, which is motivated by accelerometer data from the Baltimore Longitudinal Study on Aging. For this chapter, the curve $Y_i(t)$ is a 24-hour activity profile, where t is time and each i is a subject. Our method alternates between two steps: the first uses generalized functional principal components analysis to calculate template functions, and the second estimates smooth warping functions that map observed curves to templates. Existing approaches to registration have primarily focused on continuous functional observations, and the few approaches for discrete functional data require a pre-smoothing step; these methods are frequently computationally intensive. In contrast, we focus on the likelihood of the observed data and avoid the need for preprocessing, and we implement both steps of our algorithm in a computationally efficient way. We analyze binary functional data with observations each minute over 24 hours for 592 participants, where values represent activity and inactivity. Diurnal patterns of activity are obscured due to misalignment in the original data but are clear after curves are aligned. Simulations designed to mimic the application indicate that the proposed methods outperform competing approaches in terms of estimation accuracy and computational efficiency.

The third chapter uses registration in brain imaging to solve a common problem in multisite studies. For this application, the $Y_i(t)$ are not the images themselves, but distribution functions of image voxel intensity values. The domain t is intensity, and each i is an image. In multisite neuroimaging studies there is often unwanted technical variation across scanners and sites. These "scanner effects" can hinder detection of biological features of interest, produce inconsistent results, and lead to spurious associations. We assess scanner effects in two brain magnetic resonance imaging (MRI) studies where subjects were measured on multiple scanners within a short time frame, so

that one could assume any differences between images were due to technical rather than biological effects. We propose *mica* (**m**ultisite **i**mage harmonization by **C**DF **a**lignment), a tool to harmonize images taken on different scanners by identifying and removing within-subject scanner effects. Our goals in the present study were to (1) establish a method that removes scanner effects by leveraging multiple scans collected on the same subject, and, building on this, (2) develop a technique to quantify scanner effects in large multisite trials so these can be reduced as a preprocessing step. We found that unharmonized images were highly variable across site and scanner type, and our method effectively removed this variability by warping intensity distributions. We further studied the ability to predict intensity harmonization results for a scan taken on an existing subject at a new site using cross-validation.

The final chapter combines a dynamical systems approach, where inputs and outputs continuously evolve over time, with concepts from functional regression. The chapter introduces a nonlinear regression model for understanding the dynamics in an experiment where the outcome data are repeated observations of trajectories. Each $Y_i(t)$ is a single trajectory from experimental trial i, measured over time t. Our work is motivated by data from an experiment exploring the relationship between neural firing rates and hand trajectories of mice performing a reaching task while under neurological assessment. The result is the *flode* (**f**unctional **l**inear **o**rdinary **d**ifferential **e**quation) model, a functional regression model which is also a first-order differential equation. This models how neural activity in the motor cortex of the brain changes the position and velocity of the paw over time as it makes a reaching motion. Simulations indicate that our method performs well under several conditions, and outperforms a more traditional functional regression model.

Chapter 2

Interactive graphics for FDA

2.1 Introduction

Conceptually, FDA leverages the underlying data structure, often temporal or spatial, to improve understanding of patterns and variation. A wide array of tools have been developed for the functional data setting, for example, functional principal components analysis (FPCA) and regression models using functional responses [Ramsay and Silverman, 2005; Morris, 2015; Sørensen *et al.*, 2013]. The basic unit of observation is the curve $Y_i(t)$ for subjects $i \in \ldots, I$ in the cross-sectional setting and $Y_{ij}(t)$ for subject i at visit $j \in \ldots, J_i$ for the multilevel or longitudinal structure. Methods for functional data are typically presented in terms of continuous functions, but in practice data are observed on a discrete grid that may be sparse or dense at the subject level and that may be the same across subjects or irregular.

Many methods for FDA have standard visualization approaches that clarify the results of analyses; examples include scree plots for FPCA and coefficient function plots for function on scalar regression. Clear visualizations aid in exploratory analysis and help to communicate results to non-statistical collaborators. However, creating useful plots is often time consuming and must be repeated each time a model is changed, and no software currently exists to facilitate this process.

The `refund.shiny` package [Goldsmith and Wrobel, 2015] creates interactive visualizations for functional data analyses, allowing researchers to create common graphics for standard analyses with just a few lines of code. Currently, `refund.shiny` builds plots for functional principal components analysis (FPCA), multilevel FPCA (MFPCA), time-varying FPCA (TFPCA), and function-on-scalar regression (FoSR). The workflow separates analysis and visualization steps: analyses are performed by functions in the `refund` package [Crainiceanu *et al.*, 2015] and interactive visualizations are generated by the `plot_shiny()` function in the `refund.shiny` package. Changes to the analysis – increasing the number of retained principal components, for example, or augmenting a regression model with new predictors – are easily incorporated into the graphical interface. User interaction with the displayed graphics facilitates comparisons and streamlines navigation between visualizations.

We illustrate the tools in `refund.shiny` using a single dataset, which we describe briefly here. The diffusion tensor imaging (`DTI`) dataset available in the `refund` package includes cerebral white matter tracts for multiple sclerosis patients and healthy controls. White matter tracts are collections of axons, the projections of neurons that transmit electrical signals that are coated by a fatty substance called myelin [Goldsmith *et al.*, 2011, 2012]. DTI is a magnetic resonance imaging modality that measures diffusion of water in the brain; because water movement is restricted in white matter fibers, DTI allows the quantification of white matter tract integrity. The `DTI` dataset contains tract profiles – continuous summaries of tract properties along their major axis – for 142 subjects across multiple visits, with a median of 4 scans per subject. The dataset includes tract profiles for several tracts, the PASAT score (a continuous variable that indicates brain reactivity and attention span), subject sex, subject ID, visit number, and time of visit [Strauss *et al.*, 2006]. Because we observe tract profiles for each subject over time, the DTI dataset is a functional dataset with longitudinal structure; in order to use the same dataset across examples

we sometimes neglect this structure or subset the data. The following code can be used to install `refund` and `refund.shiny` and load the `DTI` data:

```
> install.packages("refund.shiny")
> library(refund.shiny)
> library(refund)
> data(DTI)
```

Sections 2.2, 2.3, 2.4, and 2.5 each provide a brief methodological overview of an analysis technique for FDA and describe the corresponding interactive visualization tools in the `refund.shiny` package. Section 2.6 details the structure of the `refund.shiny` package. We close in section 2.7 with a discussion.

2.2 Functional Principal Components Analysis

We start with functional principal components analysis (FPCA), one of the most common exploratory tools for functional datasets.

2.2.1 FPCA Model

FPCA characterizes modes of variability by decomposing functional observations into population level basis functions and subject-specific scores [Ramsay and Silverman, 2005]. The basis functions have a clear interpretation, analogous to that of PCA: the first basis function explains the largest direction of variation, and each subsequent basis function describes less. The FPCA model is typically written

$$Y_i(t) = \mu(t) + \sum_{k=1}^{K} c_{ik}\psi_k(t) + \epsilon_i(t) \tag{2.1}$$

where $\mu(t)$ is the population mean, $\psi_k(t)$ are a set of orthonormal population-level basis functions, c_{ik} are subject-specific scores with mean zero and variance λ_k, and $\epsilon_i(t)$ are residual curves. Estimated basis functions $\widehat{\psi}_1(t), \widehat{\psi}_2(t), \ldots, \widehat{\psi}_K(t)$ and corresponding variances $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \ldots \geq \widehat{\lambda}_K$ are obtained from a truncated Karhunen-Loève decomposition of the sample covariance $\widehat{\Sigma}(s,t) = \widehat{\text{Cov}}(Y_i(s), Y_i(t))$. In practice, the covariance $\widehat{\Sigma}(s,t)$ is often smoothed using a bivariate smoother that omits entries on the main diagonal to avoid a "nugget effect" attributable to measurement error, and scores are estimated in a mixed model framework [Yao *et al.*, 2005]. The truncation lag K is often chosen so that the resulting approximation accounts for at least 95% of observed variance.

2.2.2 Graphics for FPCA

Our example uses the `fpca.sc()` function from the `refund` package. Several other implementations of FPCA are available in `refund`, including `fpca.face()`, `fpca.ssvd()`, and `fpca2s()`, all of which are compatible with `refund.shiny`. The number of functional principal components (FPCs) is chosen by percent variance explained, with the default set to 99 percent. See `?plot_shiny` for examples.

Graphics for FPCA are implemented by the code below:

```
> fit.fpca = fpca.sc(Y = DTI$cca)
> plot_shiny(obj = fit.fpca)
```

Executing this code produces a user interface with five tabs. The first tab shows $\widehat{\mu}(t) \pm \sqrt{\widehat{\lambda}_k}\widehat{\psi}_k(t)$, and includes a drop-down menu through which the user can select k (an example for a similar tab, based on multilevel data, is shown if Section 2.3). The second tab presents static scree plots of the eigenvalues $\widehat{\lambda}_k$ and the percent variance explained by each eigenvalue. The third tab shows $\widehat{\mu}(t) + \sum_{k=1}^{K} c_k \widehat{\psi}_k(t)$, and includes slider bars through which the values of c_k can be set; adjusting the

sliders allows the user to see a fitted curve for a hypothetical subject with the selected combination of scores. The fourth tab allows users to assess quality-of-fit by plotting fitted and observed values for any subject in the dataset.

Figure 2.1: Screenshot showing tab 5 of the interactive graphics for FPCA. A scatterplot of FPC loadings $\widehat{c}_{ik}$ against $\widehat{c}_{ik'}$ is shown in the upper plot, and k and k' are selected using drop-down menus at the left. The lower plot shows fitted curves for all subjects. In the scatterplot, a subset of estimated loadings can be selected by clicking-and-dragging to create a blue box; blue curves in the plot of fitted values correspond to selected points in upper plot.

The fifth tab for the interactive graphic produced by the code above is shown as a static plot in Figure 2.1. A scatterplot of estimated FPC loadings $\widehat{c}_{ik}$ against $\widehat{c}_{ik'}$ is shown in the upper plot, and k and k' are selected using drop-down menus at the left. The lower plot shows fitted curves for all subjects. In the scatterplot, a subset of FPC loadings can be selected by clicking-and-dragging to create a blue box; blue curves in the plot of fitted values correspond to selected subjects in

upper plot. In Figure 2.1 the first and second FPCs are selected for the x and y axes of the score plot, respectively, and several subjects that have negative values for FPC 1 are highlighted. Fitted values for these subjects are clustered at the top of the y-axis, indicating that the first FPC largely represents a vertical shift from the mean. A working example of `refund.shiny` for FPCA on a different dataset is available at https://jeff-goldsmith.shinyapps.io/FPCA.

2.3 Multilevel Functional Principal Components Analysis

Multilevel functional principal components analysis (MFPCA) extends the ideas of FPCA to functional data with a multilevel structure.

2.3.1 MFPCA Model

Multilevel functional data are increasingly common in practice; in the case of our DTI example, this structure arises from multiple clinical visits made by each subject. MFPCA models the within-subject correlation induced by repeated measures as well as the between-subject correlation modeled by classic FPCA. This leads to a two-level FPC decomposition, where level 1 concerns subject-specific effects and level 2 concerns visit-specific effects. Population-level basis functions and subject-specific scores are calculated for both levels [Di *et al.*, 2009, 2014]. The MFPCA model is:

$$Y_{ij}(t) = \mu(t) + \eta_j(t) + \sum_{k_1=1}^{K_1} c_{ik}^{(1)}\psi_k^{(1)}(t) + \sum_{k_2=1}^{K_2} c_{ijk}^{(2)}\psi_k^{(2)}(t) + \epsilon_{ij}(t) \tag{2.2}$$

where $\mu(t)$ is the population mean, $\eta_j(t)$ is the visit-specific shift from the overall mean, $\psi_k^{(1)}(t)$ and $\psi_k^{(2)}(t)$ are the eigenfunctions for levels 1 and 2, respectively, and $c_{ik}^{(1)}$ and $c_{ijk}^{(2)}$ are the subject-specific and subject-visit-specific scores. Often, visit-specific means $\eta_j(t)$ are not of interest and can be omitted from the model. Estimation for MFPCA extends the approach for FPCA: estimated between- and within-covariances $\widehat{\Sigma}^{(1)}(s,t) = \widehat{\text{Cov}}(Y_{ij}(s), Y_{ij'}(t))$ for $j \neq j'$ and $\widehat{\Sigma}^{(2)}(s,t) = \widehat{\text{Cov}}(Y_{ij}(s), Y_{ij}(t))$

are derived from the observed data, smoothed, and decomposed to obtain eigenfunctions and values. Given these objects, scores are estimated in a mixed-model framework.

2.3.2 Graphics for MFPCA

MFPCA is implemented in the `mfpca.sc()` function from the `refund` package. By default, `mfpca.sc` does not calculate visit-means, but they can be calculated by specifying the `mfpca.sc()` argument `twoway = TRUE`.

Graphics for MFPCA are implemented by the code below:

```
> Y = DTI$cca
> id = DTI$ID
> fit.mfpca = mfpca.sc(Y = Y, id = id, twoway = FALSE)
> plot_shiny(fit.mfpca)
```

This code produces an interface with five tabs, which is similar to the interface for FPCA but includes features unique to multilevel analyses. Tabs 1, 2, 3, and 5 for MFPCA are $\widehat{\mu}(t) \pm \sqrt{\widehat{\lambda}_{k_L}^{(L)}}\,\widehat{\psi}_{k_L}^{(L)}(t)$, static scree plots of the estimated eigenvalues $\widehat{\lambda}_{k_L}^{(L)}$, $\widehat{\mu(t)} + \sum_{k_L=1}^{K_L} c_{k_L}^{(L)}\widehat{\psi}_{k_L}^{(L)}(t)$, and scatterplots of FPC scores (similar to Figure 2.1), respectively. These mirror the tabs for FPCA and include inset sub-tabs to toggle between level, L, to display results for level 1 or level 2. The fourth tab plots fitted and observed values for any user-selected subject in the dataset; the user can display all visits for the selected subject or choose a subset of visits. The first tab for the interactive visualization produced by the code above is displayed in Figure 2.2, and shows $\widehat{\mu}(t) \pm \sqrt{\widehat{\lambda}_2^{(1)}}\,\widehat{\psi}_2^{(1)}(t)$.

2.4 Time-varying Functional Principal Component Analysis

Time-varying functional principal components analysis (TFPCA) is developed to model functional data that are repeatedly observed from each of many subjects at multiple occasions, often their

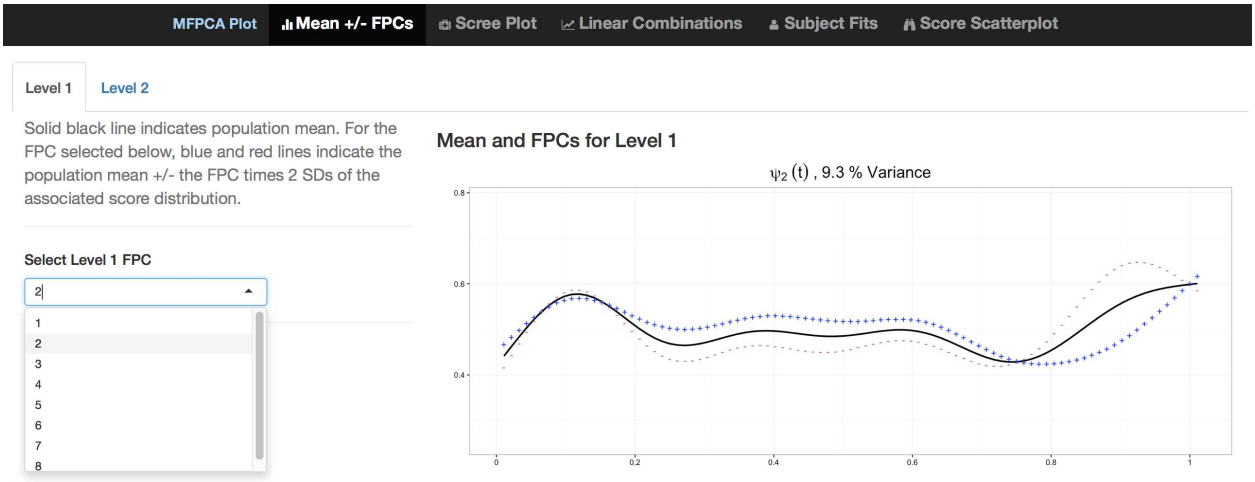

Figure 2.2: Screenshot showing tab 1 of the interactive graphic for MFPCA. The plot at right shows $\widehat{\mu}(t) \pm \sqrt{\widehat{\lambda}_{k_L}^{(L)}}\widehat{\psi}_{k_L}^{(L)}(t)$; k_L is chosen by the drop-down menu in at left, and the user can switch between level L by clicking *Level 1* or *Level 2* inset tabs at the top left.

clinical visits. In contrast to MFPCA, TFPCA accounts for actual time of visit T_{ij} at which a functional object $Y_{ij}(\cdot)$ is recorded, which allows us to study the dynamic behavior of the underlying true process, and make a time-specific prediction a trajectory at an unobserved visit time [Park and Staicu, 2015]. Other available modeling methods for longitudinal functional data that incorporate actual visit times T_{ij} include Greven *et al.* [2010] and Chen and Müller [2012].

2.4.1 TFPCA Model

TFPCA [Park and Staicu, 2015] models longitudinal functional data in two steps; first it uses FPCA to extract low dimensional features of the data and then it studies the dynamic behavior by modeling estimated FPC loadings obtained from the first step. The TFPCA model is given as follows:

$$Y_{ij}(t) = \mu(t, T_{ij}) + \sum_{k=1}^{K} c_{ik}(T_{ij})\psi_k(t) + \epsilon_{ij}(t), \tag{2.3}$$

where $\mu(t, T_{ij})$ is the population mean function that is assumed to be smooth over t and T_{ij}, $\psi_k(t)$ are eigenfunctions that are invariant to visit time T_{ij}, $c_{ik}(T_{ij})$ are corresponding FPC loadings

with mean zero and variance λ_k, and $\epsilon_{ij}(t)$ are residual curves. Scores $c_{ik}(t_{ij})$ are uncorrelated over i but correlated over j. To estimate the components of the TFPCA model, we first estimate the mean surface $\mu(t, T_{ij})$ using a bivariate smoother. Given this mean, we perform a "marginal" FPCA, estimating both basis functions and curve-specific scores, and then model the longitudinal dynamics of estimated scores $\widehat{c}_{ik}(T_{ij})$ over observation times T_{ij} using linear random effects models or FPCA. By modeling these longitudinal dynamics, the time-varying coefficient function $c_{ik}(\cdot)$ can be used to predict scores at any longitudinal time T and, as a result, to predict the full response trajectory $Y_i(\cdot, T)$.

2.4.2 Graphics for TFPCA

TFPCA is implemented in the `fpca.lfda()` function in the `refund` package. In Section 2.4.1, we have used t to denote the functional argument for consistency with the rest of the paper; however to maintain consistency with the notations used in Park and Staicu [2015], `plot_shiny()` function for TFPCA uses s to denote the functional argument and T to denote the longitudinal time.

Graphics for TFPCA are implemented by the code below:

```
> MS <- subset(DTI, case ==1)
> index.na <- which(is.na(MS$cca))
> Y <- MS$cca; Y[index.na] <- fpca.sc(Y)$Yhat[index.na]
> id <- MS$ID
> visit.index <- MS$visit
> visit.time <- MS$visit.time/max(MS$visit.time)
> fit.tfpca <- fpca.lfda(Y = Y, subject.index = id,
+                        visit.index = visit.index, obsT = visit.time,
+                        LongiModel.method = 'lme')
> plot_shiny(fit.tfpca)
```

The code produces an interface with two tabs. Tab 1 shows exploratory plots and includes three inset sub-tabs. The first sub-tab plots the observed curves for any user-selected subject, and includes options to display the observed curves of all subjects in the background and to display the

estimated pointwise mean curve, denoted by $m(t)$. The second sub-tab allows the user to see the longitudinal changes of the observed curves for a user-selected subject i; a slider bar animates the subject's visit times and highlights the corresponding observed curve in the plot. The last sub-tab shows two plots of the actual visit times T_{ij}: the bottom plot presents static histogram of visit times of all subjects, while the top plot presents all of observed visit times on a horizontal line to help visualize the sparsity of the longitudinal sampling.

Tab 2 shows estimated model components and predictions, and includes 8 inset sub-tabs. Sub-tabs 1 and 2 present static images of the estimated mean surface $\widehat{\mu}(t, T)$ and estimated marginal covariance $\widehat{\Sigma}(t, t')$. Sub-tabs 3, 4, and 5 illustrate the first step of estimation, and plot estimates of eigenfunctions $\widehat{\psi}_k(t)$, $m(t) \pm 2\sqrt{\widehat{\lambda}_k}\widehat{\psi}_k(t)$, and static scree plots of the estimated eigenvalues $\widehat{\lambda}_k$, respectively. Sub-tab 6 shows the estimated covariance of the time-varying FPC loadings $c_{ik}(\cdot)$ for user specified k. Sub-tab 7 shows the prediction of the time-varying basis coefficient $c_{ik}(T)$ for any user-selected subject i and component k; it also has an option of displaying predicted values of $c_{ik}(T)$ for all subjects in the background. Lastly, sub-tab 8 shows the prediction of a full response trajectory $Y_i(\cdot, T)$ for user-selected subject i in animation with change of values across 21 equi-spaced grid of points of T in the range of observed visit times of all subjects.

2.5 Function-on-Scalar Regression

In many cases, a length p vector of scalar covariates $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]$ is observed in addition to the function $Y_i(t)$. In these situations, it is often of interest to model the conditional expectation of the functional response as it depends on the scalar predictors; indeed, this problem has been the focus of a large literature [Brumback and Rice, 1998; Guo, 2002; Morris *et al.*, 2003; Morris and Carroll, 2006; Reiss *et al.*, 2010; Scheipl *et al.*, 2015; Goldsmith and Kitago, 2015; Goldsmith *et al.*, 2015].

2.5.1 FoSR Model

The most common function-on-scalar regression model is

$$Y_i(t) = \beta_0(t) + \sum_{k=1}^{p} x_{ik}\beta_k(t) + \epsilon_i(t) \tag{2.4}$$

where the $\beta_k(t)$ are fixed effects associated with scalar covariates and the $\epsilon_i(t)$ are residual curves. The coefficients $\beta_k(t)$ are interpreted analogously to coefficients in a (non-functional) multiple linear regression – as the expected change in response for each one unit change in the predictor – with the exception that they, like the outcome, are defined over t. Many estimation and inferential strategies are available for model (2.4); a popular approach is to expand coefficients $\beta_k(t)$ using a spline basis, which allows one to recast (2.4) as a traditional linear regression model and focus estimation on a vector of unknown spline coefficients. Our example uses the `bayes_fosr()` function in the `refund` package, which uses a rich cubic B-spline basis and estimates spline coefficients in a Bayesian framework with priors specified to enforce smoothness in the resulting coefficient functions. Both a Gibbs sampler and a computationally efficient variational approximation are available in `refund`.

2.5.2 Graphics for FoSR

Graphics for FoSR are implemented by the code below:

```
> DTI = DTI[complete.cases(DTI),]
> fit.fosr = bayes_fosr(cca ~ pasat + sex, data = DTI)
> plot_shiny(fit.fosr)
```

This code produces a interface with four tabs, each showing plots associated with model 2.4. The first tab is a plot of the observed data with the option to color curves by a user-selected covariate; this builds intuition analogously to scatterplots for non-functional regression. The second tab shows $\widehat{\beta}_0(t) + \sum_{k=1}^{p} x_k\widehat{\beta}_k(t)$, where values of x_k can bet set by slider bars for continuous covariates or

drop-down menus for categorical covariates; adjusting the sliders or drop-down menus shows the estimated conditional expectation for a specified predictor vector. The third tab, illustrated in Figure 2.3, shows estimated coefficient functions $\widehat{\beta}_k(t)$ with pointwise confidence intervals for the covariate x_k selected in a drop-down menu. The fourth tab is a plot of the residual curves $\widehat{\epsilon}_i(t)$ and allows for identification of median and outlying curves by band depth [Lopez-Pintado and Romo, 2009; Sun and Genton, 2011; Sun *et al.*, 2012]; the user can also choose to 'rainbowize by depth', which colors the curves from the median outward based on depth.

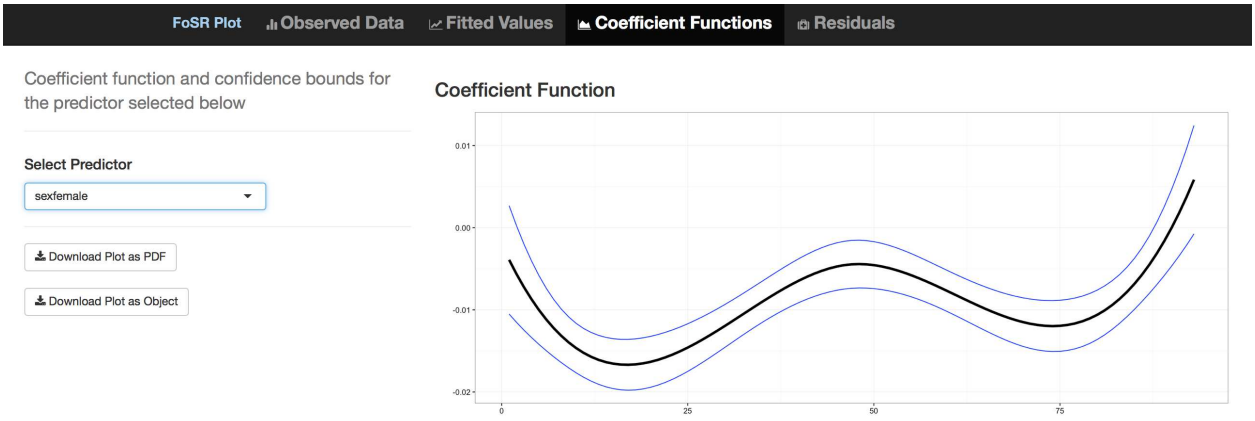

Figure 2.3: Screenshot showing tab 3 of the interactive graphic for FoSR. The plot shows the estimated coefficient function $\widehat{\beta}_k(t)$ for the selected covariate x_k with pointwise confidence intervals.

2.6 Code Structure of the `refund.shiny` Package

We now briefly describe the code infrastructure used to create the `refund.shiny` package.

As indicated in the introduction, the workflow separates visualization from analysis using the following workflow. First, one analyzes a dataset using a function in the `refund` package. The functions in `refund` take discretely observed functional data as input, perform an analysis, and return an object whose class corresponds to the method used. For example, the `fpca.sc` function return as object of class `fpca` and the `bayes.fosr` function returns an object of class `fosr`. The

primary function in `refund.shiny`, `plot_shiny`, is a generic function whose behavior depends on the class of the object passed as an argument. Because of this structure, the user experience is uniform across a variety of analyses; this also suggests a development strategy for the addition of interactive graphics as new analysis techniques become available. Lastly, by separating the analysis and visualization steps, it is possible for analysis functions developed outside of the `refund` package to return objects of a defined class and thereby take advantage of the plotting capabilities we describe.

The interactive graphics in the `refund.shiny` are built on RStudio's `R` package `shiny` [RStudio Inc., 2015], which significantly reduces the barriers to producing webpage-style representations of analysis results in `R`. Other examples of interactive graphics that utilize the `shiny` framework are `shinyMethyl` [Fortin *et al.*, 2014] for visualization of high-dimensional genomic data and `shinystan` [Stan Development Team, 2015] for exploring Bayesian models fit using Markov Chain Monte Carlo. In `refund.shiny` the plots within tabs are produced using `ggplot2` [Wickham and Chang, 2015]; it is possible to export each plot as a PDF or to save the corresponding `ggplot` object to the user's `R` workspace for further manipulation.

2.7 Concluding Remarks

Visualization has long been acknowledged as a central tool in data analysis. For functional datasets, the need for useful graphics is compounded: data are inherently complex, high-dimensional and structured. Although a robust literature for functional data exists and many methods have standard graphical representations, the creation of these graphics is often time consuming. The `refund.shiny` package was developed to ease this process by producing a visualization framework for several common functional data analyses. By leveraging new tools for interactivity, `refund.shiny` responds to user input and actions and, in so doing, can build intuition for analyses

in both statisticians and practitioners. The interfaces produced by `refund.shiny` using the `shiny` framework are web applications, rendered locally by a web browser. These applications can be hosted publicly and may, in the spirit of "visuanimations" [Genton *et al.*, 2015], be included as important parts of scientific papers and reports.

We use an analytic workflow that separates modeling from visualization. Doing so allows several methods and implementations to take advantage of the same visualization software; as an example, `fpca.sc()`, `fpca.face()`, `fpca.ssvd()`, and `fpca2s()` implement different methods for FPCA but are all compatible with `plot_shiny()`. This produces an intuitive user experience and leaves open the possibility for future approaches to FPCA or FoSR to use the `refund.shiny` package for visualization with minimal effort. Similarly, this workflow is amenable to the development of interactive visualizations for additional functional data analyses in future iterations of the package.

Chapter 3

Registration for exponential family functional data

3.1 Introduction

In the most common setting for functional data analysis, the basic unit of observation is the real-valued curve $Y_i(t)$ for subjects $i \in 1, \ldots, N$. More recently, there has been interest in exponential family functional data, where $Y_i(t)$ comes from a non-Gaussian distribution; it is typically assumed that $Y_i(t)$ has a smooth and continuous latent mean, $\mu_i(t) = E[Y_i(t)]$. Our motivation is the study of activity and inactivity using data collected with accelerometers, a setting with binary functional data. Figure 3.1 shows binary curves $Y_i(t)$ for two participants taking the value 1 when the participant is active and 0 when the participant is inactive. A solid curve shows an estimate of the smooth latent mean $\mu_i(t)$, interpreted as the probability the subject will be active at each minute in the 24 hours of observation. Other recent examples of non-Gaussian functional data include agricultural studies on the feeding behavior of pigs, spectral backscatter from long range infrared light detection, and longitudinal studies of drug use [Gertheiss *et al.*, 2015; Serban *et al.*, 2013; Huang *et al.*, 2014a].

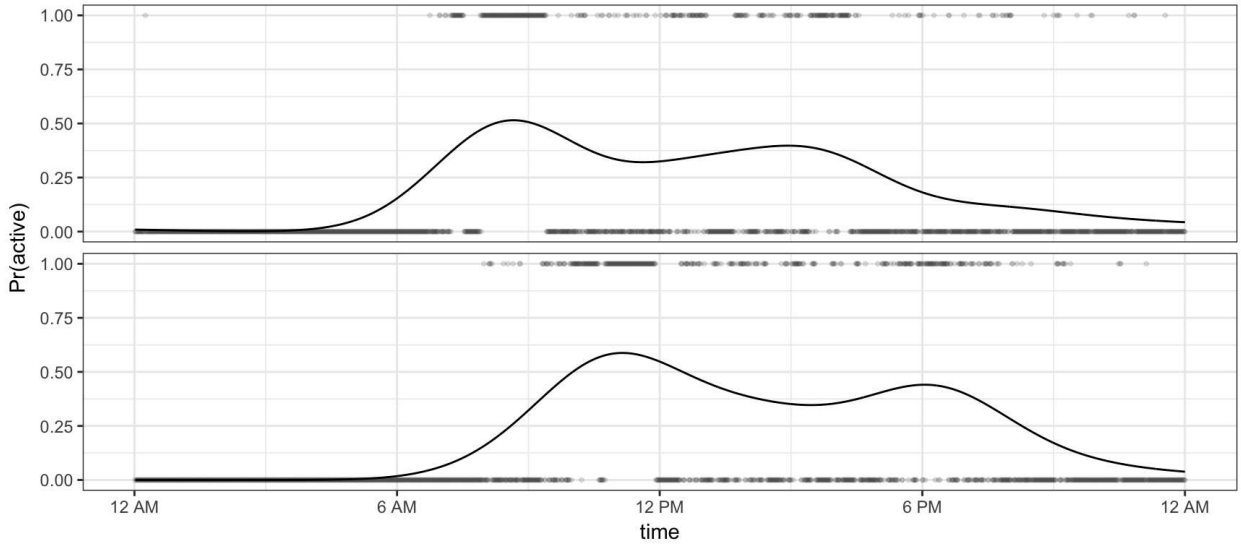

Figure 3.1: Points are binary curves for two subjects from the BLSA data before registration, where values of 1 and 0 represent activity and inactivity, respectively. The solid curves are estimates of the latent probability of activity, $\mu_i(t)$, and are fit for each subject using kernel smoothers.

Functional data often include both phase displacement, the misalignment of major features shared across curves, and amplitude variability. The process underlying phase variation may itself be of interest; additionally, when the interest is primarily in the amplitude variation, phase variation can artificially distort analyses of amplitude and mask the shared data structure. Methods for curve *registration*, which transform functional data to align features, are focused on addressing the problem of phase variation. The goal of registration is to warp the functional domain, which we will refer to as *time*, so that phase variation is minimized and the major features of the curves are aligned. This process necessitates a distinction between *chronological time* (t_i^*), which is the originally observed time for each subject, and *internal time* (t), which is the unobserved time on which major features are aligned across subjects (chronological and internal time are often referred to in the functional data literature as clock and system time, respectively). Stated differently,

internal time is the true but unknown time over which aligned curves are generated and chronological time is the shifted time on which misaligned curves are observed. The registration problem amounts to recovering the subject-specific warping functions $h_i : t \mapsto t_i^*$ which map internal time to chronological time. Inverse warping functions $h_i^{-1}(t_i^*)$ can then be used to obtain aligned curves $Y_i(t)$ from observed data $Y_i(t_i^*)$. To emphasize the conceptual difference between chronological and internal times, we index t_i^* by subject but do not index t.

We are interested in registering actigraphy data that comes from the Baltimore Longitudinal Study of Aging. The BLSA is an observational study of healthy aging and included an accelerometer for monitoring activity [Schrack *et al.*, 2014]. Our dataset includes 592 people, for whom accelerometer observations are gathered over 24 hours in one-minute epochs giving chronological times on equally spaced grids of length 1440. We are especially interested in activity and inactivity, defined using a threshold of raw accelerometer observations, as both low activity levels and excessive sedentary behavior have been associated with poor health outcomes. Moreover, there is a growing research interest in understanding temporal/diurnal patterns of accumulation of sedentary time [Diaz *et al.*, 2017; Martin *et al.*, 2014]. However, those analyses typically report diurnal averages that ignore the differences between subject specific wake time and mix together amplitude and phase.

The left panel of Figure 3.2 shows observed binary curves against chronological time. In this plot subjects appear in rows, with active and inactive minutes shown in dark and light shades, respectively. This figure clearly shows the variability in the timing of inactivity across subjects, who may start or end the day at different times, and may accrue inactive minutes in sedentary bouts at different times. Such misalignment attenuates the diurnal patterns of activity that we believe to be present based on the naturally occurring circadian rhythm. The right panel of Figure 3.2 shows estimates of the unregistered mean $\mu_i(t_i^*)$ obtained using a Gaussian kernel smoother;

these smooths illustrate the phase misalignment across subjects. The shift in timing of activity and inactivity is also seen in Figure 3.1. Specifically, the subject in the top row wakes up early, has a peak of activity, and then has a low activity level for the rest of the day, while the subject below has a similar but shifted pattern of behavior.

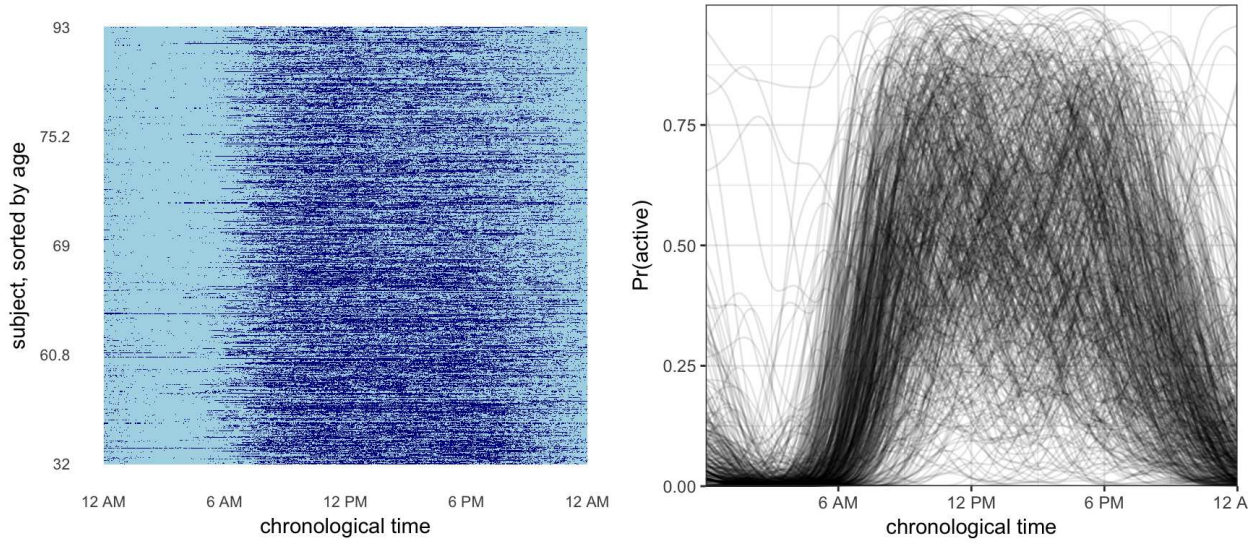

Figure 3.2: Plots of the unregistered data for 592 subjects at all 1440 minutes observed. At left is a lasagna plot, where row is the binary curve for a single subject and inactive and active observations are colored in light and dark shades, respectively. The rows are sorted by age, so that youngest subjects are at the bottom of the plot and oldest subjects are at the top. At right are smoothed curves for each subject, fit using kernel smoothers.

We propose novel methods for the registration of exponential family functional data, with emphasis on binary curves. Due to data size computational efficiency is critical, and we take this into consideration at each step of our method development. Section 3.2 provides a review of relevant literature on registration and exponential-family functional principal components analysis; Section 3.3 details our methods; Section 3.4 shows simulation results, and Section 3.5 applies our method

to the BLSA data. We conclude with a discussion in Section 3.6.

3.2 Literature Review

Our method draws on two distinct bodies of work in functional data analysis, which we review below. First, in 3.2.1, we review curve registration; this literature is primarily focused on Gaussian curves, with relatively little existing work for non-Gaussian curves. Then, in 3.2.2, we give an overview of exponential family FPCA, which is itself a relatively new area of interest in functional data analysis.

3.2.1 Registration

Several approaches for registering functional data have been proposed; we review these briefly, and suggest Marron *et al.* [2015] for a more detailed overview. Early approaches include dynamic time warping and landmark registration; for some time, however, template registration methods have been preferred. *Template registration* aligns each curve to a template curve by optimizing an objective function. This approach necessitates choosing the template, the objective function, and the optimization approach.

A common approach to template registration uses functional principal component analysis (FPCA) to select the template [Kneip and Ramsay, 2008]. First these methods estimate the template, and then estimate the warping functions for a given template; these steps are iterated until convergence. Warping functions are estimated using a sum of squared errors approach, often penalized to enforce smoothness. There is a large registration literature operating under and expanding this framework, including Sangalli *et al.* [2010] and Hadjipantelis *et al.* [2015]. Intuitively, functional principal components describe the main directions of variation in a set of curves, making FPCA a natural tool for identifying the features to which data is registered.

Srivastava *et al.* [2011] introduce a metric for calculating warping functions based on the Fisher-Rao distance. They calculate a Karcher mean template and define a square root slope function transform (SRSF) of the observed curves. Minimizing the $\mathbb{L}^2$ norm between two SRSFs is equivalent to minimizing their Fisher-Rao distance. Since the SRSF uses the derivative of the observed curve, the data to be registered are required to be smooth. The Fisher-Rao metric has been the basis for several recent approaches to registration, some which compute parameter values using dynamic programming [Srivastava *et al.*, 2011; Wu and Srivastava, 2014], and others which use Riemannian optimization [Huang *et al.*, 2014b]. Many of the SRSF-based approaches are implemented in the `fdasrvf` package [Tucker, 2017].

Although most work in registration has focused on continuous data, there are two recent exceptions. Wu and Srivastava [2014] apply the SRSF approach to binary functional data by presmoothing data with a Gaussian kernel and registering the resulting smooth curves. Panaretos and Zemel [2016] present a theoretical framework for separation of amplitude and phase variation of random point processes. The authors formalize a set of regularity conditions for warping functions that includes smoothness, proximity to the identity map, and unbiasedness, and establish a set of nonparametric estimators. However, since these estimators register the unobserved probabilities of the point processes, the authors also begin by smoothing binary curves using kernel density estimation.

In contrast to previous literature on registration we develop an approach that can be applied to continuous and discrete data and does not require presmoothing. We also emphasize computational efficiency, an important matter given our high-dimensional data application.

3.2.2 FPCA for exponential family curves

Functional principal components analysis is popular for identifying modes of variation in functional data. The most common approaches to FPCA decompose the variance-covariance matrix of demeaned functional observations; see Yao *et al.* [2005] or Goldsmith *et al.* [2013] for details on this approach. Hall *et al.* [2008] adapted the methods in Yao *et al.* [2005] for binary functional data by positing a smooth latent Gaussian process and then estimating and decomposing the covariance of this process. Serban *et al.* [2013] refined and extended this approach by improving approximations in the estimation procedure, increasing accuracy for rare events, and allowing spatial structures. However, as demonstrated in Gertheiss *et al.* [2017], the adaptation of Yao *et al.* [2005] to exponential family data has an inherent bias due to reliance on a marginal rather than conditional mean estimate.

Probabilistic FPCA is an appealing alternative to the covariance smoothing approach. This framework conceptualizes PCA as a likelihood-based model, can be approached from a Bayesian perspective, and easily accounts for sparse or irregular data. Tipping and Bishop [1999] introduce probabilistic PCA, and a related approach is used by James *et al.* [2000] for functional data. van der Linde [2008] extends probabilistic FPCA to binary and count data through a Taylor-approximated likelihood function, while Goldsmith *et al.* [2015] uses a fully Bayesian parameter specification for generalized FPCA and function-on-scalar regression. Because these approaches often relate the expected value of observed data to a smooth latent process through a link function, they are referred to as methods for generalized FPCA or GFPCA. Because all parameters are estimated simultaneously rather than sequentially, the probabilistic framework avoids the bias inherent in the covariance decomposition approach.

Our contributions to this literature focus on improving accuracy and efficiency for binary FPCA

by estimating parameters in a probabilistic framework using a novel variational EM algorithm. To do this, we adapt the approach developed by Jaakkola and Jordan [1997] for logistic regression, which has since been extended to (non-functional) binary PCA [Tipping, 1999] and multi-level PCA [Yue, 2016]. These methods rely on a variational approximation to the Bernoulli likelihood that is a true lower bound and allows for closed form updates of parameters. In contrast to van der Linde [2008], which uses a second-order Taylor expansion of the log likelihood to approximate a lower bound to the true distribution, our variational approximation is a true lower bound. While our method is optimized for binary data, similar derivations are possible for functional data from other exponential family distributions.

Consistent with this literature modeling exponential family curves, we assume a latent Gaussian process (LGP) generative model for our exponential family functional data. The LGP model assumes an unobserved smooth mean curve that serves as a "functional" natural parameter for the corresponding exponential family and from which the observed exponential family functional data is stochastically generated. In the case of binary data, the latent process is an unobserved smooth probability curve.

3.3 Methods

We first introduce the conceptual framework for our approach. Our goal is to estimate inverse warping functions h_i^{-1} which map unregistered *chronological time* t_i^* to registered *internal time* t such that $h_i^{-1}(t_i^*) = t$. Then for subject i, the unregistered and registered response curves are $Y_i(t_i^*)$ and $Y_i(t) = Y_i\left\{h_i^{-1}(t_i^*)\right\}$, respectively. Without loss of generality, we assume both t^* and t are on $[0, 1]$. We require that functions h_i^{-1} are monotonically increasing and satisfy the endpoint constraints $h_i^{-1}(0) = 0$ and $h_i^{-1}(1) = 1$. Notationally, we combine warping functions with

exponential family GFPCA through the following:

$$\begin{aligned} E\left[Y_i\left\{h_i^{-1}(t_i^*)\right\}|c_i, h_i^{-1}\right] &= \mu_i(t) \\ g\left\{\mu_i(t)\right\} &= \alpha(t) + \sum_{k=1}^{K} c_{ik}\psi_k(t). \end{aligned} \tag{3.1}$$

The aligned response curves $Y_i\left\{h_i^{-1}(t_i^*)\right\}$ for each $t_i^* \in [0,1]$ arise from the canonical exponential family of distributions with density

$$P\left[Y_i\left\{h_i^{-1}(t_i^*)\right\}|\mu_i(t)\right] = \exp\left\{\left(Y_i\{h_i^{-1}(t_i^*)\}g\{\mu_i(t)\} - b\left[g\{\mu_i(t)\}\right]\right)/\varphi + c[Y_i\{h_i^{-1}(t_i^*)\}, \varphi]\right\} \tag{3.2}$$

where $E\left[Y_i\{h_i^{-1}(t_i^*)\}|\mu_i(t)\right] = \mu_i(t) = b'\left[g\{\mu_i(t)\}\right]$, $Var\left[Y_i\{h_i^{-1}(t_i^*)\}|\mu_i(t)\right] = b''\left[g\{\mu_i(t)\}\right]\varphi$, and φ is the dispersion parameter. The subject-specific means $\mu_i(t)$ implicitly condition on parameters in model (3.1) and are used as templates in our warping step. Through link function g, the $\mu_i(t)$ are related to a linear predictor containing the population level mean $\alpha(t)$ and a linear combination of population level basis functions $\psi(t)$ and subject-specific score vectors $\boldsymbol{c}_i \sim N(0, \boldsymbol{I}_{K\times K})$. This formulation assumes that registered curves can be decomposed using GFPCA and, in doing so, places both registration and GFPCA in a single model.

Our estimation method is based on model (3.1) and alternates between the following steps:

1. Subject-specific means $\mu_i(t)$ are estimated via probabilistic GFPCA, conditional on the current estimate of inverse warping functions $h_i^{-1}(t_i^*)$.

2. Inverse warping functions h_i^{-1} are estimated by maximizing the log likelihood of the exponential family distribution under monotonicity and endpoint constraints on h_i, conditional on the current estimate of μ_i.

We iterate between steps (1) and (2) until curves are aligned.

Similar registration approaches for continuous-valued response curves have used the squared error loss for optimizing warping functions which, in a Gaussian setting, is equivalent to maximizing

the likelihood function. However, our likelihood-based approach, which registers non-Gaussian data by extending the exponential-family framework, is novel. In contrast to registration methods for discrete functional data, we register observed binary curves using smooth templates rather than aligning pre-smoothed functional data. Because our application has 592 subjects measured at 1440 time points each, computational efficiency is critical. To this end we develop a novel fast approach to binary FPCA in Step 1, which we describe in Section 3.3.1, and optimize speed in estimating warping functions in Step 2, which we describe in Section 3.3.2.

3.3.1 Binary FPCA

We first detail our novel EM approach to binary FPCA. Model (3.1) provides a conceptual framework, assuming that each curve $Y_i(t)$ is evaluated over internal time $t \in [0,1]$. In practice, data for subject i is observed on the discrete grid, $\boldsymbol{t}_i = \{t_{i1}, \ldots, t_{iD_i}\}$, which may be irregular across subjects, and therefore (in contrast to t) is indexed by subject. Functions indexed by the vector $\boldsymbol{t}_i$ are $D_i \times 1$ vectors of those functions evaluated on the observed time points (e.g. $Y_i(\boldsymbol{t}_i) = [Y_i(t_{i1}), \ldots, Y_i(t_{iD_i})]^T$ and $\psi_k(\boldsymbol{t}_i) = [\psi_k(t_{i1}), \ldots, \psi_k(t_{iD_i})]^T$). The population level mean $\alpha(t)$ and principal components $\psi_k(t)$, $1 \leq k \leq K$, are expanded using a fixed B-spline basis, $\boldsymbol{\Theta}_\phi(t)$, of K_ϕ basis functions $\theta_1(t), \ldots, \theta_{K_\phi}(t)$. Let $\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)$ be the $D_i \times K_\phi$ B-spline matrix evaluated at $\boldsymbol{t}_i$ and a $1 \times K_\phi$ vector when evaluated at a single point t_{ij}; then $\alpha(\boldsymbol{t}_i) = \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\boldsymbol{\alpha}_\Theta$ and $\Psi(\boldsymbol{t}_i) = [\psi_1(\boldsymbol{t}_i), \ldots, \psi_K(\boldsymbol{t}_i)] = \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\boldsymbol{\Psi}_\Theta$ where the vector $\boldsymbol{\alpha}_\Theta$ and matrix $\boldsymbol{\Psi}_\Theta$ of size $K_\phi \times K$ contain the spline coefficients for the mean and principal components, respectively. Observed on the discrete grid $\boldsymbol{t}_i$, the linear predictor in (3.1) becomes

$$g\{\mu_i(\boldsymbol{t}_i)\} = \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i\right). \tag{3.3}$$

We estimate parameters in model (3.3) using an EM algorithm that incorporates a variational approximation. We assume $\boldsymbol{c}_i \sim MVN(0, I)$. For the binary case that is our main interest, $g(\cdot)$ is

the logit function, for each point on the grid for the i^{th} subject, $Y_i(t_{ij}) \sim Bernoulli(\mu_i(t_{ij}))$ where $\mu_i(t_{ij}) = P(Y_i(t_{ij}) = 1|\boldsymbol{c}_i)$. It is convenient to rewrite the probability density function as

$$P\left\{Y_i(t_{ij})|\boldsymbol{c}_i\right\} = g^{-1}\Big[\{2Y_i(t_{ij}) - 1\}\{\boldsymbol{\Theta}_\phi(t_{ij})\left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i\right)\}\Big], \tag{3.4}$$

so that the full unobserved joint likelihood for the observations and score vectors is

$$L(\boldsymbol{Y}, \boldsymbol{c}) \propto \prod_{i=1}^{I}\prod_{j=1}^{D_i} g^{-1}\Big[\{2Y_i(t_{ij}) - 1\}\{\boldsymbol{\Theta}_\phi(t_{ij})\left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i\right)\}\Big] \times \prod_{i=1}^{I} \exp\Big(-\frac{\boldsymbol{c}_i^T\boldsymbol{c}_i}{2}\Big). \tag{3.5}$$

Let scalar $A_i(t_{ij}) = \boldsymbol{\Theta}_\phi(t_{ij})\left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i\right)$ and $\lambda(z) = \frac{0.5 - g^{-1}(z)}{2z}$. A variational approximation to (3.4), based on the approximation in Jaakkola and Jordan [1997], is

$$\begin{aligned}
\widetilde{P}\{Y_i(t_{ij})|\boldsymbol{c}_i, \xi_i(t_{ij})\} &= g^{-1}\{\xi_i(t_{ij})\} \\
&\times \exp\Big[\frac{\{2Y_i(t_{ij}) - 1\}\, A_i(t_{ij}) - \xi_i(t_{ij})}{2} \\
&+ \lambda\left\{\xi_i(t_{ij})\right\}\left\{A_i(t_{ij})^2 - \xi_i(t_{ij})^2\right\}\Big]
\end{aligned}$$

and is further discussed in Web Appendix A. The resulting variational joint likelihood is

$$\widetilde{L}(\boldsymbol{Y}, \boldsymbol{c}) \propto \prod_{i=1}^{I}\prod_{j=1}^{D_i} \widetilde{P}\Big\{Y_i(t_{ij})|\boldsymbol{c}_i, \xi_i(t_{ij})\Big\} \times \prod_{i=1}^{I} \exp\Big(-\frac{\boldsymbol{c}_i^T\boldsymbol{c}_i}{2}\Big). \tag{3.6}$$

We use an EM algorithm to obtain parameter estimates from (3.6) by (i) finding the posterior distribution of the scores; (ii) maximizing $\widetilde{L}(\boldsymbol{Y}, \boldsymbol{c})$ with respect to $\boldsymbol{\xi}$; and (iii) maximizing the variational likelihood with respect to $\boldsymbol{\alpha}_\Theta$ and $\boldsymbol{\Psi}_\Theta$. These three steps are described in Sections 3.3.1.1, 3.3.1.2, and 3.3.1.3; more details and simulations comparing to other GFPCA methods are given in the Appendix. A solution is attained when the squared difference between parameter estimates and their previous solution become arbitrarily small.

3.3.1.1 Calculating posterior scores

The posterior scores for each subject, derived via Bayes' rule, follow a multivariate normal distribution $\boldsymbol{c}_i|Y_i(\boldsymbol{t}_i), \xi_i(\boldsymbol{t}_i) \sim MVN(\boldsymbol{m}_i, \boldsymbol{C}_i)$ with:

$$\boldsymbol{C}_i = \left(\boldsymbol{I}_{K\times K} - 2\boldsymbol{\Psi}_\Theta^T \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)^T diag[\lambda\{\xi_i(\boldsymbol{t}_i)\}]\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\boldsymbol{\Psi}_\Theta \right)^{-1}$$

and

$$\boldsymbol{m}_i = \boldsymbol{C}_i \left(\boldsymbol{\Psi}_\Theta^T \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)^T \{Y_i(\boldsymbol{t}_i) - \frac{1}{2}\} + 2\boldsymbol{\Psi}_\Theta^T \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)^T diag[\lambda\{\xi_i(\boldsymbol{t}_i)\}]\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\boldsymbol{\alpha}_\Theta \right)$$

where $\xi_i(\boldsymbol{t}_i)$ is a vector of length D_i and $diag[\lambda\{\xi_i(\boldsymbol{t}_i)\}]$ is a $D_i \times D_i$ diagonal matrix.

3.3.1.2 Maximizing $\widetilde{L}(\boldsymbol{Y}, \boldsymbol{c})$ with respect to $\boldsymbol{\xi}$

We maximize the variational likelihood with respect to ξ_i^2, obtaining

$$\begin{aligned}
\widehat{\xi}_i(t_{ij})^2 &= E_{\widetilde{P}_{post}} \left\{ A_i(t_{ij})^2 \right\} \\
&= \boldsymbol{\alpha}_\Theta^T \boldsymbol{\Theta}_\phi(t_{ij})^T \boldsymbol{\Theta}_\phi(t_{ij}) \boldsymbol{\alpha}_\Theta + 2\boldsymbol{\alpha}_\Theta^T \boldsymbol{\Theta}_\phi(t_{ij})^T \boldsymbol{\Theta}_\phi(t_{ij}) \boldsymbol{\Psi}_\Theta \boldsymbol{m}_i \\
&+ \operatorname{tr}\left\{ \boldsymbol{\Psi}_\Theta^T \boldsymbol{\Theta}_\phi(t_{ij})^T \boldsymbol{\Theta}_\phi(t_{ij}) \boldsymbol{\Psi}_\Theta \boldsymbol{C}_i \right\} + \boldsymbol{m}_i^T \boldsymbol{\Psi}_\Theta^T \boldsymbol{\Theta}_\phi(t_{ij})^T \boldsymbol{\Theta}_\phi(t_{ij}) \boldsymbol{\Psi}_\Theta \boldsymbol{m}_i
\end{aligned}$$

where the expectation is taken with respect to the posterior distribution $\widetilde{P}\{\boldsymbol{c}_i|Y_i(\boldsymbol{t}_i), \xi_i(\boldsymbol{t}_i)\}$, using estimates of $\boldsymbol{\alpha}_\Theta$ and $\boldsymbol{\Psi}_\Theta$ from the previous iteration.

3.3.1.3 Maximizing $\widetilde{L}(\boldsymbol{Y}, \boldsymbol{c})$ with respect to $\boldsymbol{\alpha}_\Theta$ and $\boldsymbol{\Psi}_\Theta$

In this step we jointly estimate vectors of spline coefficients, which distinguishes our approach from previous binary PCA techniques and which entails additional complexity in the derivation of updates. The introduction of the spline basis and associated coefficients lowers the dimensionality of the estimation problem and enforces smoothness of the resulting $\widehat{\mu}_i(t)$.

In order to obtain updates for our population-level basis coefficients, we introduce a new representation of the model which is mathematically equivalent to the parameterization in model (3.4)

and easier to maximize. Let $\boldsymbol{s}_i = (\boldsymbol{c}_i^T, 1)^T$ of dimension $(K+1) \times 1$ and $\boldsymbol{\Phi} = (\boldsymbol{\Psi}_\Theta^T, \boldsymbol{\alpha}_\Theta)^T$ of dimension $(K+1) \times K_\phi$, and $vec(\boldsymbol{\Phi})$ be a vectorized version of $\boldsymbol{\Phi}$ with dimension $K_\phi(K+1) \times 1$. We can rewrite $A_i(\boldsymbol{t}_i)$ as $A_i(\boldsymbol{t}_i) = \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i) = (\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i) \otimes \boldsymbol{s}_i^T) vec(\boldsymbol{\Phi})$, where $\otimes$ is the Kronecker product. Maximizing the variational log-likelihood in this reparameterized form gives updates $vec(\widehat{\boldsymbol{\Phi}}) = -\left(\sum_i 2\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)^T diag[\lambda\{\xi_i(\boldsymbol{t}_i)\}] \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i) \otimes \widehat{\boldsymbol{s}_i \boldsymbol{s}_i^T} \right)^{-1} \left[\sum_i \left\{ Y_i(\boldsymbol{t}_i) - \frac{1}{2} \right\}^T \left\{ \boldsymbol{\Theta}_\phi(\boldsymbol{t}_i) \otimes \widehat{\boldsymbol{s}}_i{}^T \right\} \right]$, where $\widehat{\boldsymbol{s}}_i = (\boldsymbol{m}_i^T, 1)^T$ and

$$\widehat{\boldsymbol{s}_i \boldsymbol{s}_i^T} = \begin{pmatrix} \boldsymbol{C}_i + \boldsymbol{m}_i \boldsymbol{m}_i^T & \boldsymbol{m}_i \\ \boldsymbol{m}_i^T & 1 \end{pmatrix}$$

.

The first K rows of $vec\widehat{\boldsymbol{\Phi}}$ are the K columns of $\widehat{\boldsymbol{\Psi}}_\Theta$, and the last K_ϕ rows are $\widehat{\boldsymbol{\mu}}_\Theta$.

3.3.2 Binary Registration

We now turn to the second step in our iterative algorithm, in which warping functions are estimated for each subject conditionally on the target function $\mu_i(t)$. Conceptually, our approach is to maximize the exponential family likelihood function given by integrating the density in equation (3.2) over time. We maximize with respect to the inverse warping function $h_i^{-1}(t_i^*)$, subject to the constraint that $h_i^{-1}(t_i^*)$ is monotonic with endpoints fixed at the minimum and maximum of our domain. For binary data we maximize the Bernoulli log-likelihood

$$l(h_i^{-1}; Y_i, \mu_i) = \int \left(Y_i(t_i^*) \log \mu_i \left\{ h_i^{-1}(t_i^*) \right\} + \{1 - Y_i(t_i^*)\} \log \left[1 - \mu_i \{ h_i^{-1}(t_i^*) \} \right] \right). \tag{3.7}$$

Again, functions are observed on a discrete grid in practice, and we differentiate between subject-specific finite grids for chronological time $\boldsymbol{t}_i^* = \{t_{i1}^*, ..., t_{iD_i}^*\}$ and internal time $\boldsymbol{t}_i = \{t_{i1}, ..., t_{iD_i}\}$. Using notation similar to Section 3.3.1, we let $Y_i(\boldsymbol{t}_i^*)$, $Y_i(\boldsymbol{t}_i)$, and $h_i^{-1}(\boldsymbol{t}_i^*)$ be $D_i \times 1$ vectors corresponding to observed responses, registered responses, and inverse warping functions, respectively. We expand $h_i^{-1}(\boldsymbol{t}_i^*)$ using a B-spline basis, $\boldsymbol{\Theta}_h(\boldsymbol{t}_i^*)$, of dimension $D_i \times K_h$ to take the form

$h_i^{-1}(\boldsymbol{t}_i^*) = \boldsymbol{\Theta}_h(\boldsymbol{t}_i^*)\boldsymbol{\beta}_i = \boldsymbol{t}_i$. The $K_h \times 1$ vector of spline coefficients $\boldsymbol{\beta}_i$ allows us to express $h_i^{-1}(\boldsymbol{t}_i^*)$, and is the target of our estimation problem. We estimate $\boldsymbol{\beta}_i$ separately for each subject using constrained optimization and loop over subjects.

We modify the conceptual likelihood in equation (3.7) to incorporate the spline basis expansion of h^{-1} and to express data over the observed finite grid, which yields

$$l\left\{\boldsymbol{\beta}_i; Y_i(\boldsymbol{t}_i^*), \mu_i(\cdot)\right\} \propto \sum_{j=1}^{D_i} \left(Y_i(t_{ij}^*) \log \mu_i\{\boldsymbol{\Theta}_h(t_{ij}^*)\boldsymbol{\beta}_i\} + \{1 - Y_i(t_{ij}^*)\} \log\left[1 - \mu_i\{\boldsymbol{\Theta}_h(t_{ij}^*)\boldsymbol{\beta}_i\}\right] \right). \quad (3.8)$$

Recall that $\mu_i(\cdot)$ from (3.3) is the subject-specific mean found in the FPCA step. Estimates are constrained to be monotonic with fixed endpoints. The constraints ensure that our resulting estimates for t are monotonic and span the desired domain. We implement these constraints using linear constraint matrices, which we provide in Appendix A. The constrained optimization can be made more efficient with an analytic form of the gradient. The gradient for the general exponential family case and for the Bernoulli loss in particular also appear in Appendix A.

3.3.3 Implementation

Our methods are implemented in `R` and are publicly available on GitHub as part of the `registr` package [Wrobel *et al.*, 2018]. For Step 1, binary FPCA is custom-written with a `C++` backend for estimation. For Step 2, we implement linearly constrained optimization with the `constrOptim()` function, which uses an adaptive barrier algorithm to minimize an objective function subject to linear inequality constraints. If an analytic gradient of the objective function is not provided, then Nelder-Mead optimization is used, otherwise BFGS, a gradient descent algorithm, is used. By implementing an analytic gradient we improve accuracy and computational efficiency of our estimation.

Though our simulated and real data examples are observed on a dense regular grid, the `registr` package handles both sparse and irregular functional data. For visualizing results, `registr` is

compatible with `refund.shiny`, an `R` package that produces interactive graphics for functional data analyses [Wrobel *et al.*, 2016].

3.4 Simulations

We assess the accuracy and computational efficiency of our method using data simulated to mimic our motivating study, and compare to competing approaches described below.

3.4.1 Simulation Design

Binary functions in simulated datasets are designed to exhibit a circadian rhythm, so that simulated participants are more likely to be inactive at the beginning and end of the domain ("day") and more likely to be active in the middle of the day. Overall activity levels vary across simulated participants, as do the timing of the active period. Participants exhibit two main active periods separated by a dip, which is consistent with the BLSA data.

We first generate a grid of chronological times $\boldsymbol{t}_i^*$, which is equally spaced and shared across subjects. We generate inverse warping functions $h_i^{-1}(\boldsymbol{t}_i^*)$ using a B-spline basis with 3 degrees of freedom; coefficients are chosen from a uniform(0,1) distribution and placed in increasing order to ensure monotonically increasing warping functions. The internal times $\boldsymbol{t}_i$ for each subject are obtained by evaluating the inverse warping functions at $\boldsymbol{t}_i^*$. We simulate latent probability curves over internal time, $\mu_i(\boldsymbol{t}_i)$, from the model

$$\begin{aligned} E\left\{Y_i(\boldsymbol{t}_i)|c_i\right\} &= \mu_i(\boldsymbol{t}_i) \\ g\left\{\mu_i(\boldsymbol{t}_i)\right\} &= \alpha(\boldsymbol{t}_i) + c_i \times \psi(\boldsymbol{t}_i) \end{aligned} \tag{3.9}$$

where $\alpha(\boldsymbol{t}_i)$ and $\psi(\boldsymbol{t}_i)$ are constructed using a B-spline basis and $c_i \overset{i.i.d}{\sim} N(0,1)$. For each $t_{ij} \in i, j = 1, \ldots, D_i$, binary observations $Y_i(t_{ij})$ are sampled independently over j from a Bernoulli distribution with $\mu(t_{ij})$.

Unregistered data $Y_i(\boldsymbol{t}^*)$, observed over the grid $\boldsymbol{t}_i^*$, are defined by the warping functions $h_i(\boldsymbol{t}_i)$. Figure 3.3 shows an example of a single simulated dataset, including latent probability curves on both $\boldsymbol{t}_i^*$ and $\boldsymbol{t}$ (first row, first and second columns) and observed binary data (second row, first and second columns).

We evaluate the performance of our algorithm as a function of sample size and grid length. We simulate 25 datasets for each combination of sample sizes (50, 100, and 200) and grid lengths (taking values 100, 200, 400). For each dataset we apply the methods in Section 5.2, denoted *registr* in text and figures below, setting $K_\phi = 9$, $K_h = 3$, and using 1 FPC.

To provide a frame of reference we compare our approach with two approaches based on the SRSF framework, both of which are implemented in the `time_warping()` function in the `fdasrvf` package [Tucker, 2017]. Both implementations use smoothed versions of the binary data but use different optimization methods. The first uses dynamic programming, which is the default optimization choice for the `fdasrvf` software, and is denoted *svrf-dp* in text and figures below. The second uses Riemannian optimization and is denoted *svrf-ro*. For both competing approaches, observed binary data was smoothed using a box filter, which is built into the `fdasrvf` software. The number of box filter passes is a tuning parameter that must be selected, and we found that the overall registration results were sensitive to this choice. We considered several values (25, 50, 100, 200, and 400); in the following we use 200 passes, which generally lead to better performance in our simulations.

Methods are compared in terms of estimation accuracy and computation time, with accuracy quantified using mean integrated squared error (MISE). For each subject, integrated squared error calculations are made comparing the estimated inverse warping functions for each method, $\widehat{h}_i^{-1}(\boldsymbol{t}_i^*)$, to the true inverse warping functions $h_i^{-1}(\boldsymbol{t}_i^*)$ such that $ISE = \int_0^1 \left\{ h_i^{-1}(\boldsymbol{t}_i^*) - \widehat{h}_i^{-1}(\boldsymbol{t}_i^*) \right\}^2 dt_i$. MISE is then the average of ISEs across subjects. A sensitivity analysis of our method's performance

across values of K_ϕ and K_h is given in the Appendix.

3.4.2 Simulation Results

Figure 3.3 shows a simulated dataset with 100 subjects observed over a grid with 200 time points. From left to right, columns show observed (unregistered) data; data observed on the true internal time t; and data aligned using the *registr* method, the *srvf-dp* method, and the *srvf-ro* method. The top row shows the latent mean curves, the middle row shows plots of observed binary data, and the bottom row shows inverse warping functions using true internal time t and estimated internal times $\widehat{t}_{registr}$, $\widehat{t}_{srvf-dp}$, and $\widehat{t}_{srvf-ro}$. The latent probability curves illustrate the structure of the simulated data and the relative magnitudes of phase and amplitude variability. Binary curves illustrate the observed data, and include two periods of higher activity for each subject.

The results for *registr* in this example are encouraging, both for the latent curves and for the binary activity data in that phase variation is largely removed. Some amount of misalignment remains, which is attributable to the inherent sampling variability introduced when binary points are generated from the latent probabilities. The *srvf-dp* method also works reasonably well, although visual inspection of the probability curves and binary data suggests somewhat poorer alignment. The *srvf-ro* method has poorest alignment, although it also performs reasonably well and captures the major features in the data.

Figure 3.4 summarizes results across simulated datasets at different sample sizes and grid lengths; for reference, the data in Figure 3.3 has a median MISE for the *registr* method relative to other datasets generated with 100 subjects and 200 time points. The columns of Figure 3.4, from left to right, show results for datasets with 50, 100, and 200 subjects, respectively, and grid lengths of 100, 200, and 400 are shown within each panel. The top row shows box plots of MISE and the bottom row shows median computation times. Across all settings, *registr* outperforms both

srvf-dp and *srvf-ro* methods in terms of the MISE; this is consistent with observations in Figure 3.3. With respect to computation time, although the methods are similar for small sample sizes and grid lengths, *registr* and *srvf-ro* scale as these increase, while the burden grows dramatically for *srvf-dp*.

3.5 Analysis

We now apply our method described in Section 5.2 to the BLSA data. These data contain 592 subjects with activity counts every minute over 24 hours, for a total of 1440 measurements per subject. BLSA participants wore the accelerometer for 5 days; we average across these days to establish a typical diurnal pattern for each participant, and then threshold the result at values of 10 counts per minute to obtain the binary activity curve to be registered. We fix the dimensions of the B-spline basis functions to $K_\phi = 8$ and $K_h = 4$ and number of FPCs to $K = 2$. Total computation time was 17 minutes. In the following, we discuss registered activity profiles using language that refers to times of day. However, it is important to remember throughout this Section that registered curves are observed on internal time rather than chronological time, and times of day are person-specific in that sense.

Figure 3.5 shows the registered curves from the BLSA dataset, which can be compared with the observed data in Figure 3.2. After registration, there are two clear activity peaks: people tend to be active for an extended period of time after they wake up; this period is followed by a mid-day dip in activity, and a second, smaller, period of activity in the afternoon and evening. Figure 3.6 emphasizes this point, and the effect of registration, by plotting the subjects from Figure 3.1 after registration. The data for these two subjects are more closely aligned, as are the latent probabilities curves estimated from the aligned data. The left panel of Figure 3.5 shows the inverse warping functions which transform the BLSA data from the unregistered to the registered space.

The results of the applying the registration method to these data are consistent with expectations, in that the diurnal activity pattern observed across subjects after registration contains both morning and afternoon active periods and a period of relative inactivity around lunchtime. These results also emphasize the importance of assessing and removing phase variability in studies of daily activity patterns. The existence and number of "chronotypes", or subjects who intrinsically prefer certain hours of the day (like the colloquial night owls or early birds), is the subject of intense debate in the circadian rhythm literature [Adan *et al.*, 2012]. Aligning observed activity data as a processing step may help inform this debate, and our results are consistent with the existence of distinct chronotypes in this population. The supplementary materials contain additional analysis results for the registration of data from each day of the week separately. These results are similar to those presented in this Section.

3.6 Discussion

We present a novel approach to curve registration for functional data from exponential family distributions which avoids the need for pre-smoothing, and our attention to computational efficiency is necessitated by our data. Simulations suggest our approach compares favorably to competing methods in the settings we examined. Our scientific results are plausible and meaningful in the context of activity measurement. Finally, our code for registration and binary probabilistic FPCA is publicly available in the `registr` package.

Our approach assumes exponential family functional data is generated from a latent Gaussian process. While this is a common modeling choice for binary functional data that works well in practice, it may not provide a suitable framework for theoretical considerations in which the grid size goes to infinity. Possible future work building on Descary and Panaretos [2016], which considers modeling continuous functional data with both low rank structure and local correlation in the

Gaussian setting, may provide a scientifically meaningful way forward but is beyond our current scope.

Though `registr` outperformed the SRSF-based approaches in our simulations, we expect that the SRSF method will be better suited to some cases including, potentially, smooth Gaussian curves. Indeed, when curves are absolutely continuous SRSF has the added theoretical benefits of translation and scale invariance and consistency of the warping procedure. For discrete data or noisy Gaussian data, where a smoothing parameter must be chosen before applying SRSF methods, it is unclear if any method will be uniformly superior and we recommend considering multiple approaches to registration.

Because of the nature of our application, we optimize performance for registering binary curves. While our method can be applied to functional data from any exponential family, one will not reap the computational benefits we highlight here without at least some additional work optimizing the FPCA algorithm for additional distributions; computationally efficient implementations for the Poisson distribution will be relevant for studies using accelerometer data. For our application, we chose to threshold activity count data and register the resulting binary curves, which aligned general patterns of activity and inactivity. Though we could have chosen to register the raw counts using a Poisson distribution, exploratory analyses suggested that aligning raw activity counts may be overly influenced by extreme values.

Though we focus on amplitude alignment for this paper, the inverse warping functions contain information on phase variation and are potential analysis objects for future scientific work. Subsequent analyses will examine whether aligned data are more clearly affected by covariates like age and sex, and how the phase alignment relates to these covariates. Finally, we note that our emphasis has been on the temporal structure of inactivity, and additional work to connect these results with the accrual of sedentary minutes in bouts is needed.

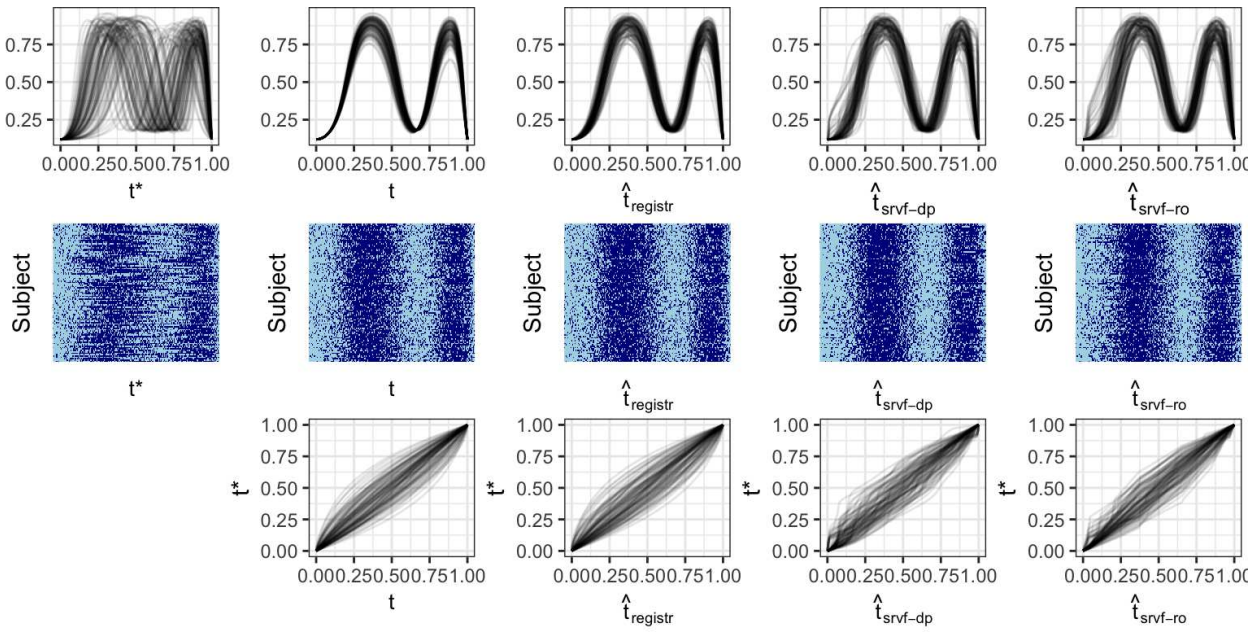

Figure 3.3: For top and center rows, from left to right we have: unregistered curves, curves registered using true inverse warping functions, curves registered using *registr* method, curves registered using *fdasrvf* method with dynamic programming optimization, curves registered using *fdasrvf* method with Riemannian optimization. The top row shows the true latent probability curves which are used to generate the binary curves but not used to estimate warping since they are unknown in a real data application. The middle row shows the binary curves as a heatmap-style plot, as in Figure 3.2. The bottom row shows the true, *registr* method, *fdasrvf* method with dynamic programming, and *fdasrvf* method with Riemannian optimization inverse warping functions.

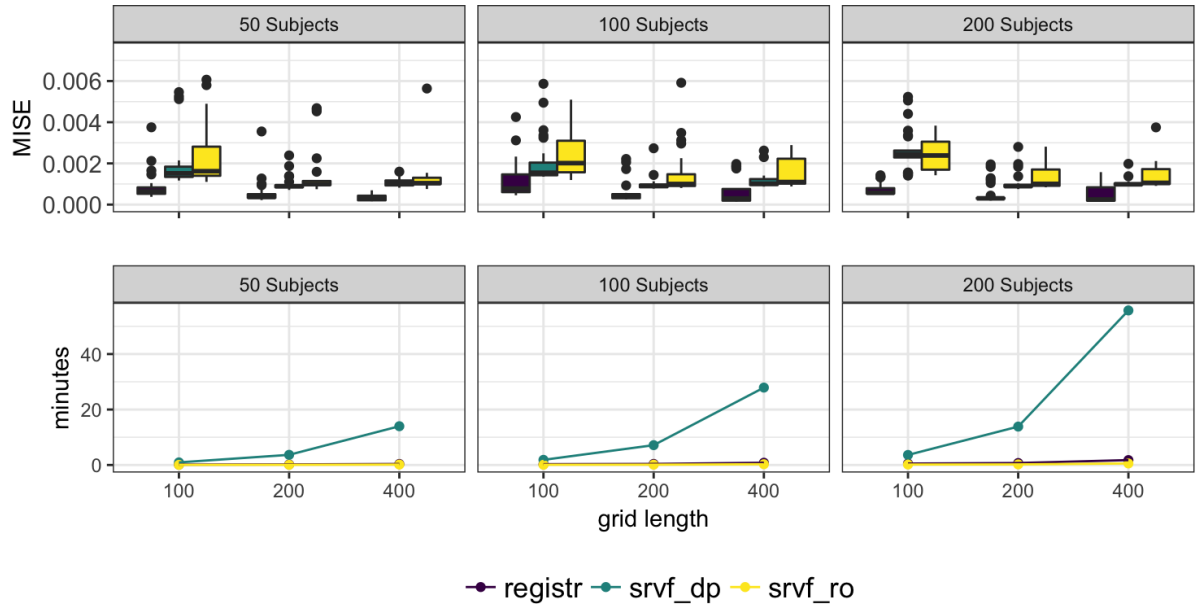

Figure 3.4: This figure shows mean integrated squared errors (top row) and median computation times (bottom row) for *registr* (darkest shade), *srvf-dp* (medium shade), and *srvf-ro* (lightest shade) methods across varying sample sizes and grid lengths. The columns, from left to right, show sample sizes 50, 100, and 200, respectively. Within each panel we compare grid lengths of 100, 200, and 400.

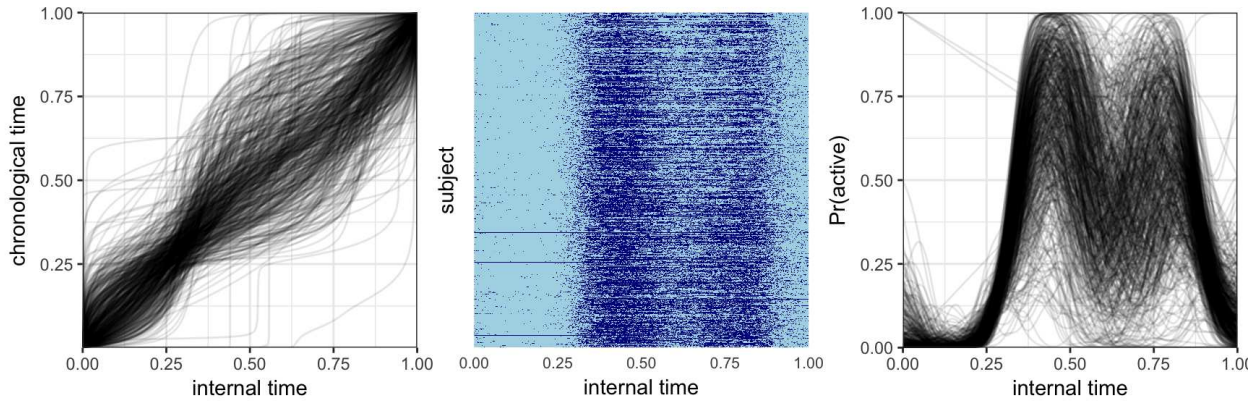

Figure 3.5: Plots of the registered BLSA data. Left panel shows inverse warping functions from alignment of the data; center panel shows a plot of the aligned binary data; and right panel shows smooths of the aligned data. See Figure 3.2 for the unregistered data.

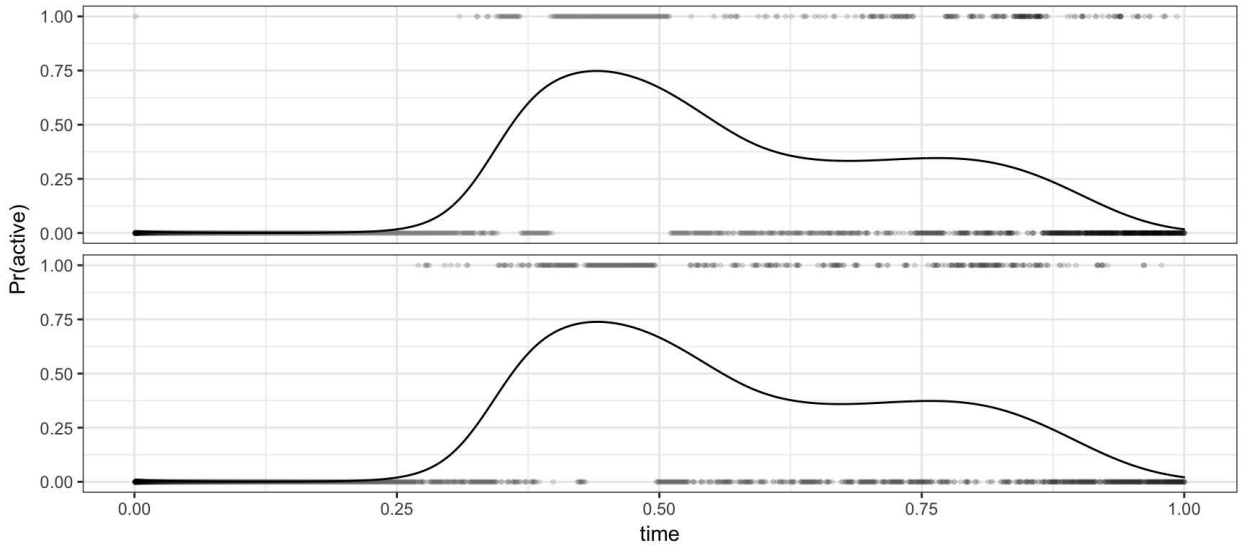

Figure 3.6: These are binary curves for the same two subjects from the BLSA data as in Figure 3.1 but now the curves are registered. Here the lines represent estimates of the latent probability that come from our binary FPCA algorithm.

Chapter 4

Intensity warping for multisite MRI harmonization

4.1 Introduction

Medical imaging has become an established practice in clinical studies and medical research, leading to situations where images must be compared across site locations, scanners, or scanner types. Upgrades in scanner technology within a site may render old data not comparable to data collected on a newer machine, and this presents challenges in studies where acquisition techniques change over time. Multisite studies have become common as well; examples include large neuroimaging studies such as the Alzheimer's Disease Neuroimaging Initiative [Mueller *et al.*, 2005] and the Human Connectome Project [Van Essen *et al.*, 2013], as well as targeted clinical trials studying multiple sclerosis (MS) interventions such as Kappos *et al.* [2006] and Hauser *et al.* [2017].

Measurement across multiple sites and scanners introduces unwanted technical variability in the images [Schnack *et al.*, 2004]. Going forward we will refer to technical artifacts introduced across either sites or scanners as "scanner effects." Scanner effects in imaging studies can reduce power to detect true differences across images and distort downstream measurements of regional

volumes, brain lesions, and other biological features of interest [Schnack *et al.*, 2010; Jovicich *et al.*, 2013; Cannon *et al.*, 2014; Keshavan *et al.*, 2016; Schwartz *et al.*, 2019]. In structural magnetic resonance imaging (MRI) studies, detection of scanner effects is particularly challenging because images are collected in arbitrary units of voxel intensity; as a result, raw MRI intensities are often not comparable across study visits even within the same subject and scanner. We refer to unwanted technical variability within the same scanner and subject that are due to arbitrary unit intensity values as "intensity unit effects." Though often conflated, intensity unit effects and scanner effects are distinct sources of unwanted technical variation and should be treated separately. We refer to methods intended to address intensity unit effects as "normalization" methods to distinguish them from methods intended to reduce scanner effects, which we term "harmonization" methods. Both scanner effects and intensity unit effects are present in multisite MRI studies, and in practice they can be challenging to separate.

Scanner effects can be due to differences in scanner hardware, scanner software, scan acquisition protocol, or other unknown sources. When present, images collected at different sites may have systematically different distributions of intensity values. For example, Shinohara *et al.* [2017] showed substantial differences in volumetrics across sites and scanner types even for a single, biologically stable subject measured under standardized protocols at the same field strength on platforms produced by the same vendor. The left panel of Figure 4.1 shows smoothed PDFs of intensity values for this single subject, who was scanned twice at each of seven sites across the U.S. Large scanner effects are evident; smaller but visible differences within site show that intensity unit effects are present as well. In subsequent analyses, scanner effects produced inconsistent measurements of MS lesion volume both when lesions were segmented manually or by a variety of automated software pipelines [Shinohara *et al.*, 2017] .

The issue of arbitrary units has long been recognized and is the subject of a large literature

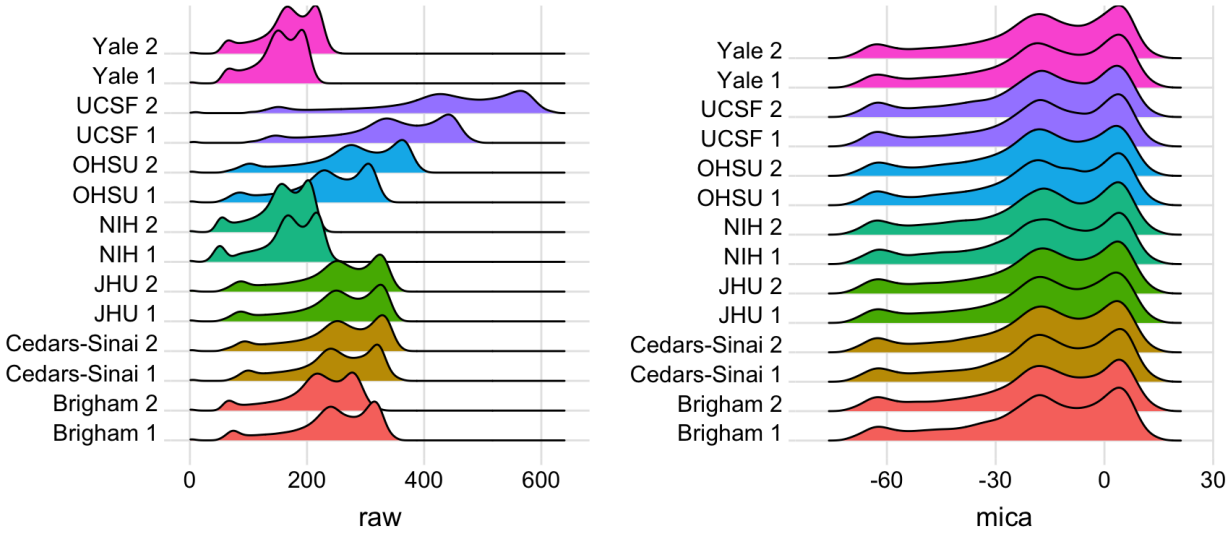

Figure 4.1: Smoothed PDFs of voxel intensities for scan-rescan data across seven sites in the NAIMS pilot study: Brigham and Women's Hospital (Brigham), Cedars-Sinai, Johns Hopkins University (JHU), National Institutes of Health (NIH), Oregon Health & Sciences University (OHSU), University of California San Francisco (UCSF), and Yale University (Yale). Left panel shows raw voxel intensities; right panel shows densities after *mica* harmonization and White Stripe normalization. At each site two scans were collected; a 1 or 2 after site name indicates the first or second scan, respectively.

on intensity normalization [Nyúl *et al.*, 1999; Shinohara *et al.*, 2011, 2014; Ghassemi *et al.*, 2015]. Intensity normalization methods facilitate comparability across subjects measured on the same scanner and standardize voxel intensity values; for a review of several methods see Shah *et al.* [2011]. Histogram matching is an early approach that aligns densities of voxel intensities to quantiles of an image template constructed from several control subjects. Though popular, histogram matching often fails to preserve biological characteristics of individual scans and removes useful information regarding variation among subjects. Shinohara *et al.* [2014] formalized the principles of image normalization and introduced the White Stripe method. White Stripe normalizes images using patches of normal appearing white matter (NAWM), so that rescaled intensity values are biologically

interpretable as units of NAWM. White Stripe can effectively normalize white matter across subjects and is a useful preprocessing step for automated lesion segmentation in MS [Sweeney *et al.*, 2013b,a; Valcarcel *et al.*, 2018], but technical variability can remain in the gray matter.

Unlike intensity normalization methods, which target intensity unit effects, harmonization methods aim to reduce scanner effects so that downstream analyses are more comparable across sites and scanners [Fortin *et al.*, 2017; Yu *et al.*, 2018]. Fortin *et al.* [2018] described a voxel-wise regression method, based on tools from genomics, that harmonizes cortical thickness measurements from MRI scans. This method succeeds in removing scanner effects for measurements extracted from each image; in contrast, our goal in the present study was to develop an effective harmonization method that can be applied to the entire brain. Similar tools from genomics are used to correct for scanner effects in multisite diffusion tensor imaging data [Fortin *et al.*, 2017] and multisite functional MRI data [Yu *et al.*, 2018]. However, these harmonization methods require spatial registration to a population template, which can lower image resolution and make it challenging to detect important disease features such as MS lesions. Ideally, an all-purpose harmonization method would remove scanner effects from the whole brain without requiring that all subjects be spatially registered to the same template image.

In the past, "normalization" has been used to simultaneously address the problems we characterize as unit and scanner effects, although these are more correctly viewed as distinct problems. As a result, intensity normalization techniques such as histogram matching and White Stripe are often used to address harmonization issues [Schnack *et al.*, 2004; Shinohara *et al.*, 2014; Fortin *et al.*, 2016]. Unlike harmonization techniques mentioned previously, these normalization techniques can be applied to the whole brain, do not require spatial registration, and reduce intensity unit effects. When scanner effects are due to the same voxel intensity transformations used to reduce unit effects, the normalization techniques will reduce scanner effects as well. However, often they

fail to reduce much of the variability across sites, especially when large nonlinear scanner effects are present. Additionally, histogram matching normalizes voxel intensities across images at the cost of removing biological variability across subjects which can distort structures and mask inter-subject differences of interest.

Here, we introduce a new image intensity harmonization framework for multisite studies. We use data in which a subject was scanned on multiple scanners closely enough in time that any image differences can be attributed to differences across acquisition platforms (scanner effects) rather than biological effects. Our objectives in this study were to (1) establish a method that removes scanner effects by leveraging multiple scans collected on the same subject, and, building on this, (2) develop a technique to estimate scanner effects in large multisite trials so these can be reduced with preprocessing steps. The first objective establishes a framework for understanding harmonization, and the second relates to the practical use of this framework in multisite studies. We propose **m**ultisite **i**mage harmonization by **C**DF **a**lignment (*mica*), which harmonizes images by aligning cumulative distribution functions (CDFs) of voxel intensities. Our approach estimates nonlinear, monotonically increasing transformations of the voxel intensity values in one scan such that the resulting intensity CDF perfectly matches the intensity CDF from a second ("target") scan. CDFs can be perfectly aligned using standard approaches to curve registration in the functional data analysis literature [Srivastava *et al.*, 2011; Tucker *et al.*, 2013; Wrobel *et al.*, 2018]. Although these intensity transformations, called warping functions, are defined using CDFs, they can be applied to voxel-level intensity values to produce a harmonized image. For a subject measured on different scanners in close succession, this allows us to identify and remove scanner effects; mappings established in this way can be used to reduce the impact of scanner effects in multisite studies.

We outline our harmonization approach using two data sets with distinct but related problems. The North American Imaging in Multiple Sclerosis (NAIMS) pilot study [Shinohara *et al.*, 2017;

Dworkin *et al.*, 2018; Oh *et al.*, 2018; Papinutto *et al.*, 2018; Schwartz *et al.*, 2019] found large scanner effects in a single subject with biologically stable MS, and we use these data to show that *mica* can reduce technical variability across sites while preserving the ability to detect MS lesions. A second study, which we refer to as the trio2prisma study, scanned ten healthy subjects on two different machines and found systematic nonlinear differences between the scanners. We used *mica* to harmonize images from the first scanner so that they are comparable to images collected on the second scanner; this demonstrates how our method can be used to create a mapping between scanners, and that scanner effects can be removed when data are available from both scanners for all subjects. Since scan-rescan data are often only available for a subset of study subjects, we also employed a leave-one-scan-out cross-validation approach to assess the utility of our harmonization method in this common setting. For both studies, we used *mica* to understand and, to the extent possible, remove scanner variability. We paired our method with White Stripe to remove intensity unit effects as well as scanner effects, though other intensity normalization methods could be used instead.

In the next section, we describe our data and the *mica* methodology. We then present the results of our technique in different settings, followed by a discussion.

4.2 Materials and Methods

4.2.1 Data and processing

4.2.1.1 NAIMS dataset

The NAIMS steering committee developed a brain MRI protocol relevant to MS lesion quantification [Shinohara *et al.*, 2017]. Using this protocol, two scans were collected at each of seven sites across the United States on a 45-year-old man with clinically stable relapsing-remitting MS. All scans

were performed on 3T Siemens scanners (four Skyra, two TimTrio, and one Verio). At each site, scan-rescan imaging was performed on the same day, with the subject exiting the machine between scans. The participant was also assessed at the beginning and end of the study on the same scanner to confirm disease stability by clinical and MRI measures .

Each image was bias-corrected using the N4 inhomogeneity correction algorithm [Tustison *et al.*, 2010], then brain extraction was performed using the FSL BET skull-stripping algorithm [Smith, 2002]. After performing *mica* harmonization as described in Section (4.2.2), T1-weighted (T1-w) and fluid attenuated inversion recovery (FLAIR) images were White Stripe normalized [Shinohara, R T and Muschelli, J, 2018] to remove intensity unit effects and enable automated MS lesion detection using the MIMoSA [Valcarcel *et al.*, 2018] software pipeline.

4.2.1.2 trio2prisma dataset

The trio2prisma data were collected from ten healthy subjects ages 19 to 29 at the University of Pennsylvania. For each subject, brain MRI scans were obtained on both a Siemens Trio machine and a Prisma scanner. Scans were performed between 2 and 11 days apart for each subject (mean 4.2 days), a time window in which we expect no significant structural changes in the brain. We focused on T1-w images for the trio2prisma data, though our method can be applied to other modalities as well. Images were bias-corrected, skull-stripped, and White Stripe normalized using the same algorithms described for the NAIMS data. Because normalization methods have often been used for harmonization in the past, we compared *mica* to White Stripe and histogram matching normalization. To assess method performance on this data, we compared white and gray matter segmentations for *mica*-harmonized images to White Stripe and histogram matching normalized images. All white and gray matter segmentations were obtained using multi-atlas Joint Label Fusion [Wang *et al.*, 2013].

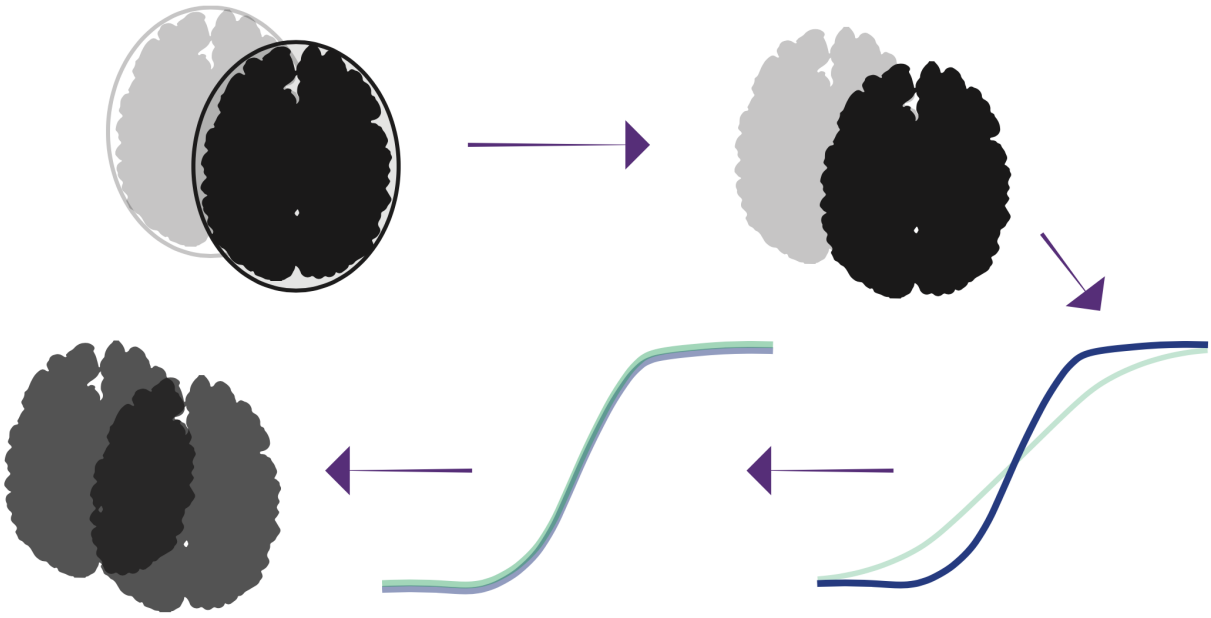

Figure 4.2: Harmonization pipeline. Raw images are N4 bias-corrected, skull-stripped, voxel intensities are converted to CDFs, CDFs are aligned by warping intensity values. The transformation of intensity values that produces this alignment is called a warping function, and the nonlinear transformation is applied to the raw images to produce harmonized images.

4.2.2 Methodology

Our framework for image harmonization uses non-linear transformations of image intensity values to remove scanner effects. The transformations were calculated by aligning distribution functions of intensity values. For a particular imaging modality (for example, T1-w), $Y_{ijk}(v)$ represents the intensity at a given voxel v for scan j of subject i measured at site k. Then $f_{ijk}(x)$ and $F_{ijk}(x)$ represent the probability density function (PDF) and CDF, respectively, for the voxel intensities of image Y_{ijk} measured over intensities x. Within each subject we assumed variability in voxel intensities across visits j and sites k is due to scanner and intensity unit effects rather than biological change, and that non-biological differences could be removed by aligning all CDFs for the i^{th} subject to a subject-specific "template CDF," $F_{it}(x)$, for template t; template choices for

our motivating studies are described below.

For image Y_{ijk} we estimate the nonlinear monotonic transformation of the intensity values, or *warping function*, $h_{ijk}^{-1}(x) = \widetilde{x}$, which aligns the CDF $F_{ijk}(x)$ to its template via

$$F_{ijk}\left\{h_{ijk}^{-1}(x)\right\} = F_{ijk}(\widetilde{x}) = F_{it}(x). \tag{4.1}$$

After alignment, the CDF of the original images becomes identical to the CDF of the template. For this reason, we use the notation $F_{it}(x)$ to represent the *mica*-harmonized CDF as well as the template for alignment. We further denote $f_{ijk}\left\{h_{ijk}^{-1}(x)\right\} = f_{it}(x)$ and $Y_{it}(v)$ to be the *mica*-harmonized PDFs and images, respectively. The aligned PDFs, $f_{it}(x)$, can be recovered from CDFs by differentiation. The warping functions $h_{ijk}^{-1}(x) = \widetilde{x}$ define a new intensity value, $\widetilde{x}$, for each original intensity value in x. Since each $Y_{ijk}(v)$ is a voxel intensity in x, harmonized images Y_{it} take values in $\widetilde{x}$ and are obtained by $h_{ijk}^{-1}\{Y_{ijk}(v)\} = Y_{it}(v)$. Figure (4.2) shows a schematic of this process: images were bias corrected and skull-stripped, voxel-intensities were converted to CDFs, CDFs were aligned, and warping functions from CDF alignment were used to generate harmonized images.

Given this framework for quantifying scanner effects, we now address objectives (1) and (2) stated in Section (4.1). Our first objective, to establish a method that removes scanner effects, is illustrated using both the NAIMS and the trio2prisma data. For NAIMS data, we obtained empirical CDFs of T1-w and FLAIR images from the NAIMS dataset. Within an imaging modality, each CDF is given by $F_{ijk}(x), i = 1, j \in \{1, 2\}, k \in \{1, ..., 7\}$. We used the Karcher mean as the common template $F_{it}(x)$ to which all CDFs within a modality are aligned, though in principle other templates could be used. For the trio2prisma data, we obtained empirical CDFs of T1-w images. Each CDF is given by $F_{ijk}(x), i \in \{1, ..., 10\}, j = 1, k \in \{\text{Trio, Prisma}\}$. For each subject, we used the CDF from the Prisma image, $F_{i\text{Prisma}}(x)$, as the template to which we align the CDF from the

Trio image, $F_{i\text{Trio}}(x)$. Functions from the `fdasrvf R` package [Tucker, 2017] were used to perform alignment.

Our second objective was to develop a technique to estimate scanner effects in large multisite trials; to illustrate this, we used warping functions from the trio2prisma data. In such studies, most subjects are only measured on a single scanner. At best, only a subset of subjects will have scans collected at all locations in the study. In order to harmonize scans for all subjects in this real-world setting, we propose to use *mica* to estimate warping functions for the subset of subjects who have multiple scans, average these warping functions across subjects; and use the resulting mean to harmonize images for subjects with only a single scan available. We assessed the performance of this approach using leave-one-scan-out cross validation in the trio2prisma data. Specifically, we removed the Prisma scan for one subject and computed the *mica* warping functions $\{h_i^{-1}(x)\}$ for the remaining subjects. We then computed the pointwise mean of these warping functions; using this as the warping function for the removed subject, we obtained a predicted Prisma scan from the known Trio scan. This process was repeated for each of the ten subjects. In the subsequent sections, scans harmonized using this leave-one-scan-out (*loso*) approach will be referred to below as *loso*-harmonized images and Trio scans harmonized using the full data will be referred to as *mica*-harmonized scans.

4.2.3 Statistical performance

All analyses were performed in the R software environment.

4.2.3.1 NAIMS data

To assess the performance of our method on the NAIMS data we quantified T2-hyperintense lesion volume from the 3D FLAIR and T1-w images in both the White Stripe normalized and

mica-harmonized images using MIMoSA [Valcarcel *et al.*, 2018] for automated lesion segmentation. Because the number and volume of lesions are important metrics for monitoring MS disease progression [Bakshi *et al.*, 2008] and the evaluation of therapeutic efficacy [Filippi *et al.*, 2006], eliminating non-biological variability in detected lesion volumes will help clinicians deliver the best possible care to their patients.

We quantified mean and variance of lesion volumes within and across sites after applying White Stripe alone and after applying *mica* followed by White Stripe.

4.2.3.2 trio2prisma data

For the trio2prisma data, we compared *mica* and *loso* to the histogram matching algorithm proposed by Nyúl *et al.* [1999], as implemented in Fortin *et al.* [2016]. For better performance we first removed background voxels before running the histogram matching algorithm. To quantify performance of the methods we computed Hellinger distance of images before and after normalization, both within and across subjects. The Hellinger distance operates on PDFs of intensities, and its square is given by

$$h^2(f_l, f_k) = \frac{1}{2} \int \left(\sqrt{f_l(x)} - \sqrt{f_k(x)} \right)^2 dx \tag{4.2}$$

for PDFs $f_l(x)$ and $f_k(x)$. We visualized CDFs and PDFs and calculated Hellinger distances (Figures 4.4, 4.5, and 4.6, respectively) using images that had been *mica* or *loso*-harmonized but not yet White Stripe normalized. This is to isolate and visualize the effects of our method. For downstream analyses, including automated white and gray matter segmentation, we applied White Stripe normalization to the *mica* and *loso*-harmonized images to remove any residual intensity unit effects. We then estimated gray and white matter volumes and compare these across harmonization methods.

4.3 Results

For the NAIMS pilot data, we compared White Stripe normalized images to images processed using the *mica* approach outlined in section (4.2.2). For the trio2prisma data, we compared four harmonization strategies: no harmonization, histogram matching, *mica*, and *loso*. The main findings from these comparisons are summarized in the following two sections.

4.3.1 *mica* reduces variation in lesion volumes across sites in the NAIMS study

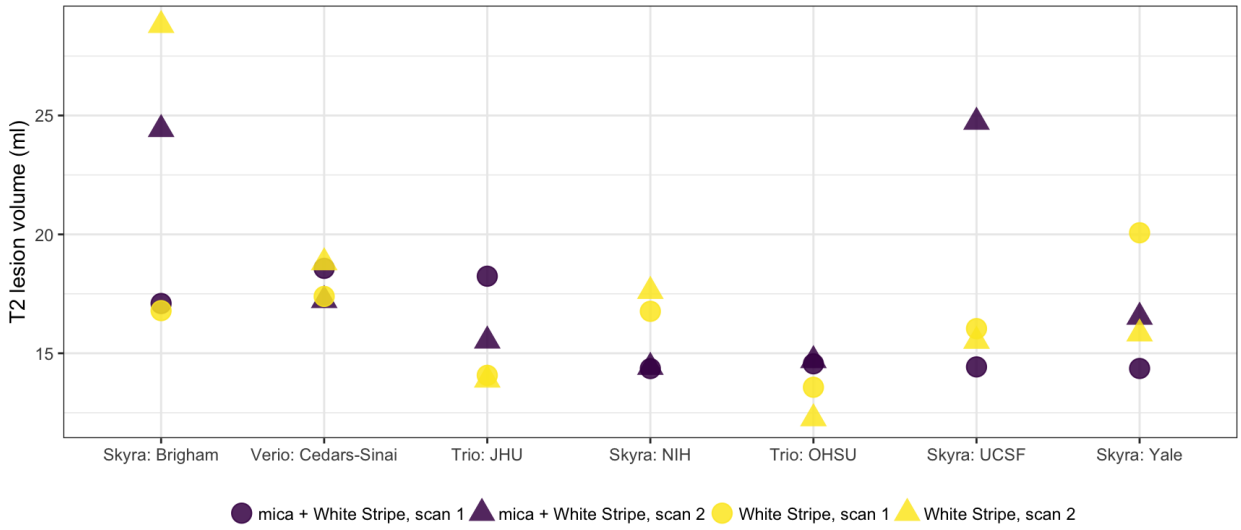

Figure 4.3: Estimated T2 lesion volumes for scan-rescan pairs at each of 7 sites in the NAIMS study. Circles indicate scan 1 and triangles indicate scan 2. Light and dark colors are volumes for White Stripe normalized images and *mica* normalized images, respectively.

We *mica*-harmonized then White Stripe normalized the NAIMS scans, and then quantified MS lesion volume to assess the effect of scanner variability on a common downstream analysis before and after *mica* harmonization. The left panel of Figure 4.1 shows PDFs of raw voxel intensities from the NAIMS study images, and the right panel shows PDFs of images that have been *mica*-harmonized then White Stripe normalized. The raw PDFs show small differences within site, which

are attributable to intensity unit effects, and larger differences across site, which are attributable to scanner effects. Scanner effects are particularly large between the UCSF site and other sites. After *mica* harmonization, the images across and within site have the same distributions of voxel intensities.

Figure 4.3 shows estimated T2-hyperintense lesion volume across sites for both White Stripe alone and White Stripe in conjunction with *mica* for scan-rescan pairs across the seven NAIMS sites. Compared to White Stripe alone, *mica* in conjunction with White Stripe yielded less variable lesion volume measurements across sites (variance 11.8 ml^2 vs. 17.1 ml^2) and similar lesion volume measurements within sites (variance 12.4 ml^2 vs. 11.9 ml^2). We see a larger impact across sites than within sites, suggesting that our method decreases site-to-site variance as expected and, together with White Stripe, performs comparably to existing methods for within site variance.

4.3.2 *mica* preserves variation across subjects in the trio2prisma study

An appropriate harmonization method for multisite studies should reduce variability across scanners within the same subject but preserve biological differences across subjects. Here, we evaluate results from the trio2prisma data with these goals in mind. We compared *mica* and *loso*-harmonized images to images processed by histogram matching.

Figures 4.4 and 4.5 show CDFs and PDFs, respectively, under different harmonization scenarios. Visual inspection of intensity PDFs and CDFs in untransformed images suggests differences across scanners: the Prisma scans tend to have lower intensity values and higher peaks than the Trio scans. For both *mica* and histogram matching, within-subject technical variability is reduced because PDFs of Trio scans and Prisma scans are aligned. *mica* accomplishes this by mapping the Trio scan to the original Prisma scan, thus preserving the original features of the Prisma scans including variability across subjects. Histogram matching must be applied to scans from both

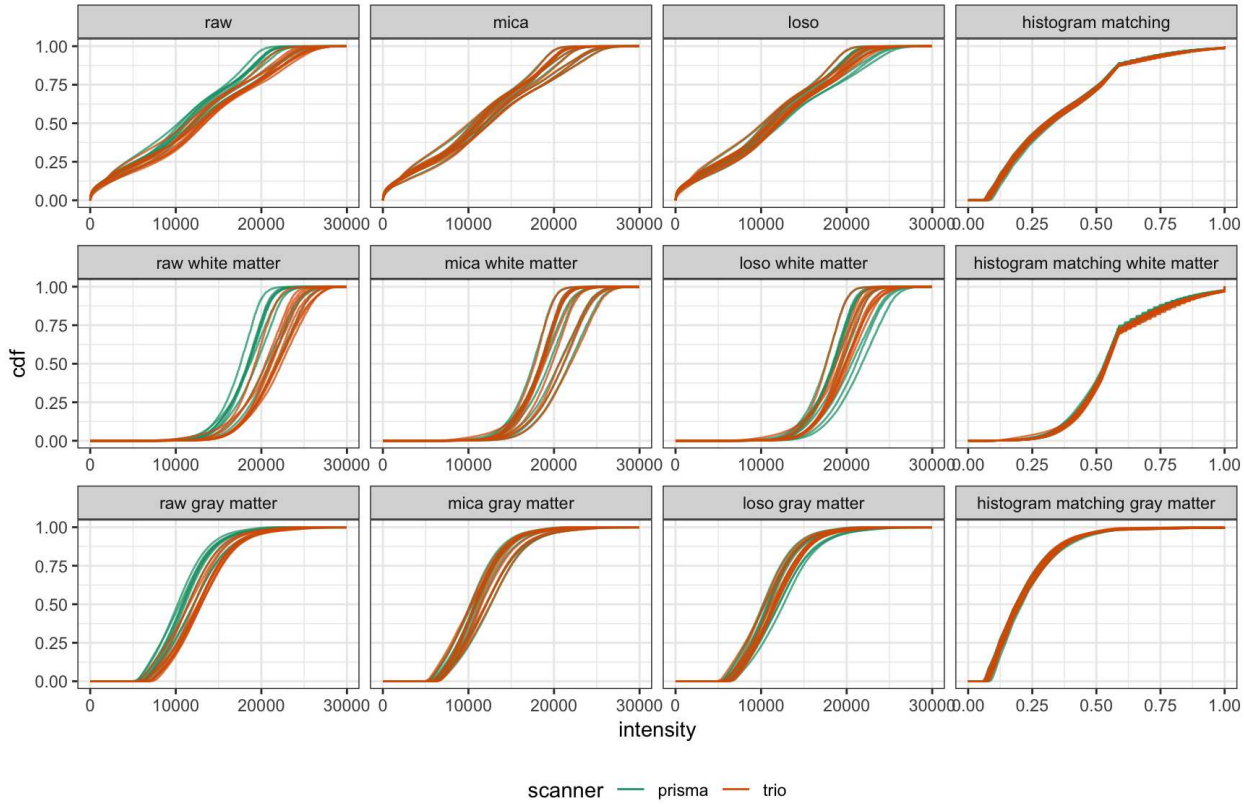

Figure 4.4: CDFs of intensities before and after harmonization by tissue type in the trio2prisma study. Rows indicate tissue type, with whole brain, white matter, and gray matter shown in rows 1, 2, and 3, respectively. Columns correspond to different harmonization methods.

the Trio and Prisma scanners, and reduces within-subject variability at the expense of eliminating desired differences across subjects. *loso* provides reasonable harmonization in that it maps Trio scans into the same range of intensity values as Prisma scans, but has less accuracy in reducing within-subject variability than *mica* or histogram matching. However, much of the desired across-subject variability is retained.

We quantified the variability across subjects using the Hellinger distance from equation (4.2) on PDFs of voxel intensities. Figure 4.6 displays boxplots of these pairwise distances for the original Trio scans, original Prisma scans, and scans processed by histogram matching, *loso*, and *mica*. The

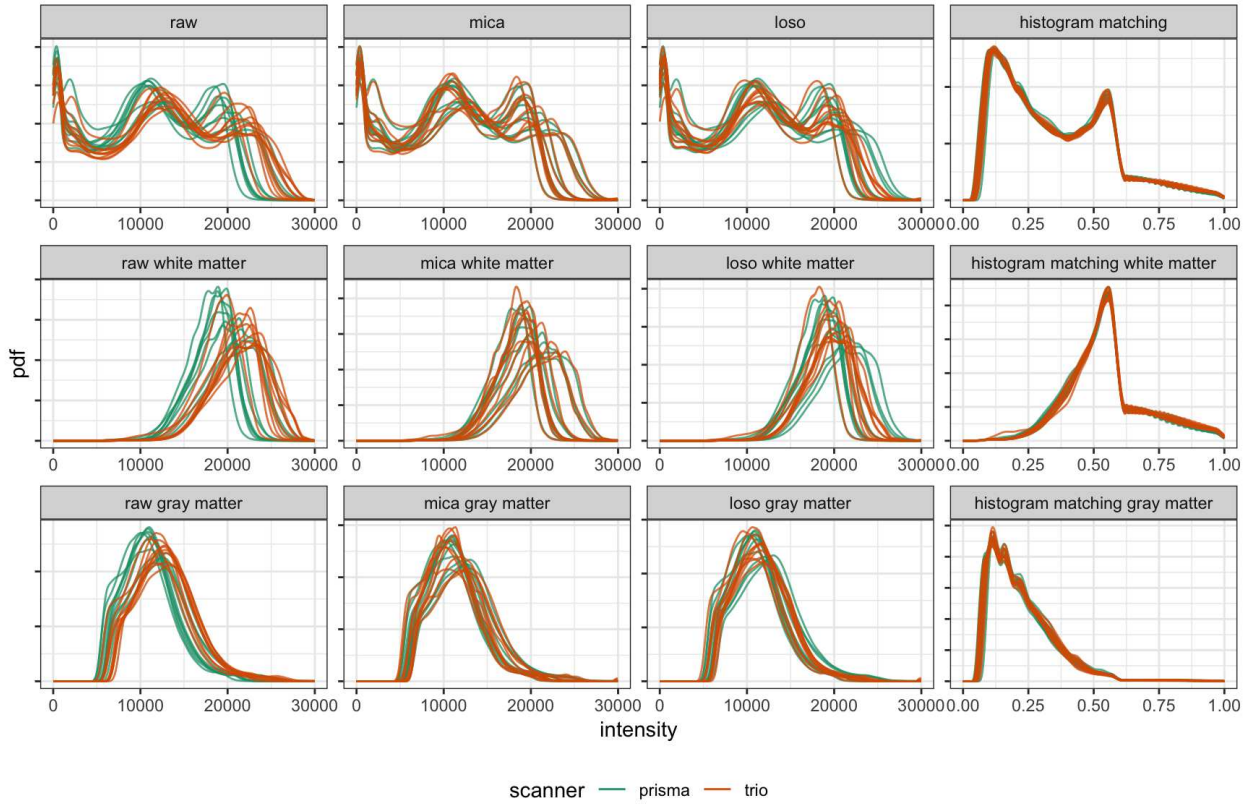

Figure 4.5: Smoothed PDFs of intensities before and after harmonization by tissue type in the trio2prisma study. Rows indicate tissue type, with whole brain, white matter, and gray matter shown in rows 1, 2, and 3, respectively. Columns correspond to different harmonization methods.

figure is divided into distances calculated on the full skull-stripped images (left column), white matter (middle column), and gray matter (right column). The *mica*-harmonized Trio scans have similar across-subject variability to the Prisma scans. The *loso* scans have variability comparable to the original Trio scans but smaller than the Prisma scans. Histogram matching virtually eliminates inter-subject variability, including that which is presumably biological.

Figure 4.7 shows an axial slice of the Trio image for one subject from the trio2prisma dataset. The slice is shown for raw intensity values (center), intensity values after *mica* harmonization (left), and intensity values after histogram matching (right). Here, *mica*-harmonization brightens

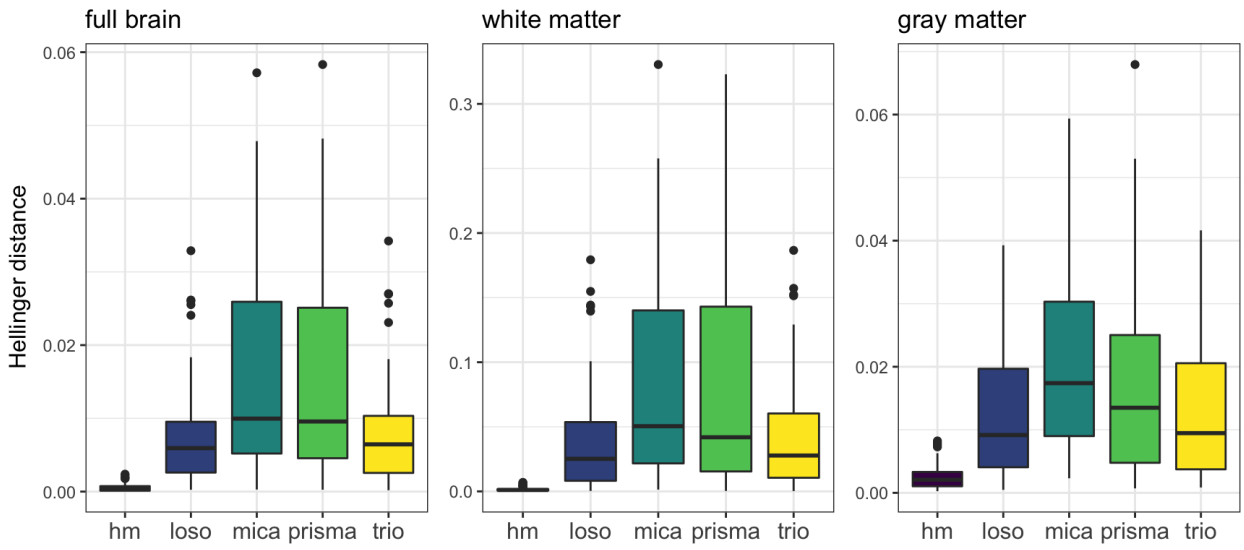

Figure 4.6: Boxplots of Hellinger distances across subjects, shaded by method. Columns show results for full brain (left), white matter (middle), and gray matter (right).

the contrast between white and gray matter but does not distort the shape of biological features in the tissue. Histogram matching, however, drastically changes the appearance of the image, converting some gray matter to CSF and some white matter to gray matter.

Finally, neither harmonization nor normalization methods should bias assignment of tissue type. After harmonization or normalization, we expect that segmentation volumes from harmonized Trio scans should be similar to segmentation volumes from unharmonized and unnormalized (raw) Trio scans. We estimated white and gray matter volumes on original Trio scans and after histogram matching, White Stripe, *mica* followed by White Stripe, and *loso* followed by White Stripe. Figure 4.8 shows these volumes for each subject and tissue type. All methods have at least some difference in segmentation volume compared to the raw data. The *mica*, *loso*, and White Stripe methods all performed similarly, with volumes that are close to those of the raw images but slightly lower for the gray matter and slightly higher for the white matter. Histogram matching, however, had much lower segmentation volumes in both the gray matter and the white matter than either the raw data

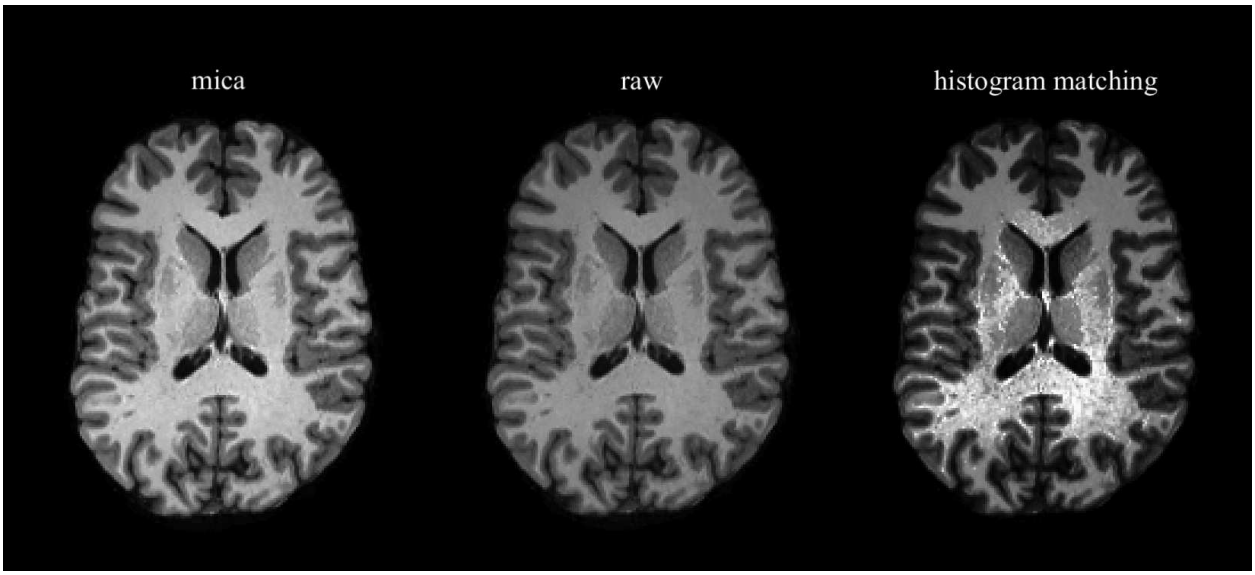

Figure 4.7: Axial slice of skull-stripped images from a single subject in the trio2prisma dataset. Center panel shows the raw intensity values from an image collected on the Trio scanner. Left and right panels show the same image after *mica* harmonization and histogram matching, respectively.

or any other method. As shown in Figure 4.7, histogram matching severely distorts the image; we believe this distortion causes the segmentation algorithm to convert some gray matter to CSF and some white matter to gray matter, which explains the consistently lower volumes.

4.4 Discussion

Unwanted technical variability due to scanner effects in multisite clinical trials and observational studies is an increasingly common problem; to mitigate these scanner effects we introdce *mica*, a method that harmonizes structural MRI images by defining nonlinear transformations between CDFs of voxel intensities. To specifically target scanner effects, we developed a paradigm for understanding scanner effects and intensity unit effects as related but distinct sources of technical variability in MRI scans. Intensity unit effects are due to arbitrary MRI unit intensities within a single scanner, and scanner effects are unwanted technical artifacts introduced across scanners or

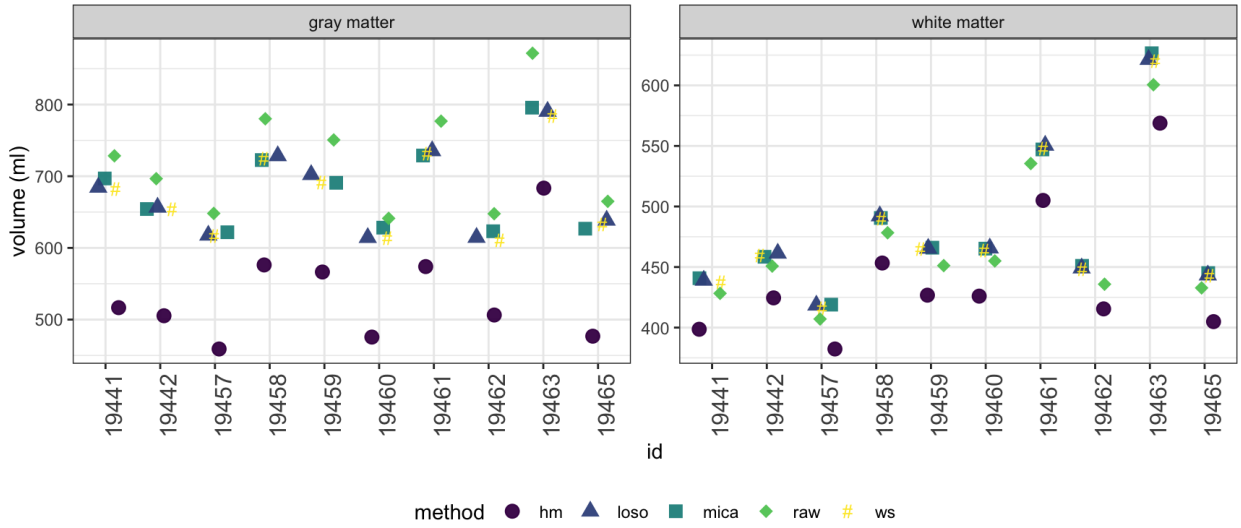

Figure 4.8: Segmented brain volume in the gray matter (left) and white matter (right) for each trio2prisma subject across harmonization approaches. We compare no normalization or harmonization (raw), histogram matching (hm), White Stripe normalization (ws), *mica*, and *loso*.

sites. We also distinguish between approaches targeting these sources of variability: normalization methods address intensity unit effects, and harmonization methods, the focus of our study, address scanner effects.

Our data came from two small studies, the NAIMS pilot study and the trio2prisma study, with multiple images per subject taken on multiple scanners, and nonlinear scanner effects. We found that *mica* reduced within-subject variability in whole brain scans as well as white and gray matter while preserving biological variability across subjects. We also found that *mica*, paired with White Stripe, enhanced reproducibility of measurements of MS lesion volume across sites.

Normalization methods such as histogram matching and White Stripe are sometimes used for harmonization, but they are inadequate in cases where across-site differences are much larger than those within site. Additionally, histogram matching can reduce biological variability across subjects and White Stripe can leave residual technical variability in the gray matter. While we differentiate

conceptually between intensity unit effects and scanner effects, we also acknowledge that in reality these artifacts can be challenging to separate. As a result, *mica* is likely to remove some intensity unit effects and intensity normalization methods are likely to remove some scanner effects when applied separately. In particular, White Stripe alone will likely perform well as a harmonization method when scanner effects are small, linear transformations. Histogram matching, however, is likely to remove desired variability across subjects and bias results.

Because our method is flexible and operates on the full brain, we can map images from one scanner to another. This mapping is only exact for a particular subject when images are available from both scanners, which is not realistic for most studies. That said, our leave-one-scan-out analysis suggests that when systematic site differences are present, *mica* can help understand scanner effects and mitigate those differences. Before conducting multisite studies, we recommend obtaining a baseline measurement of scanner variability by having a subset of patients measured at all sites. Our method can then be applied to all images collected to remove average scanner variability. We acknowledge that this solution is imperfect in the sense that average scanner variability collected from a subset of patients in a trial will not always capture the true scanner variability for each subject. However, our simple and easy-to-apply methodology is an important step forward for an increasingly prevalent problem. There is evidence that scanner effects may vary across covariates such as gender and age, so extensions to *mica* that incorporate covariates may address some of the issues outlined above.

4.5 Software

To enable use of *mica* we have written an *R* software package which is available for download at https://github.com/julia-wrobel/mica.

Chapter 5

A dynamical systems model for the relationship between the motor cortex and skilled movement

5.1 Introduction

Our motivating data comes from a study that collected 3D trajectories of paw position over time as a mouse made a trained reaching motion for a food pellet; the paw reach trajectories were measured concurrently with neural activity in the motor cortex, an area of the brain known to be important for voluntary movement. These data were collected in an effort to understand the dynamics between motor cortex activation and induced behavior. This is an example from the increasingly common class of problems where outcome and responses are measured densely in parallel. For these data streams, we want to understand the relationship between inputs and outputs that are both functions measured on the same domain.

Recent work using these data to investigate how brain activation triggers or inhibits fine motor control during reaches suggests that the dynamics of the arm during complex voluntary movements

are tightly coupled to input from the motor cortex [Guo *et al.*, 2015; Sauerbrei *et al.*, 2018]. To better quantify how brain activity affects current and future paw position, we need a method that (1) allows future position to depend on past but not future neural firing rate, (2) allows future position to be affected by initial position, (3) has parameters that model the relationship between the paw trajectory and the brain as a dynamical system of inputs and outputs, the state of which evolves over time, and (4) can accommodate repeated functional observations across trials. These problems cannot be simultaneously addressed by current methods. We develop a novel regression framework that combines ordinary differential equations (ODEs) and functional regression and is well-suited to address the problems our data presents. This work is connected to both the ODE and functional data analysis literatures, which we review in Sections 5.1.3 and 5.1.4, respectively. First, in Sections 5.1.1 and 5.1.2, we describe our motivating data and model structure in more detail.

5.1.1 Paw trajectory data

The motivating data were collected as part of a study on the specific role of the motor cortex in enacting skilled movement, where a skilled movement is defined as a voluntary behavior that requires coordination and precision. Several experiments from Guo *et al.* [2015] and Sauerbrei *et al.* [2018] show that the motor cortex is necessary and sufficient for enacting a learned skill. Specifically, the motor cortex continually generates a signal driving reach-to-grasp movements in mice.

In the experimental framework that generated our motivating data, a single mouse was trained to reach for a food pellet in a memorized location after hearing an auditory cue. The mouse was fixed at the head to reduce variability in posture, the auditory cue was played, and the mouse enacted the task of picking up its paw from a resting location to reach for and grasp the food

pellet. Video recordings of the task completion were used to extract 3D trajectories of paw position from lift (the point at which the paw leaves its rest position) to grasp (the point at which the paw grasps the food pellet). Prior to experimentation a sensor array was implanted in the motor cortex, with data processing to isolate 25 neurons. Firing rates of the 25 neurons in the motor cortex were simultaneously recorded at a rate of 500 recordings per second to capture the animal's neural dynamics. This describes a single trial of the experiment, which was repeated 147 times. A schematic of this experiment is depicted in Appendix B.

For each trial i, paw position was recorded in the x, y, and z directions over 4 seconds, resulting in trivariate functional observations $\{Y_i^{P_x}(t), Y_i^{P_y}(t), Y_i^{P_z}(t)\}$. Because we treat each direction independently, going forward we simplify notation to $Y_i(t)$ by omitting the superscripts P_x, P_y, and P_z. The auditory cue was played 0.5 seconds into the trial, on average lift occurred at 0.77 seconds, and on average grasp occurred at 0.88 seconds into the trial. For our analysis we limit the time frame to the period 0.05 seconds before lift to 0.25 seconds after lift for each trial, and data is linearly shifted so that the timing of lift for each trial is aligned.

The top row of Figure (5.1) shows the paw positions across trials and axes, from 0.05 seconds before lift to 0.25 seconds after lift. Across axes, paw position at time t depends on initial paw position at the start of the trial. The middle row of Figure (5.1) shows heat maps of the first 2 seconds of firing data for 3 of the 25 neurons, which were chosen because they are representative of patterns seen across neurons. In figures showing neural firing, each row is a trial and each column is a point in time; dark or light shading indicates that a neuron is off or on, respectively. After the auditory cue at 0.5 seconds, neurons at location 2 are mildly activated, neurons at location 6 are highly activated, and neurons at location 9 become less activated. Activity within neurons was fairly consistent across trials, but large differences are seen across neurons.

Firing rates of the 25 neurons were reduced to five dimensions using Gaussian process factor

analysis (GPFA), a standard technique for decomposing noisy neural spiking activity into smooth low-dimension neural trajectories [Yu *et al.*, 2009]. From a neurobiological perspective, extracting emergent patterns in the motor cortex using GPFA is a better way of assessing how neural activity drives behavior than using the raw neural firing data, because it increases generalizability across neurons and trials. From a statistical perspective, GPFA also reduces risk of collinearity when using the neural firing rates as covariates in a regression setting.

Previous work used initial position and neural activity data to predict paw trajectories for held-out trials. However, this work did not allow for the relationship between position and neural activity to vary over time, and did not enhance interpretation of this system of inputs and outputs. We describe our model below; this work introduces a novel regression method that is well-suited to our scientific context.

5.1.2 flode model

The biological underpinnings of our data are a dynamical system where initial position and paw are being acted on by outside forces coming from the motor cortex; these forces drive changes in velocity of the paw which then influences position. We introduce the *flode* (**f**unctional **l**inear **o**rdinary **d**ifferential **e**quation) model, a novel functional regression framework that represents this neurobiological system of inputs (motor cortical activity) and outputs (paw position). The *flode* model is a first-order ordinary differential equation (ODE), which allows us to incorporate how change in paw position influences position at time t, reflecting the dynamic nature of our data. In its differential form, our model is

$$y_i'(t) = -\alpha y_i(t) + \delta_i(t) + \mathcal{B}_0(t) + \sum_{p=1}^{P} \mathcal{B}_p(t) x_{ip}(t), \tag{5.1}$$

where $y_i(t)$ and $y_i'(t)$ are the paw position and first derivative of paw position (velocity) at time

t, $x_{ip}(t), p \in 1 \dots P$ are trial-specific forcing functions, and α, $\delta_i(t)$, and $\mathcal{B}_p(t), p \in 0 \dots P$ are parameters to be estimated from the data. Forcing functions, analogous to covariates in a traditional regression model, are external input forces that act on the ODE system.

This is a buffered system, meaning the response time is longer than the time interval in which the input changes. The scalar parameter α, called the buffering parameter, indicates the amount of buffering on the system. As $\alpha \to 0$, buffering increases, and the effects of forcing functions and initial position persist in time. As α grows larger, the effects of forcing functions and initial position becomes instantaneous. The $\mathcal{B}_p(t)$ are coefficient functions that measure the impact of changes in the forcing function $x_{ip}(t)$ on the system, interpreted as the change in paw velocity at time t, $y_i'(t)$, given a one unit change in forcing function $x_{ip}(t)$. $\mathcal{B}_0(t)$ and $\delta_i(t)$ are the population-level and trial-specific intercepts, respectively. The $\delta_i(t)$ terms capture residual within-trial correlation; while much of fine motor control is known to be driven by the motor cortex, other brain regions also contribute to the paw reaching motion, and the $\delta_i(t)$ term is intended to capture changes in position driven by unmeasured forces.

Many systems of differential equations cannot be solved analytically, which makes traditional statistical estimation techniques with the observed data Y as the outcome challenging. However, the class of ODEs we consider has a solution, which we parameterize in terms of the initial value, given by

$$Y_i(t) = y_i(0)e^{-\alpha t} + \int_0^t e^{-\alpha(t-s)}\delta_i(s)ds + \sum_{p=0}^{P}\int_0^t e^{-\alpha(t-s)}\mathcal{B}_p(s)x_{ip}(s)ds + \epsilon_i(t). \tag{5.2}$$

We differentiate between $y_i(t)$, the true (unobserved) paw position at time t, and $Y_i(t)$, the paw position at time t observed with measurement error $\epsilon_i(t)$. Thus, in model (5.2) above, we assume the outcome $Y_i(t)$ is measured with error but depends on the true initial position $y_i(0)$.

The *flode* model is a first-order linear differential equation, which reflects the biological process

that is hypothesized to generate the observed data. To familiarize readers with ordinary differential equations and their use in statistics, we review the ODE literature in Section 5.1.3. An overview of the functional data analysis literature dedicated to regression, provided in Section 5.1.4, is also pertinent since the paw trajectories will be conceptualized and modeled as i.i.d. realizations of functions that are observed over time.

5.1.3 ODEs

Systems of ordinary differential equations (ODEs) can be used to directly model the relationship between an outcome and its derivatives, leading to widespread popularity for modeling dynamical systems in physics, biology, neuroscience, and other disciplines. First-order ODEs, which incorporate only the first derivative of y, follow the form given in Equation (5.1), though the $\delta_i(t)$ term we include is unconventional.

Equation (5.1) is also said to be a linear differential equation because its right hand side can be written as a linear combination of y and terms that do not contain y [Tennenbaum and Pollard, 1985]. Though more complex ODEs are possible, such as those of higher order or with nonlinearity, we believe the simpler model can capture the dynamics of our data. When analytically solvable, most ODEs do not have a unique solution. It is therefore common, and useful for our data setting, to solve in terms of the initial value $y(0)$.

Most applications of ODEs in science and engineering focus on restrictive rather than general settings, in part because parameter estimation for general models is challenging. In the past this specificity has limited their use in statistics, but they are growing in popularity. Chen *et al.* [2017] reconstructs gene regulatory networks by estimating sparse nonlinear ODEs for noisy gene expression data, building on previous work [Lu *et al.*, 2011; Henderson and Michailidis, 2014].

In their instant classic Dynamic Data Analysis, Ramsay and Hooker [2017] conceptualize dy-

namical systems as data-driven statistical models. The book provides a framework for estimating a large class of differential equations, as well as an excellent overview of ODE-based models that expands on earlier work from Ramsay *et al.* [2007] for parameter estimation in nonlinear ODEs. Separate estimation frameworks are provided for linear and nonlinear ODEs though both involve a tradeoff between the best fit to a particular prespecified ODE and a smooth fit to the data, enforced using B-spline expansions. While this general framework is well-suited to estimate parameters for a single realization of an ODE, it does not accommodate multiple trials or the complexities that arise in that case.

5.1.4 Functional regression models

Our data setting and proposed methods are also closely related to functional data analysis. In functional data analysis, curve $Y_i(t)$ is the fundamental unit of statistical analysis [Ramsay and Silverman, 2005], and functional analogs of univariate methods like regression, PCA, and others build on this framework. Functional regression models capture the relationship between outcome curves $Y_i(t), i \in 1 \dots N$ from N independent trials, and the covariate(s) x_i, which can be scalar or functional. In particular, function-on-function regression allows for both functional responses and functional predictors that can be observed on different domains, and the response is related to the predictor through integration of a coefficient surface [Ramsay and Silverman, 2005].

Some special cases of function-on-function regression include the linear functional concurrent model [Fan and Zhang, 2008; Goldsmith and Schwartz, 2017] and the historical functional regression model [Malfait and Ramsay, 2003]. The concurrent model uses the current value of the predictor to measure the response at each time, but doesn't allow the covariates to affect future values of the response. The historical functional model allows the response at time t to be influenced only by the predictors up to time t; this is ideal for data where the response and predictor are measured

on the same domain, and prevents future values of the predictors from influencing the present value of the response. Advances in functional regression and accompanying software allow for historical functional regression models with scalar and functional covariates, as well as functional trial-specific random effects [Scheipl *et al.*, 2015, 2016; Crainiceanu *et al.*, 2015]. The historical model with trial-specific random intercept $\gamma_i(t)$ is given by

$$Y_i(t) = \gamma_i(t) + \beta_0(t) + \sum_{p=1}^{P} \int_{s=0}^{t} \beta_p(t,s) x_{ip}(s) ds + \epsilon_i(t). \tag{5.3}$$

Here $\beta_0(t)$ is the population-level intercept, and each $\beta_p(t,s)$ is a coefficient surface. This flexible model is designed to handle repeated functional observations, and inclusion of the random intercept $\gamma_i(t)$ accounts for within-trial residual correlation in the errors after modeling the relationship between the outcome and the covariates curves.

Conceptually both the integrated *flode* model in 5.2 and historical functional regression use predictors, including their recent history, to understand current values of the response function. Because of these high-level similarities we find it useful to compare and contrast these methods. If we assume the surface $\beta(t,s)$ from Equation (5.3) takes the form $e^{-\alpha(t-s)}\mathcal{B}(s)$ from Equation (5.2), $\gamma_i(t) = y_i(0)e^{-\alpha t} + \int_0^t e^{-\alpha(t-s)}\delta_i(s)ds$, and $\beta_0(t) = \int_0^t e^{-\alpha(t-s)}\mathcal{B}_0(s)ds$, then *flode* can be considered a special case of the historical functional regression model. However, the *flode* surface $e^{-\alpha(t-s)}\mathcal{B}(s)$ is very restricted compared to the more general historical surface $\beta(s,t)$; as a result, the historical model is likely too flexible and may overfit data that is generated by the *flode* model.

These assumptions are not trivial. From a conceptual standpoint, *flode* introduces a new framework for thinking about the relationship between inputs and outputs in an ODE system, and the historical model does not offer this interpretation. Initial position is a crucial element of the *flode* framework because it provides a specific analytic solutions to the ODE in 5.1; in contrast initial position is not a natural element of the historical model and does not have precedent in the functional

regression literature. Explicitly incorporating initial position into a functional regression context is both critical for our dynamical systems approach and a novel contribution in its own right. Finally, the *flode* model is nonlinear in its parameter α, a development which other functional regression methods haven't directly addressed.

5.2 Methods

Our work introduces models (5.1) and (5.2), a novel framework for modeling functional observations with an explicit dynamical systems interpretation.

5.2.1 Model formulation

The *flode* method is a system of differential equations, where equation (5.1) represents the model on the scale of the paw velocity, and equation (5.2) on the scale of the paw position. Because we observe paw position data rather than paw velocities, we estimate parameters using the paw position model. However, we are interested in interpretation on the velocity scale.

In this section we explain our parameter estimation approach. The buffering parameter α will be estimated using nonlinear least squares. Since we observe initial position with error, $Y_i(0)$, we also need to estimate true initial position, $y_i(0)$. The random effects $\delta_i(t)$ and coefficient functions $\mathcal{B}_p(t)$ will be expanded using B-splines. Under these conditions all parameters will be estimated jointly using the algorithm described in Section 5.2.2.

To induce smoothness and reduce dimensionality, the trial-specific random intercepts $\delta_i(t)$ and coefficient functions $\mathcal{B}_p(t)$ are expanded using a fixed B-spline basis, $\boldsymbol{\Theta}(t)$, of K_t basis functions $\theta_1(t), \ldots, \theta_{K_t}(t)$, such that $\delta_i(t) = \boldsymbol{\Theta}(t)\boldsymbol{d}_i$ and $\mathcal{B}_p(t) = \boldsymbol{\Theta}(t)\boldsymbol{b}_p$, where $\boldsymbol{d}_i$, $i \in 1 \ldots N$ is a $K_t \times 1$ vectors of spline coefficients for the random intercept of the ith trial, and $\boldsymbol{b}_p$, $p \in 1 \ldots P$ is a $K_t \times 1$ vector of spline coefficients for the pth coefficient function. Using this representation each forcing

function term becomes.

$$\begin{aligned}
\sum_{p=1}^{P}\int_{s=0}^{t} e^{-\alpha(t-s)}\cdot x_{ip}(\boldsymbol{s})\cdot\mathcal{B}_p(s)ds &= \sum_{p=1}^{P}\int_{s=0}^{t} e^{-\alpha(t-s)}\cdot x_{ip}(s)\cdot\Theta(s)\boldsymbol{b}_p ds \\
&= \sum_{p}\left(\int_{s=0}^{t}\left[\{e^{-\alpha(t-s)}\cdot x_{ip}(s)\}\otimes 1_{K_t}^T\right]\cdot\Theta(s)ds\right)\boldsymbol{b}_p \\
&= \sum_{p} x_{ip}^*(t,\alpha)\boldsymbol{b}_p \\
&= \boldsymbol{x}_i^*(t,\alpha)\boldsymbol{b},
\end{aligned}$$

where $\otimes$ denotes the element-wise Kronecker product, and 1_{K_t} is a length K_t column vector with each entry equal to 1. We define $\boldsymbol{x}_i^*(t,\alpha) = \{x_{i1}^*(t,\alpha)|\ldots|x_{iP}^*(t,\alpha)\}$ and $\boldsymbol{b} = \left(\boldsymbol{b}_1^T|\ldots|\boldsymbol{b}_P^T\right)^T$. Similarly, the random intercept term becomes

$$\begin{aligned}
\int_{s=0}^{t} e^{-\alpha(t-s)}\cdot\delta_i(s)ds &= \int_{s=0}^{t} e^{-\alpha(t-s)}\cdot\Theta(s)\boldsymbol{d}_i ds \\
&= \left[\int_{s=0}^{t}\left\{e^{-\alpha(t-s)}\otimes 1_{K_t}^T\right\}\cdot\Theta(s)ds\right]\boldsymbol{d}_i \\
&= \mathcal{D}^*(t,\alpha)\boldsymbol{d}_i,
\end{aligned}$$

Finally, we define $y_{i0}^*(t,\alpha) = y_i(0)e^{-\alpha t}$.

Though the conceptual model is expressed over continuous time domain t, in practice, each trajectory Y_i is observed on the discrete grid, $\boldsymbol{t} = \{t_1, t_2, \ldots, t_D\}$, which we assume to be equally spaced and shared across trials. Functions $Y_i(\boldsymbol{t})$ evaluated on this grid are vectors of length D, and $\mathcal{D}^*(\boldsymbol{t},\alpha)$ and $x_{ip}^*(\boldsymbol{t},\alpha)$ are $D\times K_t$ matrices. Letting $\boldsymbol{\Theta}(\boldsymbol{t})$ be the $D\times K_t$ spline matrix evaluated at $\boldsymbol{t}$, then $\delta_i(\boldsymbol{t}) = \boldsymbol{\Theta}(\boldsymbol{t})\boldsymbol{d}_i$ and $\mathcal{B}_p(\boldsymbol{t}) = \boldsymbol{\Theta}(\boldsymbol{t})\boldsymbol{b}_p$. Putting these terms together and evaluating on grid $\boldsymbol{t}$ gives the observed data model,

$$Y_i(\boldsymbol{t}) = y^*_{i0}(\boldsymbol{t}, \alpha) + \mathcal{D}^*(\boldsymbol{t}, \alpha)\boldsymbol{d}_i + \boldsymbol{x}^*_i(\boldsymbol{t}, \alpha)\boldsymbol{b} + \epsilon_i(\boldsymbol{t}). \tag{5.4}$$

We use the notation $g^*(t, \alpha)$ above to highlight that terms $\boldsymbol{x}^*_i(t, \alpha)$, $\mathcal{D}^*(t, \alpha)$, and $y^*_{i0}(t, \alpha)$ are all functions of both time t and the model parameter α. However, throughout this section these terms will be used interchangeably with the terms $\boldsymbol{x}^*_i$, $\mathcal{D}^*$, and y^*_{i0} for notational simplicity. Naturally, on a discrete grid the integral defined above needs to be approximated numerically. For numeric integration we use a Riemannian approach, but other approaches would be reasonable as well.

We assume both the spline coefficients for the trial-specific intercept, $\boldsymbol{d}_i$, and the white noise, $\epsilon_i(t)$, are random and have the following distributions

$$\epsilon_i(t) \sim N(0, \sigma^2 I_D)$$

$$\boldsymbol{d}_i \sim N(0, \Sigma_{Kt \times Kt}),$$

which induces a conditionally normal distribution on the observed data given the random effects,

$$Y_i | \boldsymbol{d}_i \sim N\left(y^*_{i0} + \mathcal{D}^*\boldsymbol{d}_i + \boldsymbol{x}^*_i\boldsymbol{b}, \sigma^2 I_D\right).$$

For the purpose of the derivations below we simplify the random intercept variance to $\Sigma_{Kt} = \lambda I_{Kt}$.

We estimate the buffering parameter α, variance parameters σ^2 and λ, true initial positions $y_i(0)$, and spline coefficients $\boldsymbol{b}$ and $\boldsymbol{d}_i$ using the expectation-maximization algorithm described in below. The algorithm incorporates a nonlinear least squares step to optimize the α parameter.

5.2.2 EM algorithm for estimating fixed and random effects

We use an expectation-maximization (EM) algorithm to find the maximum likelihood estimates (MLEs) of both fixed and random effects, following precedent from Laird and Ware [1982] for

longitudinal data and Walker [1996] for nonlinear mixed models. Our goal is to estimate the experiment-wide fixed effects $\mathbf{\Phi} = \left\{\alpha, \boldsymbol{b}, y_i(0), \sigma^2, \lambda\right\}$ and the random effect spline coefficients $\boldsymbol{d}_i$. In the M-step of the algorithm we estimate the MLE of the fixed effects when the random effects are observed, $\widehat{\mathbf{\Phi}} = \underset{\mathbf{\Phi}}{\text{argmax}}\{l(\mathbf{\Phi}|Y)\}$, and in the E-step we get estimates for the random effects by taking the expectation of the $\boldsymbol{d}_i$ under the posterior distribution of $\boldsymbol{d}_i$ given the data Y_i.

5.2.2.1 M-step

When the random effects $\boldsymbol{d}_i$ are known, the MLE of $\mathbf{\Phi}$ maximizes the joint log-likelihood

$$\begin{aligned} l(\mathbf{\Phi})) &= \log p(Y, \boldsymbol{d}; \mathbf{\Phi}) \\ &= \log p(Y|\boldsymbol{d}; \mathbf{\Phi}) + \log p(\boldsymbol{d}; \mathbf{\Phi}) \\ &= \log p\left\{Y|\boldsymbol{d}; \alpha, \mathbf{b}, y_i(0), \sigma^2\right\} + \log p(\boldsymbol{d}; \lambda). \end{aligned}$$

This leads to the following fixed effects

$$\begin{aligned} \hat{\alpha} &= \underset{\alpha}{\text{argmin}} \epsilon^T \epsilon \\ \widehat{\mathbf{b}} &= \left(\boldsymbol{x}^{*T}\boldsymbol{x}^*\right)^{-1} \boldsymbol{x}^{*T}\left(Y - y_0^* - \boldsymbol{m}\mathcal{D}^* < \boldsymbol{d} >\right) \\ \widehat{y}_i(0) &= \frac{(e^{-\alpha \boldsymbol{t}})^T \left\{Y_i - \mathcal{D}^* < \boldsymbol{d}_i > - \boldsymbol{x}_i^* \boldsymbol{b}\right\}}{(e^{-2\alpha \boldsymbol{t}})^T 1_D} \\ \widehat{\sigma}^2 &= \frac{\epsilon^T \epsilon}{ND} \\ \widehat{\lambda} &= \frac{\sum_i tr(< \boldsymbol{d}_i \mathbf{d}_i^T >)}{NK_t}. \end{aligned}$$

The notation $tr(A)$ indicates the trace of matrix A, and 1_D is a length D column vector with each

entry equal to 1. When not indexed by i, the vectors Y and y_0^* denote length ND stacked forms of their trial-specific length D counterparts, Y_i and y_{i0}^*. Similarly, $\boldsymbol{d}$ is a stacked length NK_t vector, and $\boldsymbol{x}^*$ and $\boldsymbol{m}\mathcal{D}^*$ are stacked $ND \times K_t$ matrices. The residual sum of squares, $\epsilon^T\epsilon$, is given by

$$\begin{aligned}
\epsilon^T\epsilon &= (Y - y_0^* + \boldsymbol{m}\mathcal{D}^*\boldsymbol{d} + \boldsymbol{x}^*\boldsymbol{b})^T(Y - y_0^* + \boldsymbol{m}\mathcal{D}^*\boldsymbol{d} + \boldsymbol{x}^*\boldsymbol{b}) \\
&= Y^TY - 2Y^T\left(y_0^* + \boldsymbol{m}\mathcal{D}^*\boldsymbol{d} + \boldsymbol{x}^*\mathbf{b}\right) + y_0^{*T}y_0^* + 2y_0^{*T}\left(\boldsymbol{m}\mathcal{D}^* < \boldsymbol{d} > + \boldsymbol{x}^*\mathbf{b}\right) \\
&+ \left(\boldsymbol{x}^*\mathbf{b}\right)^T\left(\boldsymbol{x}^*\mathbf{b}\right) + 2\left(\boldsymbol{x}^*\mathbf{b}\right)^T\boldsymbol{m}\mathcal{D}^* < \mathbf{d} > + < \mathbf{d}^T\boldsymbol{m}\mathcal{D}^{*T}\boldsymbol{m}\mathcal{D}^*\mathbf{d} > .
\end{aligned}$$

The notation $< \ldots >$ represents the expected values of $\boldsymbol{d}$ and $\boldsymbol{d}^T\mathcal{D}^{*T}\mathcal{D}^*\boldsymbol{d}$, the estimation of which are detailed in the E-step below.

5.2.2.2 E-step

Bayes' rule leads to the posterior distribution of the random intercept coefficients,

$$\mathbf{d}_i|Y_i \sim N\left(\mathbf{m}_i, \boldsymbol{C}\right),$$

where

$$\boldsymbol{C} = \left\{\frac{1}{\lambda}I_{K_t} + \frac{\mathcal{D}^{*T}\mathcal{D}^*}{\sigma^2}\right\}^{-1},$$

and

$$\mathbf{m}_i = \frac{\boldsymbol{C}\mathcal{D}^{*T}\left(Y_i - y_{i0}^* - \boldsymbol{x}_i^*\boldsymbol{b}\right)}{\sigma^2}.$$

Then the solutions to $< \boldsymbol{d}_i >$ and $< \boldsymbol{d}_i^T\mathcal{D}^{*T}\mathcal{D}^*\boldsymbol{d}_i >$ are $\mathbf{m}_i$ and $tr(\mathcal{D}^{*T}\mathcal{D}^*\boldsymbol{C}) + \mathbf{m}_i^T\mathcal{D}^{*T}\mathcal{D}^*\mathbf{m}_i$, respectively. We iterate between the M-step and the E-step to obtain a solution. The algorithm converges when the squared difference between the current estimate of $\widehat{\boldsymbol{\Phi}}$ and its value in the previous iteration become arbitrarily small.

The random intercept in the *flode* model is included to capture residual within-trial correlation in the paw trajectories. If one is willing to assume that the residuals are uncorrelated, then for each trial $\delta_i(t) = 0$ and the *flode* model simplifies, which allows parameters $\widehat{\mathbf{\Phi}}$ to be maximized directly without the E-step.

5.2.3 Implementation

Our methods are implemented in `R` and publicly available on `GitHub`. We use nonlinear least squares to estimate α, which is implemented using the `optim` function, which uses a golden-section search algorithm to minimize the squared error loss in Equation 5.4.Good initialization is important for fast convergence when using the `optim` function. For this reason, we recommend doing a grid search to find a value α_0 that minimizes the loss function when $\delta_i(t) = 0$, and use this to initialize our full EM algorithm. Initial position $y_i(0)$ is initialized using the observed initial position $Y_i(0)$, and random effects $\delta_i(\boldsymbol{t})$ are initialized at 0.

5.3 Simulations

We assess the performance of our method using simulations designed to mimic the structure of our motivating data. Simulated data is generated from the *flode* model in Equation 5.2, varying over the true value of the α parameter to obtain simulation settings that evaluate the sensitivity of our method as α changes.

5.3.1 Simulation design

Each simulated dataset has $N = 100$ univariate paw trajectories $Y_i(\boldsymbol{t})$ with a population intercept $\mathcal{B}_0(t)$ and one forcing function $\boldsymbol{x}_1(t)$. All trials share the same equally-spaced grid, $\boldsymbol{t} \in [0, 1]$, of length $D = 50$. To reflect how initial values vary across trials in the motivating data, for

each trial i, initial position $y_i(0)$ is sampled from $N(0, 5)$. The forcing function takes the form $\boldsymbol{x}_{i1}(\boldsymbol{t}) = scale_i \times \sin(\pi_i \boldsymbol{t} + shift_i)$, where $scale_i$ and $shift_i$ are randomly-drawn, trial-specific scale and shift parameters. Random intercepts $\delta_i(\boldsymbol{t})$ are constructed using 10 B-spline basis functions $\boldsymbol{\Theta}(\boldsymbol{t})$ and spline coefficients $\boldsymbol{d}_i$, are drawn from $\boldsymbol{d}_i \sim N(0, \lambda I_{10})$, where $\lambda = 50$. Measurement errors $\epsilon_i(\boldsymbol{t})$ are drawn from $\epsilon_i(\boldsymbol{t}) \sim N(0, \sigma^2 I_D)$, where $\sigma^2 = 0.1$, an amount of residual variance which is comparable to that seen in our motivating data.

Figure 5.2 shows three simulated datasets when $\alpha = 2$, $\alpha = 6$, and $\alpha = 12$. The middle and bottom rows show paw position trajectories $Y_i(t)$ and coefficient surfaces $e^{-\alpha(t-s)}\mathcal{B}_1(s)$, respectively, across α values. The top row shows (from left to right) forcing functions $x_{i1}(t)$ and random intercepts on the derivative scale $\delta_i(t)$, which do not depend on α and are shared across these three datasets. The middle panel highlights the buffering effect of α. When $\alpha = 2$ buffering is high, meaning initial position has a consistent effect on the overall trajectory over the time span of the trial. When $\alpha = 12$ buffering is low, and the impact of initial position and forcing functions becomes instantaneous.

We evaluate performance of our model as a function of the buffering parameter α. For each $\alpha \in (2, 4, 6, 8, 10, 12)$, we simulate 25 different datasets, and apply the methods described in Section 5.2 to each dataset. For model estimation we choose $K_t = 10$ B-spline basis functions. We initialize α using a rough grid search over $\alpha \in [1, 14]$ to find the value of α that minimize sum of squared error when $\delta_i(t) = 0$. The true initial position $y_i(0)$ is initialized using the observed initial position $Y_i(0)$, and random effects $\delta_i(\boldsymbol{t})$ are initialized at 0.

5.3.2 Comparison with historical functional regression

We compare *flode* to the historical functional regression model in (5.3). This model is implemented using the `pffr` function from the `refund` package in `R` [Crainiceanu *et al.*, 2015], and is denoted *fhist*

in text and figures below. Comparisons between *flode* and *fhist* are made based on recovery of the true coefficient surfaces. We define the surface from the *flode* model as $\beta_1^{flode}(s,t) = e^{-\alpha(t-s)}\mathcal{B}_1(s)$, and compare it to *fhist* surface $\beta_1^{fhist}(s,t)$. Surface recovery accuracy is quantified using the integrated squared error (ISE), where for *flode* $ISE = \int_t \int_s \left\{\beta_1(s,t) - \widehat{\beta}_1^{flode}(s,t)\right\}^2 dsdt$ and for *fhist* $ISE = \int_t \int_s \left\{\beta_1(s,t) - \widehat{\beta}_1^{fhist}(s,t)\right\}^2 dsdt$. We also compare *flode* and *fhist* based on recovery of the true measurement error, $\sigma^2 = 0.1$.

The buffering parameter α is an important component of the *flode* model but is not estimated by the historical functional model. In figures below, in addition to comparing the performance of *flode* and *fhist*, we also visualize how well our *flode* implementation recovers the true value of α across simulation scenarios.

5.3.3 Simulation results

Figure 5.3 shows results from a single simulated dataset with $\alpha = 6$ and 100 trials. From top to bottom, rows show observed (gray) and fitted (red) values, true (gray) and estimated (red) random effects on the data scale, and coefficient surfaces. The top and middle rows show results for *fhist* (left column) and *flode* (right column), while the bottom row shows the *fhist*, *flode*, and true surfaces, respectively. For this simulated dataset, both *flode* and *fhist* produce reasonable results for the fitted values. However, it is clear from the random effects and coefficient surfaces that *flode* and *fhist* are estimating these overall fits in different ways, and that *flode* is recovering the true surface values.

Figure 5.4 summarizes results for *flode* and *fhist* across datasets generated using different values of α. The left panel shows $\log ISE$, and the right panel shows estimated measurement errors $\widehat{\sigma}^2_{flode}$ and $\widehat{\sigma}^2_{fhist}$. Across values of α, *flode* outperforms *fhist* in terms of the ISE, which is consistent with observations in Figure 5.3. At low values of α, the difference in performance between the methods

is smaller, and ISE variability for *flode* is high when $\alpha = 2$. Measurement error is slightly biased away from the true value $\sigma^2 = 0.1$ for both models, though the bias is larger for the *fhist* model across values of α. The *flode* model is slightly overfitting the data, while *fhist* underfits the data.

Figure 5.5 shows estimated values $\widehat{\alpha}$ (top row) and random effects variance $\widehat{\lambda}$ (bottom row) from *flode* across datasets with different true values of α. Our *flode* implementation recovers close to the true value of α, though values are slightly biased towards zero for datasets with higher true values of α. Our estimates for λ are also slightly biased towards zero, an effect which is also pronounced for higher true values of α. Though not shown, the simulations described above were also performed at the increased sample size of $N = 200$ trials. When sample size increases, the variance of α and λ estimates decreases, the algorithm converges in fewer iterations, and ISE values are lower.

5.4 Data Analysis

In this section, we apply the methods described in Section 5.2 to the mouse paw trajectory data introduced in Section 5.1.1. Our dataset consists of 147 paw trajectory trials from a single mouse, where each trajectory was collected under the same experimental conditions. Accompanying paw trajectories are measurements of brain activity in the motor cortex, as summarized by GPFA [Yu *et al.*, 2009], for a total of 5 forcing functions. Position and neural activity were recorded concurrently at a rate of 500 measurements per second. We restrict our analysis to the period just before lift (when the paw leaves a resting location) to just after grasp (when the paw grasps a food pellet). Because grasp occurred at different times across trials, we linearly interpolate the data to an even grid of length $D = 50$ that is shared across trials.

We present a univariate analysis of the trajectories in the x, y, and z directions. For each axis, we set the number of B-spline bases to $K_t = 10$ and fit the *flode* model in (5.2). The parameter α was initialized by doing a grid search over values in $[2, 12]$ to find the value, α_0, that minimizes the

model when each $\delta_i(t) = 0$. This α_0 was then used as a starting value for the *flode* algorithm. The results of this analysis are described and interpreted below.

Estimated values of the buffering parameter, are, for each axis, $\widehat{\alpha}_x = 3.01$, $\widehat{\alpha}_y = 3.50$, and $\widehat{\alpha}_z = 3.01$. These values are close, indicating similar amounts of buffering across axes. For Figure 5.6, the first row shows observed (gray) and fitted (red) values for paw position. The second row shows random effects on the derivative scale for each trial, $\delta_i(t)$. The third row shows these random effects on the data scale, $\int_s e^{-\alpha(t-s)}\delta_i(s)ds$. The first, second, and third columns show results for the x, y, and z axis, respectively. The dotted line through each plot occurs at $t = 0.05$ seconds, which is the time of paw lift for each trial. The fitted values are capturing the data well. The random effects show more residual variance right after lift (during the time of the actual reach) than in other parts of the trial, suggesting that maybe there is something driving the reaching movement that we are not measuring.

Coefficient functions and coefficient surfaces are shown in Figures 5.7 and 5.8, respectively. For surfaces we only show results from the x axis; results from the y and z axes followed the same trends.

5.5 Discussion

We present *flode*, a nonlinear regression model that has context in both functional data analysis and systems of ordinary differential equations. Drawing from both of these literatures is necessitated by our application; the differential equations formulation of our model allows for an interpretation of our paw data as trajectories whose speed and position are dynamically influenced by inputs from the brain, and tools from functional data analysis allow us to efficiently model repeated observations that are trajectories while incorporating smoothness in the coefficient functions. Though we are motivated by a specific application in neurobiology, our methods are general and broadly

useful for anyone trying to study a dynamical system of inputs and outputs where the outputs are functions over time. Our novel method compares favorably with historical functional regression in the simulation settings we examined, and produces reasonable results for our motivating data. Our methods are publicly available in an `R` package.

We believe this work is an exciting addition to a nascent field in statistics, with many possible future directions. A study on the asymptotics of the coefficients estimated in this model so that large sample confidence intervals and hypothesis tests can be computed would help researchers draw inferences about the relationships between inputs and outputs of the dynamical system. Extensions to include more complex systems of ordinary differential equations, including higher order and non-linear ODEs would increase the flexibility of our modeling framework and allow for the study of a larger class of repeated measurements of dynamical systems.

The *flode* model was developed based on our current understanding of biological processes, and we're working to expand that framework to include more complex inputs. For example, we view the $\delta_i(t)$ term as capturing correlation due to unmeasured forces acting on the system. Prior work suggests that this signal is coming from the thalamus and it would be useful to work with neurobiologists to collect data and develop a model that incorporates neural information from multiple sources within the brain, with the ultimate goal of recreating reaching movements based only on initial position and neural activity patterns.

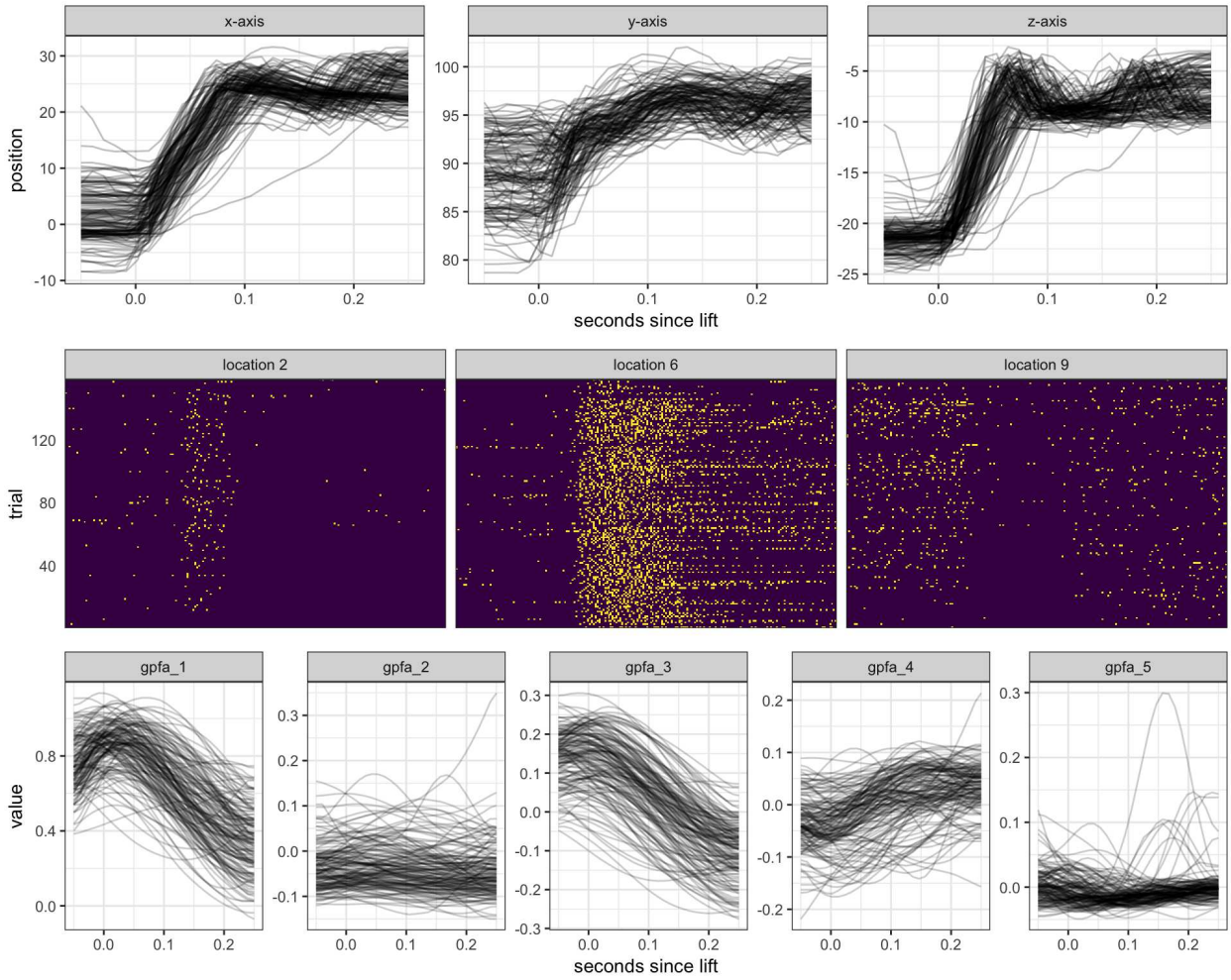

Figure 5.1: Top row: Paw trajectories along x, y, and z axes for 147 trials. Middle row: neural firing rates for 3 of the 25 neurons. Each row is a trial and each column is a point in time, and dark or light shading indicates that a neuron is off or on, respectively, at that point in time. After auditory cue, neurons show light activation at location 2, high activation at location 6, and dampening in activation at location 9. Bottom row: The five factors from Gaussian process factor analysis, shown for all 147 trials.

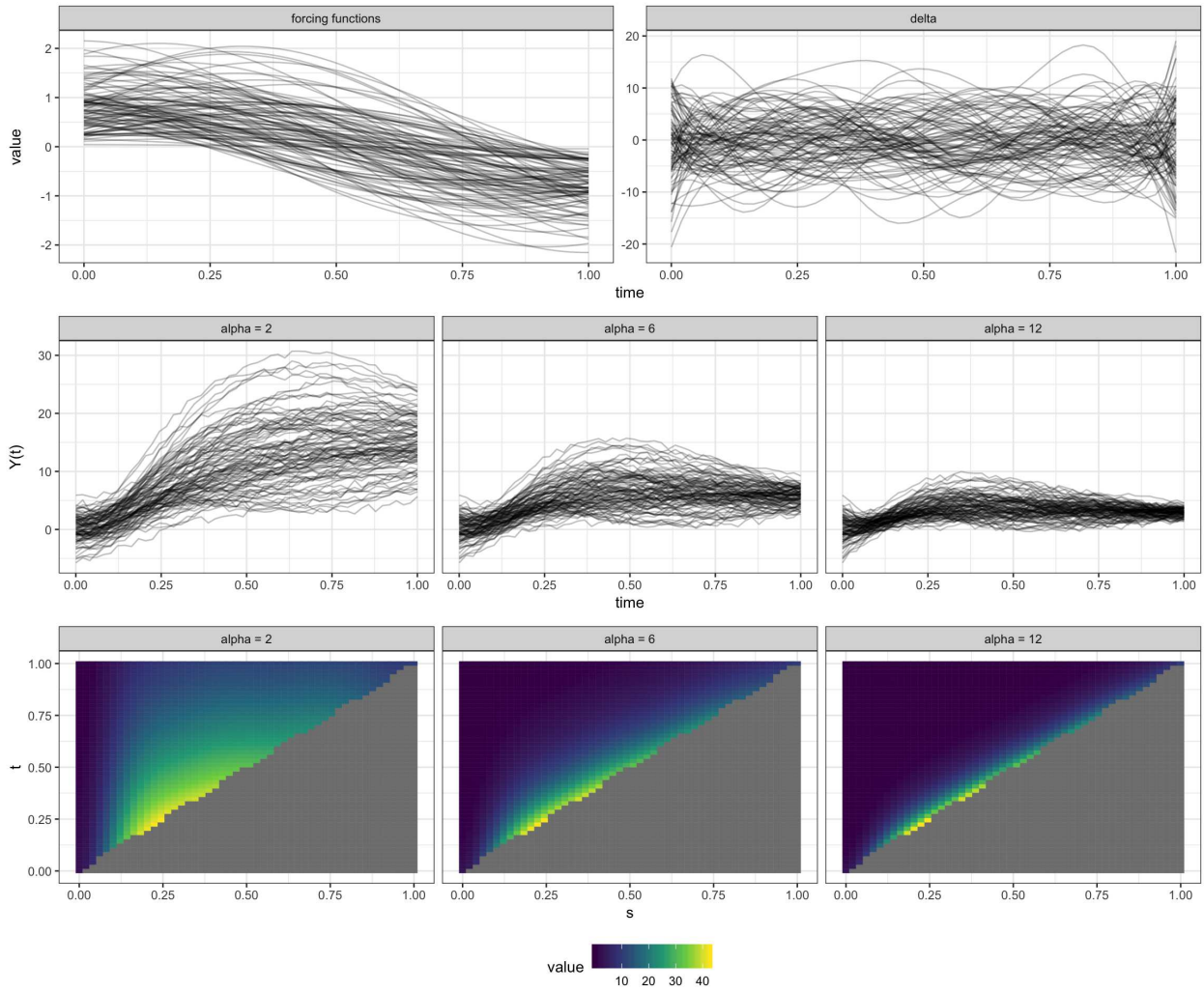

Figure 5.2: This figure shows simulated data when $\alpha = 2$, $\alpha = 6$, and $\alpha = 12$. Top row: Left column shows forcing functions $x_{i1}(t)$, right column shows random effects on the paw velocity scale, $\delta_i(t)$. Middle row: Observed paw positions $Y_i(t)$ for three different values of α. When α is small initial position has a larger effect on the overall trajectory. Bottom row: Coefficient surfaces $e^{-\alpha(t-s)}\mathcal{B}_1(s)$ for three different values of α.

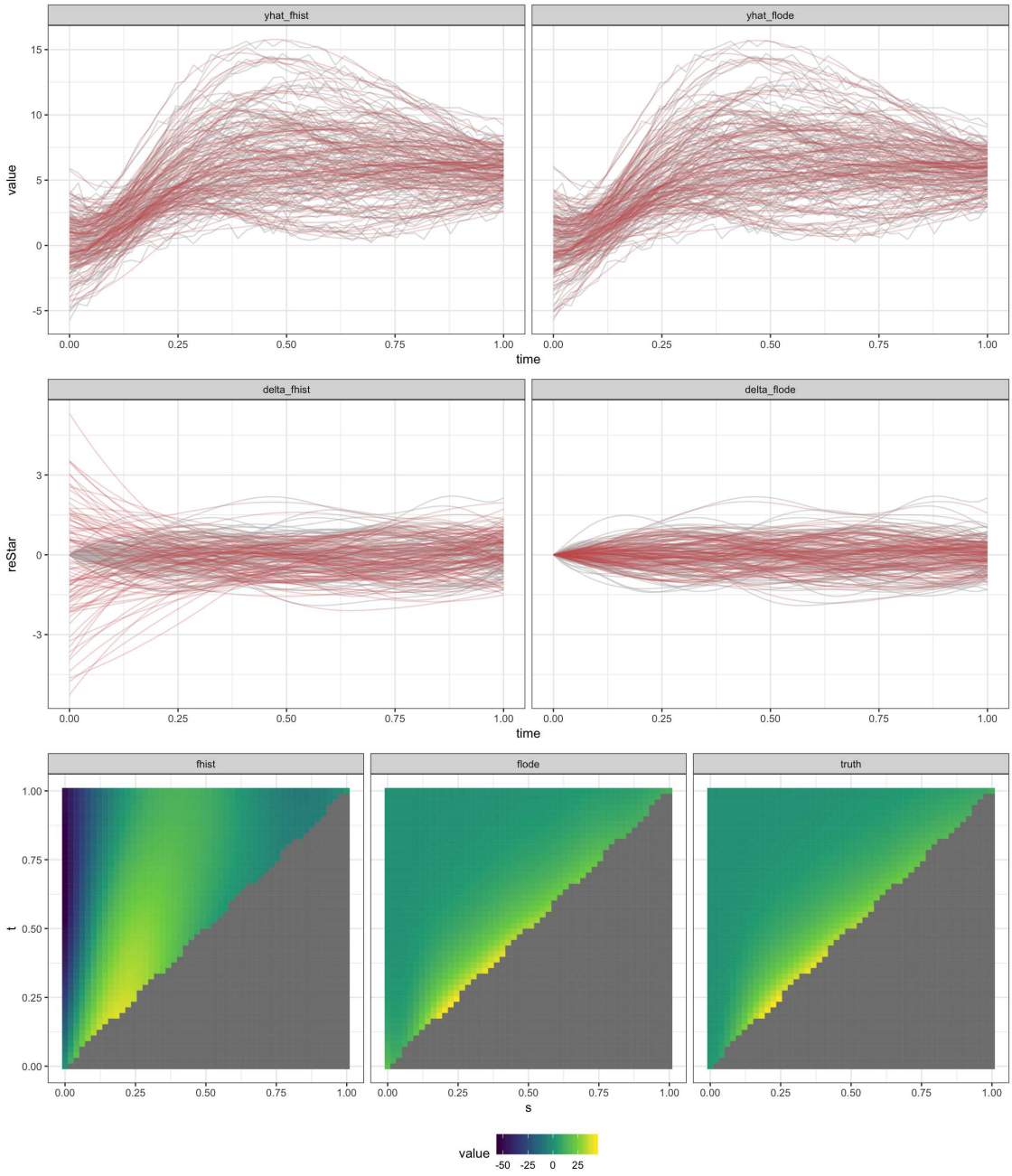

Figure 5.3: Top row: Fitted values from *fhist* and *flode*. Second row: Residuals from *fhist* and *flode*. Third row: Random intercepts from *fhist* and *flode*. Values for *flode* are shown on the data scale so that they are comparable with *fhist*. Bottom row: Estimated surfaces from *fhist* and *flode*. Both models run on the same dataset with $\alpha = 6$ and $N = 100$ trials.

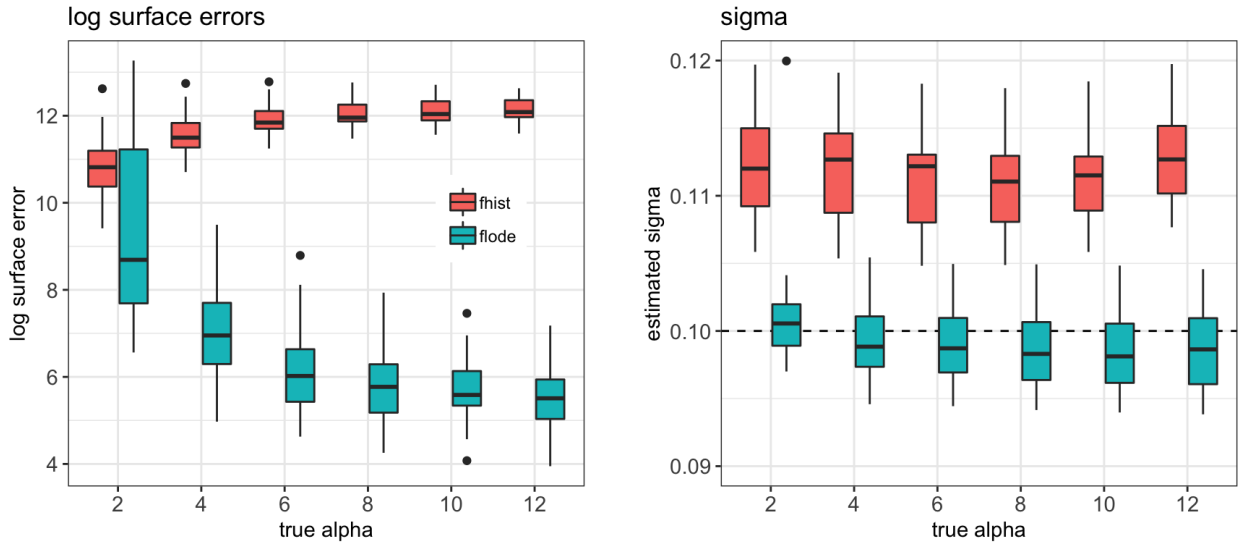

Figure 5.4: Log surface errors (left panel) and estimated measurement error $\widehat{\sigma}^2$ (right panel) for *flode* (red) and *fhist* (green) across varying values of α when $N = 100$ trials. Dotted line in right panel is through the true value $\sigma^2 = 0.1$.

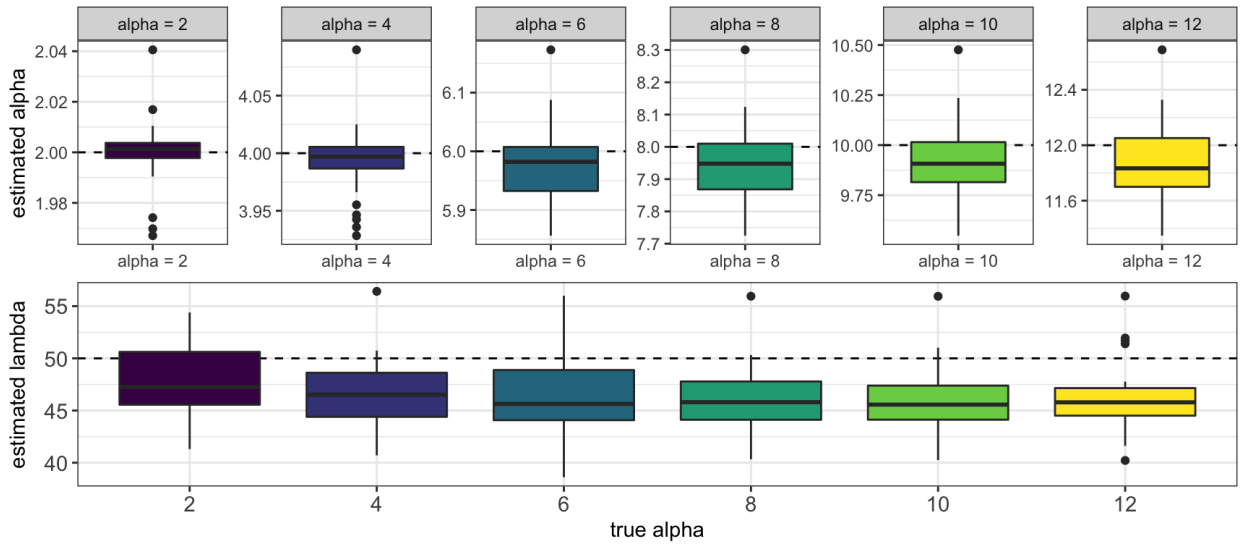

Figure 5.5: Estimated $\widehat{\alpha}$ (top row) and $\widehat{\lambda}$ (botttom row) values from *flode* model across simulated datasets with different true values of α. The horizontal dotted line the the plot on the bottom row represents the true value, $\lambda = 50$.

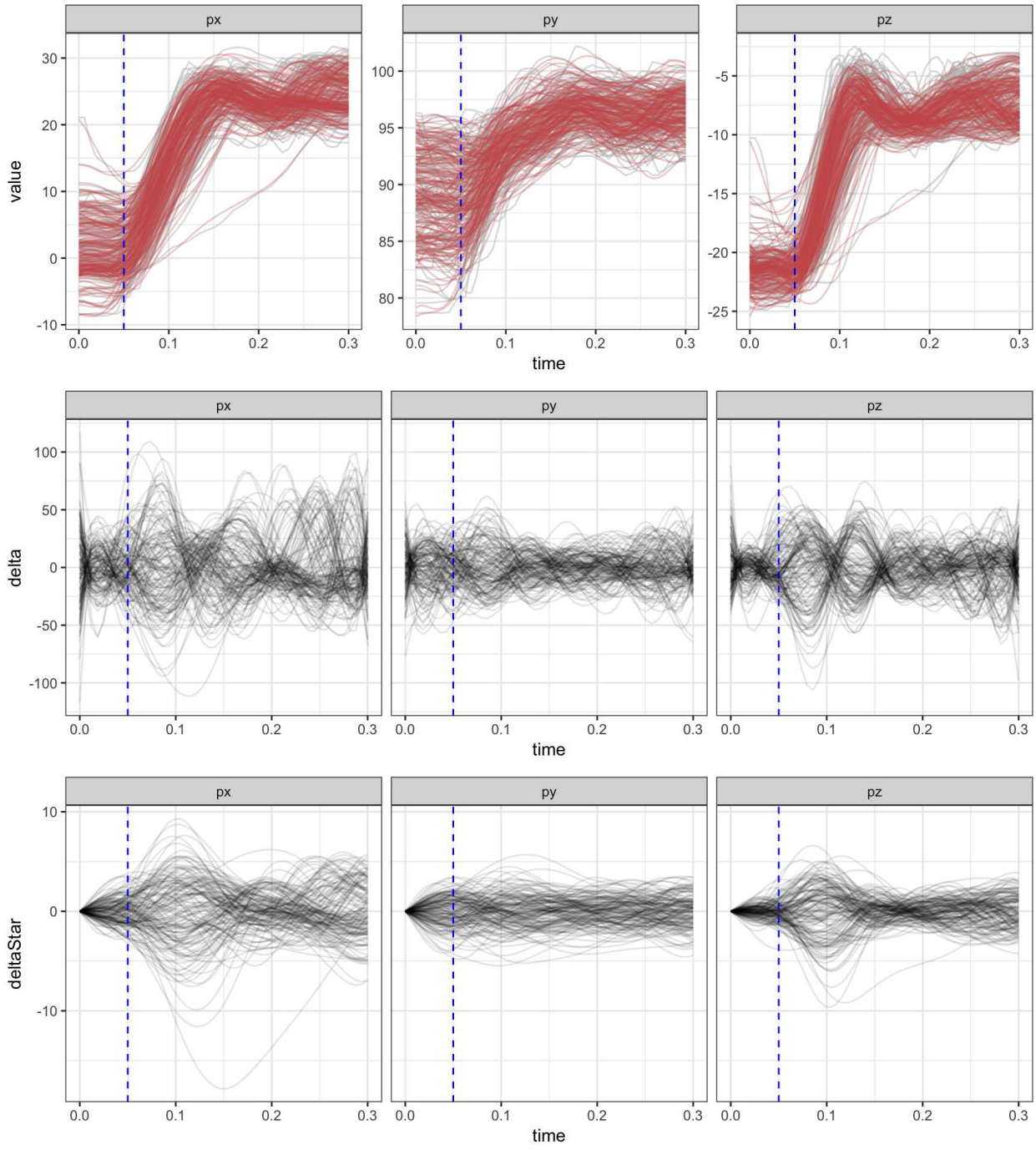

Figure 5.6: This figure shows fitted values, estimated random effects, and integrated random effects across axes for the paw data. The vertical dotted line occurs at the time of lift for each trial.

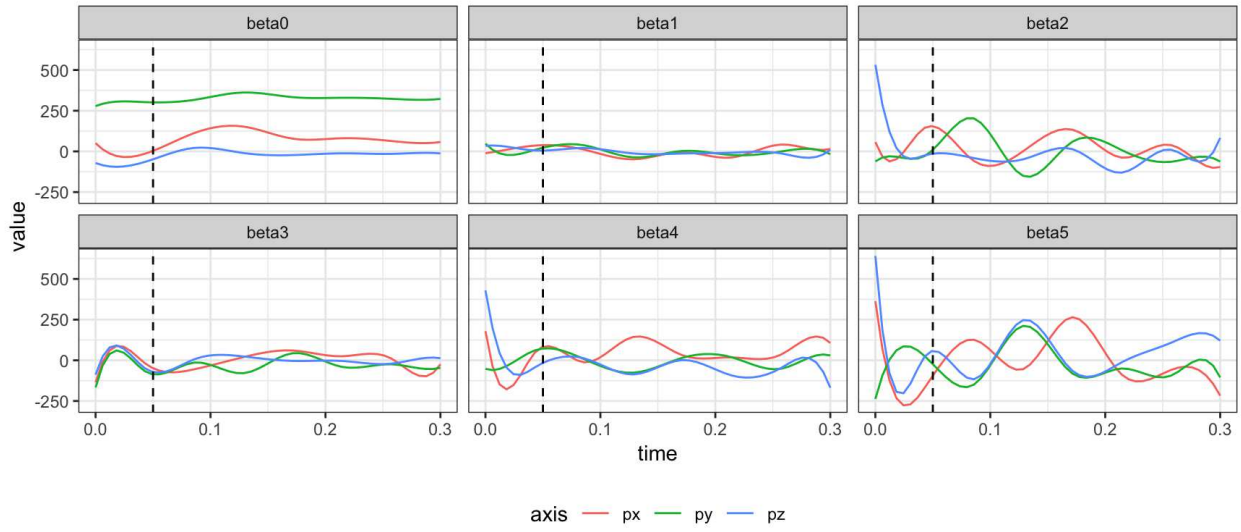

Figure 5.7: This figure shows fitted intercept and coefficient functions across axes for the paw data. The vertical dotted black line occurs at the point of lift.

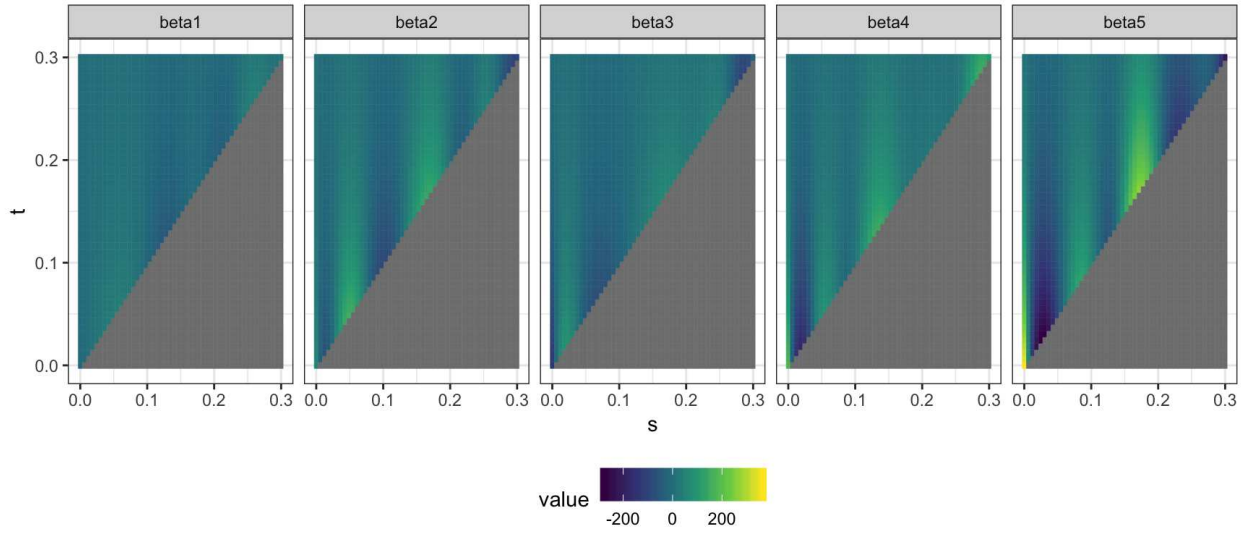

Figure 5.8: This figure shows estimated surfaces for the x-axis of the paw data for each forcing function. Similar results were seen for the y and z axes.

Bibliography

Ana Adan, Simon N Archer, Maria Paz Hidalgo, Lee Di Milia, Vincenzo Natale, and Christoph Randler. Circadian typology: a comprehensive review. *Chronobiology international*, 29(9):1153–1175, 2012.

Rohit Bakshi, Mohit Neema, Brian C. Healy, Zsuzsanna Liptak, Rebecca A. Betensky, Guy J. Buckle, Susan A. Gauthier, James Stankiewicz, Dominik Meier, Svetlana Egorova, Ashish Arora, Zachary D. Guss, Bonnie Glanz, Samia J. Khoury, Charles R. G. Guttmann, and Howard L. Weiner. Predicting Clinical Progression in Multiple Sclerosis With the Magnetic Resonance Disease Severity Scale. *Archives of Neurology*, 65(11):1449–1453, 11 2008.

B Brumback and J Rice. Smoothing spline models for the analysis of nested and crossed samples of curves. *Journal of the American Statistical Association*, 93:961–976, 1998.

Tyrone D Cannon, Frank Sun, Sarah Jacobson McEwen, Xenophon Papademetris, George He, Theo GM van Erp, Aron Jacobson, Carrie E Bearden, Elaine Walker, Xiaoping Hu, et al. Reliability of neuroanatomical measurements in a multisite longitudinal study of youth at risk for psychosis. *Human brain mapping*, 35(5):2424–2434, 2014.

Kehui Chen and Hans-Georg Müller. Modeling repeated functional observations. *Journal of the American Statistical Association*, 107:1599–1609, 2012.

Shizhe Chen, Ali Shojaie, and Daniela M Witten. Network reconstruction from high-dimensional ordinary differential equations. *Journal of the American Statistical Association*, 112(520):1697–1707, 2017.

C Crainiceanu, P Reiss, J Goldsmith, J Gellar, Harezlak J, M W McLean, B Swihart, L Xiao, Y Chen, S Greven, M G Kundu, J Wrobel, L Huang, L Huo, and F Scheipl. *refund: Regression with Functional Data*, 2015. R package version 0.1-13.

Marie-Hélène Descary and Victor M Panaretos. Functional data analysis by matrix completion. *arXiv preprint arXiv:1609.00834*, 2016.

C-Z Di, C M Crainiceanu, B S Caffo, and N M Punjabi. Multilevel functional principal component analysis. *Annals of Applied Statistics*, 4:458–488, 2009.

C-Z Di, C M Crainiceanu, and S J Jank. Multilevel sparse functional principal component analysis. *Stat*, 3:126–143, 2014.

KM Diaz, VJ Howard, B Hutto, and et al. Patterns of sedentary behavior and mortality in u.s. middle-aged and older adults: A national cohort study. *Annals of Internal Medicine*, 2017.

JD Dworkin, P Sati, A Solomon, DL Pham, R Watts, ML Martin, D Ontaneda, MK Schindler, DS Reich, and RT Shinohara. Automated integration of multimodal mri for the probabilistic detection of the central vein sign in white matter lesions. *American Journal of Neuroradiology*, 39(10):1806–1813, 2018.

Jianqing Fan and Wenyang Zhang. Statistical methods with varying coefficient models. *Statistics and its Interface*, 1(1):179, 2008.

Massimo Filippi, Jerry S Wolinsky, Giancarlo Comi, CORAL Study Group, et al. Effects of oral

glatiramer acetate on clinical and mri-monitored disease activity in patients with relapsing multiple sclerosis: a multicentre, double-blind, randomised, placebo-controlled study. *The Lancet Neurology*, 5(3):213–220, 2006.

J P Fortin, E Fertig, and K Hansen. shinymethyl: interactive quality control of illumina 450k dna methylation arrays in r [version 1; referees: 2 approved]. *f1000research*, 3:175, 2014.

Jean-Philippe Fortin, Elizabeth M Sweeney, John Muschelli, Ciprian M Crainiceanu, Russell T Shinohara, Alzheimer's Disease Neuroimaging Initiative, et al. Removing inter-subject technical variability in magnetic resonance imaging studies. *NeuroImage*, 132:198–212, 2016.

Jean-Philippe Fortin, Drew Parker, Birkan Tunc, Takanori Watanabe, Mark A Elliott, Kosha Ruparel, David R Roalf, Theodore D Satterthwaite, Ruben C Gur, Raquel E Gur, et al. Harmonization of multi-site diffusion tensor imaging data. *Neuroimage*, 161:149–170, 2017.

Jean-Philippe Fortin, Nicholas Cullen, Yvette I Sheline, Warren D Taylor, Irem Aselcioglu, Philip A Cook, Phil Adams, Crystal Cooper, Maurizio Fava, Patrick J McGrath, et al. Harmonization of cortical thickness measurements across scanners and sites. *NeuroImage*, 167:104–120, 2018.

M G Genton, S Castruccio, P Crippa, S Dutta, R Huser, Y Sun, and S Vettori. Visuanimation in statistics. *Stat*, 4:81–96, 2015.

J Gertheiss, V Maier, EF Hessel, and A-M Staicu. Marginal functional regression models for analyzing the feeding behavior of pigs. *Journal of Agricultural, Biological, and Environmental Statistics,*, To Appear, 2015.

J Gertheiss, J Goldsmith, and A-M Staicu. A note on modeling sparse exponential-family functional response curves. *Computational Statistics and Data Analysis*, 105:46–52, 2017.

Rezwan Ghassemi, Robert Brown, Sridar Narayanan, Brenda Banwell, Kunio Nakamura, and Douglas L Arnold. Normalization of white matter intensity on t1-weighted images of patients with acquired central nervous system demyelination. *Journal of Neuroimaging*, 25(2):184–190, 2015.

J Goldsmith and T Kitago. Assessing systematic effects of stroke on motor control using hierarchical function-on-scalar regression. *Journal of the Royal Statistical Society: Series C*, page To Appear, 2015.

Jeff Goldsmith and Joseph E Schwartz. Variable selection in the functional linear concurrent model. *Statistics in medicine*, 36(14):2237–2250, 2017.

J Goldsmith and J Wrobel. *refund.shiny: Interactive plotting for functional data analyses*, 2015. R package version 0.1.

J Goldsmith, J Bobb, C M Crainiceanu, B Caffo, and D Reich. Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20:830–851, 2011.

J Goldsmith, C M Crainiceanu, B Caffo, and D Reich. Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements. *Journal of the Royal Statistical Society: Series C*, 61:453–469, 2012.

J Goldsmith, S Greven, and C M Crainiceanu. Corrected confidence bands for functional data using principal components. *Biometrics*, 69:41–51, 2013.

J Goldsmith, V Zipunnikov, and J Schrack. Generalized multilevel function-on-scalar regression and principal component analysis. *Biometrics*, 71:344–353, 2015.

S Greven, C M Crainiceanu, B Caffo, and D Reich. Longitudinal functional principal component analysis. *Electronic Journal of Statistics*, 4:1022–1054, 2010.

Jian-Zhong Guo, Austin R Graves, Wendy W Guo, Jihong Zheng, Allen Lee, Juan Rodriguez-Gonzalez, Nuo Li, John J Macklin, James W Phillips, Brett D Mensh, et al. Cortex commands the performance of skilled movement. *Elife*, 4:e10774, 2015.

W Guo. Functional mixed effects models. *Biometrics*, 58:121–128, 2002.

P Z Hadjipantelis, J A D Aston, H-G Müller, and J P Evans. Unifying amplitude and phase analysis: a compositional data approach to functional multivariate mixed-effects modeling of mandarin chinese. *Journal of the American Statistical Association*, 110:545–559, 2015.

P Hall, H-G Müller, and F Yao. Modelling sparse generalized longitudinal observations with latent gaussian processes. *Journal of the Royal Statistical Society: Series B*, 70:703–723, 2008.

Stephen L. Hauser, Amit Bar-Or, Giancarlo Comi, Gavin Giovannoni, Hans-Peter Hartung, Bernhard Hemmer, Fred Lublin, Xavier Montalban, Kottil W. Rammohan, Krzysztof Selmaj, Anthony Traboulsee, Jerry S. Wolinsky, Douglas L. Arnold, Gaelle Klingelschmitt, Donna Masterman, Paulo Fontoura, Shibeshih Belachew, Peter Chin, Nicole Mairon, Hideki Garren, and Ludwig Kappos. Ocrelizumab versus interferon beta-1a in relapsing multiple sclerosis. *New England Journal of Medicine*, 376(3):221–234, 2017. PMID: 28002679.

James Henderson and George Michailidis. Network reconstruction using nonparametric additive ode models. *PloS one*, 9(4):e94003, 2014.

H Huang, L Yehua, and Y Guan. Joint modeling and clustering paired generalized longitudinal trajectories with application to cocaine abuse treatment data. *Journal of the American Statistical Association*, 109.508:1412–1424, 2014.

Wen Huang, Kyle A Gallivan, Anuj Srivastava, Pierre-Antoine Absil, et al. Riemannian optimization for elastic shape analysis. In *Mathematical theory of Networks and Systems*, 2014.

T S Jaakkola and M I Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, 1997.

G M James, T J Hastie, and C A Sugar. Principal component models for sparse functional data. *Biometrika*, 87:587–602, 2000.

Jorge Jovicich, Moira Marizzoni, Roser Sala-Llonch, Beatriz Bosch, David Bartrés-Faz, Jennifer Arnold, Jens Benninghoff, Jens Wiltfang, Luca Roccatagliata, Flavio Nobili, et al. Brain morphometry reproducibility in multi-center 3 t mri studies: a comparison of cross-sectional and longitudinal segmentations. *Neuroimage*, 83:472–484, 2013.

Ludwig Kappos, Jack Antel, Giancarlo Comi, Xavier Montalban, Paul O'Connor, Chris H. Polman, Tomas Haas, Alexander A. Korn, Goeril Karlsson, and Ernst W. Radue. Oral fingolimod (fty720) for relapsing multiple sclerosis. *New England Journal of Medicine*, 355(11):1124–1140, 2006. PMID: 16971719.

Anisha Keshavan, Friedemann Paul, Mona K Beyer, Alyssa H Zhu, Nico Papinutto, Russell T Shinohara, William Stern, Michael Amann, Rohit Bakshi, Antje Bischof, et al. Power estimation for non-standardized multisite studies. *NeuroImage*, 134:281–294, 2016.

A Kneip and J O Ramsay. Combining registration and fitting for functional models. *Journal of the American Statistical Association*, 103:1155–1165, 2008.

N M Laird and J H Ware. Random-effects models for longitudinal data. *Biometrics*, pages 963–974, 1982.

S Lopez-Pintado and J Romo. On the concept of depth for functional data. *Journal of the American Statistical Association*, 104:486–503, 2009.

Tao Lu, Hua Liang, Hongzhe Li, and Hulin Wu. High-dimensional odes coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association*, 106(496):1242–1258, 2011.

N Malfait and J O Ramsay. The historical functional linear model. *Canadian Journal of Statistics*, 31:115–128, 2003.

J S Marron, J O Ramsay, L M Sangalli, and A Srivastava. Functional data analysis of amplitude and phase variation. *Statistical Science*, 30(4):468–484, 2015.

K R Martin, A Koster, R A Murphy, D R Van Domelen, M Y Hung, R J Brychta, K Y Chen, and T B Harris. Changes in daily activity patterns with age in u.s. men and women: National health and nutrition examination survey 2003?04 and 2005?06. *Journal of the American Geriatrics Society*, 62(7):1263–1271, 2014.

J S Morris and R J Carroll. Wavelet-based functional mixed models. *Journal of the Royal Statistical Society: Series B*, 68:179–199, 2006.

Jeffrey S Morris, Marina Vannucci, Philip J Brown, and Raymond J Carroll. Wavelet-Based Nonparametric Modeling of Hierarchical Functions in Colon Carcinogenesis. *Journal of the American Statistical Association*, 98:573–583, 2003.

J S Morris. Functional regression analysis. *Annual Review of Statistics and Its Application*, 2(1), 2015.

Susanne G Mueller, Michael W Weiner, Leon J Thal, Ronald C Petersen, Clifford R Jack, William Jagust, John Q Trojanowski, Arthur W Toga, and Laurel Beckett. Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative (adni). *Alzheimer's & Dementia*, 1(1):55–66, 2005.

László G Nyúl, Jayaram K Udupa, et al. On standardizing the mr image intensity scale. *image*, 1081, 1999.

Jiwon Oh, Rohit Bakshi, Peter A Calabresi, Ciprian Crainiceanu, Roland G Henry, Govind Nair, Nico Papinutto, R Todd Constable, Daniel S Reich, Daniel Pelletier, et al. The naims cooperative pilot project: Design, implementation and future directions. *Multiple Sclerosis Journal*, 24(13):1770–1772, 2018.

V M Panaretos and Y Z Zemel. Amplitude and phase variation of point processes. *The Annals of Statistics*, 44:771–812, 2016.

Nico Papinutto, Rohit Bakshi, Antje Bischof, Peter A Calabresi, Eduardo Caverzasi, R Todd Constable, Esha Datta, Gina Kirkish, Govind Nair, Jiwon Oh, et al. Gradient nonlinearity effects on upper cervical spinal cord area measurement from 3d t1-weighted brain mri acquisitions. *Magnetic resonance in medicine*, 79(3):1595–1601, 2018.

SY Park and A-M Staicu. Longitudinal functional data analysis. *Stat*, 4:212–226, 2015.

James Ramsay and Giles Hooker. *Dynamic data analysis*. Springer, 2017.

J O Ramsay and B W Silverman. *Functional Data Analysis*. New York: Springer, 2005.

Jim O Ramsay, Giles Hooker, David Campbell, and Jiguo Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

P T Reiss, L Huang, and M Mennes. Fast function-on-scalar regression with penalized basis expansions. *International Journal of Biostatistics*, 6:Article 28, 2010.

RStudio Inc. *shiny: Web Application Framework for R*, 2015. R package version 0.12.2.

L M Sangalli, P Secchi, S Vantini, and V Vitelli. k-mean alignment for curve clustering. *Computational Statistics & Data Analysis*, 54:1219–1233, 2010.

Britton Sauerbrei, Jian-Zhong Guo, Matteo Mischiati, Wendy Guo, Mayank Kabra, Nakul Verma, Kristin Branson, and Adam Hantman. Motor cortex is an input-driven dynamical system controlling dexterous movement. *bioRxiv*, page 266320, 2018.

Fabian Scheipl, Ana-Maria Staicu, and Sonja Greven. Functional additive mixed models. *Journal of Computational and Graphical Statistics*, 24(2):477–501, 2015.

Fabian Scheipl, Jan Gertheiss, Sonja Greven, et al. Generalized functional additive mixed models. *Electronic Journal of Statistics*, 10(1):1455–1492, 2016.

Hugo G Schnack, Neeltje EM van Haren, Hilleke E Hulshoff Pol, Marco Picchioni, Matthias Weisbrod, Heinrich Sauer, Tyrone Cannon, Matti Huttunen, Robin Murray, and René S Kahn. Reliability of brain volumes from multicenter mri acquisition: a calibration study. *Human brain mapping*, 22(4):312–320, 2004.

Hugo G Schnack, Neeltje EM van Haren, Rachel M Brouwer, G Caroline M van Baal, Marco Picchioni, Matthias Weisbrod, Heinrich Sauer, Tyrone D Cannon, Matti Huttunen, Claude Lepage, et al. Mapping reliability in multicenter mri: Voxel-based morphometry and cortical thickness. *Human brain mapping*, 31(12):1967–1982, 2010.

J A Schrack, V Zipunnikov, J Goldsmith, J Bai, E M Simonshick, C M Crainiceanu, and L Ferrucci. Assessing the "physical cliff": Detailed quantification of aging and physical activity. *Journal of Gerontology: Medical Sciences*, 2014.

Daniel L Schwartz, Ian Tagge, Katherine Powers, Sinyeob Ahn, Rohit Bakshi, Peter A Calabresi, R Todd Constable, John Grinstead, Roland G Henry, Govind Nair, et al. Multisite reliability

and repeatability of an advanced brain mri protocol. *Journal of Magnetic Resonance Imaging*, 2019.

N Serban, A-M Staicu, and R J Carrol. Multilevel cross-dependent binary longitudinal data. *Biometrics*, 69:903–913, 2013.

Mohak Shah, Yiming Xiao, Nagesh Subbanna, Simon Francis, Douglas L Arnold, D Louis Collins, and Tal Arbel. Evaluating intensity normalization on mris of human brain with multiple sclerosis. *Medical image analysis*, 15(2):267–282, 2011.

Russell T Shinohara, Ciprian M Crainiceanu, Brian S Caffo, María Inés Gaitán, and Daniel S Reich. Population-wide principal component-based quantification of blood–brain-barrier dynamics in multiple sclerosis. *NeuroImage*, 57(4):1430–1446, 2011.

Russell T Shinohara, Elizabeth M Sweeney, Jeff Goldsmith, Navid Shiee, Farrah J Mateen, Peter A Calabresi, Samson Jarso, Dzung L Pham, Daniel S Reich, Ciprian M Crainiceanu, et al. Statistical normalization techniques for magnetic resonance imaging. *NeuroImage: Clinical*, 6:9–19, 2014.

R T Shinohara, J Oh, G Nair, P A Calabresi, C Davatzikos, J Doshi, R G Henry, G Kim, K A Linn, N Papinutto, et al. Volumetric analysis from a harmonized multisite brain mri study of a single subject with multiple sclerosis. *American Journal of Neuroradiology*, 38(8):1501–1509, 2017.

Shinohara, R T and Muschelli, J. *WhiteStripe: White Matter Normalization for Magnetic Resonance Images using WhiteStripe*, 2018. R package version 2.3.1.

Stephen M Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.

H Sørensen, J Goldsmith, and L Sangalli. An introduction with medical applications to functional data analysis. *Statistics in Medicine*, 32:5222–5240, 2013.

A Srivastava, W Wu, S Kurtek, E Klassen, and J S Marron. Registration of functional data using fisher-rao metric. *arXiv preprint arXiv*, 1103.3817, 2011.

Stan Development Team. *shinystan: Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models*, 2015. R package version 2.0.1.

E Strauss, E Sherman, and O Spreen. *Compendium of neuropsychological tests: Administration, norms, and commentary*. New York: Oxford University Press, 2006.

Y Sun and M G Genton. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2), 2011.

Y Sun, M G Genton, and D W Nychka. Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked? *Stat*, 1(1):68–74, 2012.

Elizabeth M Sweeney, Russell T Shinohara, Navid Shiee, Farrah J Mateen, Avni A Chudgar, Jennifer L Cuzzocreo, Peter A Calabresi, Dzung L Pham, Daniel S Reich, and Ciprian M Crainiceanu. Oasis is automated statistical inference for segmentation, with applications to multiple sclerosis lesion segmentation in mri. *NeuroImage: clinical*, 2:402–413, 2013.

EM Sweeney, RT Shinohara, CD Shea, DS Reich, and CM Crainiceanu. Automatic lesion incidence estimation and detection in multiple sclerosis using multisequence longitudinal mri. *American Journal of Neuroradiology*, 34(1):68–73, 2013.

Morris Tennenbaum and Harry Pollard. Ordinary differential equations: an elementary textbook for students of mathematics, engineering, and the sciences, 1985.

M E Tipping and C Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61:611–622, 1999.

M E Tipping. Probabilistic visualisation of high-dimensional binary data. *Advances in neural information processing systems*, pages 592–598, 1999.

J D Tucker, W Wu, and A Srivastava. Generative models for functional data using phase and amplitude separation. *Computational Statistics and Data Analysis*, 61:50–66, 2013.

J D Tucker. *fdasrvf: Elastic Functional Data Analysis*, 2017. R package version 1.8.1.

Nicholas J Tustison, Brian B Avants, Philip A Cook, Yuanjie Zheng, Alexander Egan, Paul A Yushkevich, and James C Gee. N4itk: improved n3 bias correction. *IEEE transactions on medical imaging*, 29(6):1310–1320, 2010.

A m Valcarcel, K A Linn, S N Vandekar, T D Satterthwaite, J Muschelli, P A Calabresi, D L Pham, M L Martin, and R T Shinohara. Mimosa: An automated method for intermodal segmentation analysis of multiple sclerosis brain lesions. *Journal of Neuroimaging*, 2018.

A van der Linde. Variational Bayesian Functional PCA. *Computational Statistics and Data Analysis*, 53:517–533, 2008.

David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

Stephen Walker. An em algorithm for nonlinear random effects models. *Biometrics*, pages 934–944, 1996.

Hongzhi Wang, Jung W Suh, Sandhitsu R Das, John B Pluta, Caryne Craige, and Paul A Yushkevich. Multi-atlas segmentation with joint label fusion. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):611–623, 2013.

H Wickham and W Chang. *ggplot2: An Implementation of the Grammar of Graphics*, 2015. R package version 1.0.1.

Julia Wrobel, So Young Park, Ana Maria Staicu, and Jeff Goldsmith. Interactive graphics for functional data analyses. *Stat*, 5:108–118, 03 2016.

Julia Wrobel, Vadim Zipunnikov, Jennifer Schrack, and J Goldsmith. Registration for exponential family functional data. *Biometrics*, 2018.

W Wu and A Srivastava. Analysis of spike train data: Alignment and comparisons using the extended fisher-rao metric. *Electronic Journal of Statistics*, 8:1776–1785, 2014.

F. Yao, H.G. Müller, and J.L. Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005.

Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.

Meichen Yu, Kristin A Linn, Philip A Cook, Mary L Phillips, Melvin McInnis, Maurizio Fava, Madhukar H Trivedi, Myrna M Weissman, Russell T Shinohara, and Yvette I Sheline. Statistical harmonization corrects site effects in functional connectivity measurements from multi-site fmri data. *Human brain mapping*, 39(11):4213–4227, 2018.

Chen Yue. *Generalizations, extensions and applications for principal component analysis*. PhD thesis, Johns Hopkins University, 2016.

Part I

Appendices

Appendix A

Appendix to Registration for exponential family functional data

A.1 Methods

Here we provide extra details of our methods.

A.1.1 Variational approximation to Bernoulli likelihood

Our method for binary functional principal components analysis uses a variational approximation to the Bernoulli likelihood given in Equation (6) of our manuscript. Recall that for subject i measured at time j observation $Y_i(t_{ij}) \sim Bernoulli(\mu_i(t_{ij}))$ where $\mu_i(t_{ij})$ is defined in Equation (3). For convenience we rearrange the usual formulation of the Bernoulli distribution to get the probability density function given in Equation (4) by

$$\begin{aligned} P\{Y_i(t_{ij})|\boldsymbol{c}_i\} &= \mu_i(t_{ij})^{Y_i(t_{ij})}\{1-\mu_i(t)\}^{1-Y_i(t_{ij})} \\ &= g^{-1}\{A_i(t_{ij})\}^{Y_i(t_{ij})}\left[1-g^{-1}\{A_i(t_{ij})\}\right]^{1-Y_i(t_{ij})} \\ &= g^{-1}\left[\{2Y_i(t_{ij})-1\}\,A_i(t_{ij})\right] \end{aligned}$$

where $A_i(t_{ij}) = \boldsymbol{\Theta}_\phi(t_{ij})\,(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i)$. Equality holds in the last step because observations are limited to $\{0, 1\}$ and when g is the logit function $g^{-1}(-z) = 1 - g^{-1}(z)$.

We use the variational approximation for logistic regression outlined by Jaakkola and Jordan [1997] and extended to binary PCA by Tipping and Bishop [1999], with additional basis expansion $\boldsymbol{\Theta}_\phi(t_{ij})$ embedded in $A_i(t_{ij})$ to allow for a functional data framework. This approximation is a lower bound on $P\{Y_i(t_{ij})|\boldsymbol{c}_i\}$, given by

$$\begin{aligned} P\{Y_i(t_{ij})|\boldsymbol{c}_i\} &= g^{-1}\left[\{2Y_i(t_{ij}) - 1\}\, A_i(t_{ij})\right] \\ &\geq g^{-1}\{\xi_i(t_{ij})\} \\ &\times \exp\left[\frac{\{2Y_i(t_{ij}) - 1\}\, A_i(t_{ij}) - \xi_i(t_{ij})}{2} + \lambda\left\{\xi_i(t_{ij})\right\}\left\{A_i(t_{ij})^2 - \xi_i(t_{ij})^2\right\}\right] \\ &= \widetilde{P}\{Y_i(t_{ij})|\boldsymbol{c}_i, \xi_i(t_{ij})\}, \end{aligned}$$

where $\lambda(z) = \frac{0.5 - g^{-1}(z)}{2z}$ and $\xi_i(t_{ij})$ is the variational parameter. Equality of the original distribution $P\{Y_i(t_{ij})|\boldsymbol{c}_i\}$ and the variational distribution is attained when $\widetilde{P}\{Y_i(t_{ij})|\boldsymbol{c}_i, \xi_i(t_{ij})\}$ is maximized with respect to $\xi_i(t_{ij})$, and the value of $\xi_i(t_{ij})$ at the maximum is
$\{2Y_i(t_{ij}) - 1\}\, A_i(t_{ij})$.

A.1.2 Updating α_Θ and Ψ_Θ for binary FPCA

We obtain parameter updates for binary FPCA by maximizing the variational likelihood given in Equation (7) of our manuscript. In Section (3.1.3) we obtain updates for $\boldsymbol{\alpha}_\Theta$ and $\boldsymbol{\Psi}_\Theta$ by reparameterizing Equation (7) such that $\boldsymbol{\Phi} = (\boldsymbol{\Psi}_\Theta^T, \boldsymbol{\alpha}_\Theta)^T$. This reparameterization leads to the

variational log-likelihood below:

$$\begin{aligned}
\widetilde{l}(\boldsymbol{Y},\boldsymbol{c}) &\propto \sum_{j=1}^{D_i}\sum_{i=1}^{I}\log\widetilde{P}\Big\{Y_i(t_{ij})|\boldsymbol{c}_i,\xi_i(t_{ij})\Big\}-\sum_i \boldsymbol{c}_i^T\boldsymbol{c}_i \\
&\propto \sum_i\left[\left\{Y_i(\boldsymbol{t}_i)-\frac{1}{2}\right\}^T A_i(\boldsymbol{t}_i)-\frac{1}{2}\xi_i(\boldsymbol{t}_i)\mathbb{1}_{D_i\times 1}+A_i^T(\boldsymbol{t}_i)diag\left[\lambda\left\{\xi_i(\boldsymbol{t}_i)\right\}\right]A_i(\boldsymbol{t}_i)\right] \\
&\propto \sum_i\left\{Y_i(\boldsymbol{t}_i)-\frac{1}{2}\right\}^T\left\{\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\otimes\boldsymbol{s}_i^T\right\}vec(\boldsymbol{\Phi}) \\
&+ \sum_i vec(\boldsymbol{\Phi})^T(\boldsymbol{\Theta}_\phi\left\{\boldsymbol{t}_i\right\}^T\otimes\boldsymbol{s}_i\}\,diag\left[\lambda\left\{\xi_i(\boldsymbol{t}_i)\right\}\right]\left\{\boldsymbol{\Theta}_\phi(\boldsymbol{t}_i)\otimes\boldsymbol{s}_i^T\right\}vec(\boldsymbol{\Phi}).
\end{aligned}$$

Maximizing with respect to $\boldsymbol{\Phi}$ gives estimates $\widehat{\boldsymbol{\Phi}}$.

A.1.3 Optimization constraints for the warping step

Section (3.3) refers to optimization constraints for the `R` function `constrOptim()` implemented in the warping step of our algorithm. We constrain inverse warping function to be monotonic with fixed endpoints, and these constraints are enforced through $\boldsymbol{\beta}_i$, the warping function B-spline coefficients for each subject. To ensure that estimated inverse warping functions $\widehat{h}_i^{-1}$ span the same domain as chronological time t_i^*, we fix the outer coefficients $\beta_{i,1}$ and β_{i,K_h}. Thus in practice we estimate the K_h-2 inner spline coefficients $\boldsymbol{\beta}_{i,inner}=(\beta_{i,2},...,\beta_{i,K_h-1})^T$.

To enforce monotonicity of the warping functions we must ensure $\beta_1<\beta_2<...<\beta_{K_h-1}$. Using the notation from the `constrOptim()` function, we define a matrix ui and a vector ci such that

$$ui\times\boldsymbol{\beta}_{i,inner}-ci\geq 0. \tag{A.1}$$

This leads to a ui matrix of dimension $(K_h-1)\times(K_h-2)$ and a size (K_h-1) vector, ci, that take the forms:

$$ui = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ -1 & 1 & 0 & ... & 0 \\ 0 & -1 & 1 & & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & & & -1 & 1 \\ 0 & & & 0 & -1 \end{pmatrix}$$

and

$$ci = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}$$

such that

$$\begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ -1 & 1 & 0 & ... & 0 \\ 0 & -1 & 1 & & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & & & -1 & 1 \\ 0 & & & 0 & -1 \end{pmatrix} \begin{pmatrix} \beta_{i,2} \\ \beta_{i,3} \\ \vdots \\ \beta_{i,K_h-1} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} > \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

A.1.4 Analytic gradient for exponential family registration

For the general exponential family case, this gradient is

$$\frac{dl(Y_i(\boldsymbol{t}_i^*), \boldsymbol{\beta}_i)}{d\boldsymbol{\beta}_i} = \frac{1}{\varphi} \sum_{j=1}^{D_i} \left\{ \left(Y_i(t_{ij}^*) - b' \left[g \left\{ \mu_i(t_{ij}) \right\} \right] \right) \times \Theta_h(t_{ij}^*)^T \Theta_\phi' \{ \boldsymbol{\Theta}_h(t_{ij}^*) \boldsymbol{\beta}_i \} \left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i \right) \right\}, \quad \text{(A.2)}$$

where $\boldsymbol{\Theta}_\phi'(\boldsymbol{t}_i)$ is a $D_i \times K_h$ matrix of first derivatives of the B-spline basis functions used to reconstruct $\boldsymbol{t}_i$, and $b' \left[g \left\{ \mu_i(t_{ij}) \right\} \right] = \mu_i(t_{ij}) = g^{-1} \left[\Theta_\phi \{ \boldsymbol{\Theta}_h(t_{ij}^*) \boldsymbol{\beta}_i \} (\boldsymbol{\alpha}_\Theta + \boldsymbol{\psi}_\Theta \boldsymbol{c}_i) \right]$. For the Bernoulli loss function $\varphi = 1$ and $g^{-1}(z) = \frac{1}{1+e^{-z}}$, so the gradient becomes

$$\begin{aligned} \frac{dl\{Y_i(\boldsymbol{t}_i^*), \boldsymbol{\beta}_i\}}{d\boldsymbol{\beta}_i} &= \sum_{j=1}^{D_i} \left[\left(Y_i(t_{ij}^*) - \frac{1}{1 + e^{-\Theta_\phi \{ \boldsymbol{\Theta}_h(t_{ij}^*) \boldsymbol{\beta}_i \} (\boldsymbol{\alpha}_\Theta + \boldsymbol{\psi}_\Theta \boldsymbol{c}_i)}} \right) \right. \\ &\times \left. \Theta_h(t_{ij}^*)^T \Theta_\phi' \{ \boldsymbol{\Theta}_h(t_{ij}^*) \boldsymbol{\beta}_i \} \left(\boldsymbol{\alpha}_\Theta + \boldsymbol{\Psi}_\Theta \boldsymbol{c}_i \right) \right]. \end{aligned}$$

A.2 Simulations and analysis

Here we provide extra results from simulations and analysis of BLSA data.

A.2.1 Functional principal components for BLSA data

Figure (A.1) shows the effects of the estimated principal component basis functions for the BLSA data after the registration process. The first principal component is a vertical shift around the population mean, $\alpha(t)$, indicating a higher or lower probability of being active. More interesting is the second principal component, which shows that some subjects have higher probability of activity earlier in the day, while others have higher probability of activity later in the day.

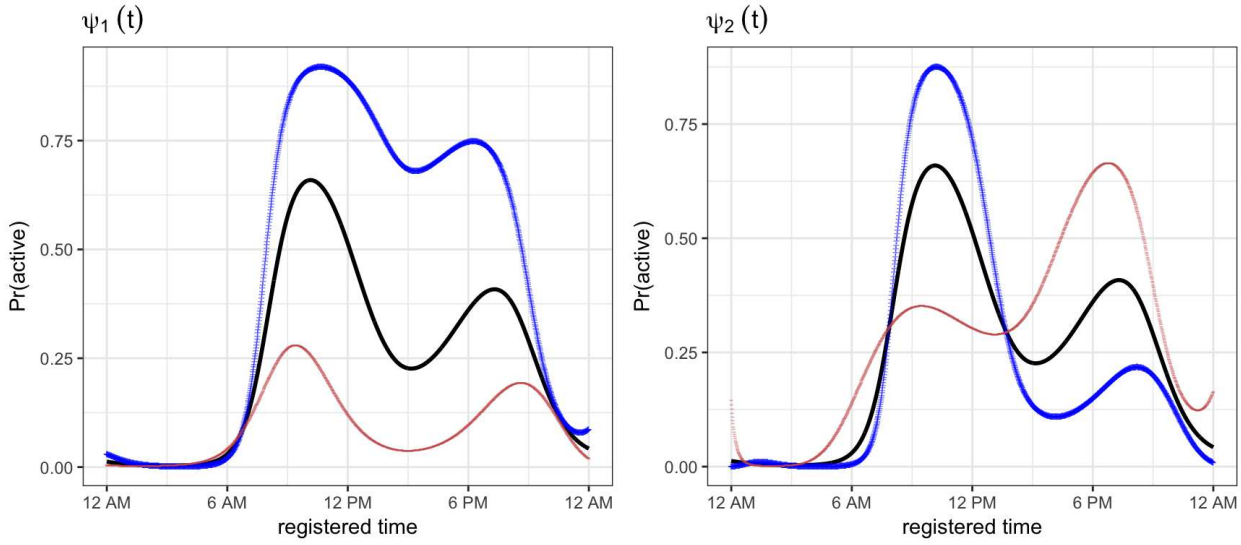

Figure A.1: Estimated binary FPCA basis functions after registration process, illustrated by plotting $g^{-1}\Big[\alpha(t) \pm \psi_k(t)\Big]$ for basis functions $k \in \{1, 2\}$.

A.2.2 Optimizing parameters

As a sensitivity analysis we evaluate our method as a function of parameters K_ϕ and K_h. We evaluated all combinations of $K_\phi \in \{5, 10, 15\}$, $K_h \in \{3, 4, 5, 6\}$ and grid length $D \in \{50, 100, 200\}$ using the same simulation setup and performance metrics as in Section (4). Mean integrated squared errors are given in Figure (A.2) and computation times across these simulation scenarios are given in Figure (A.3).

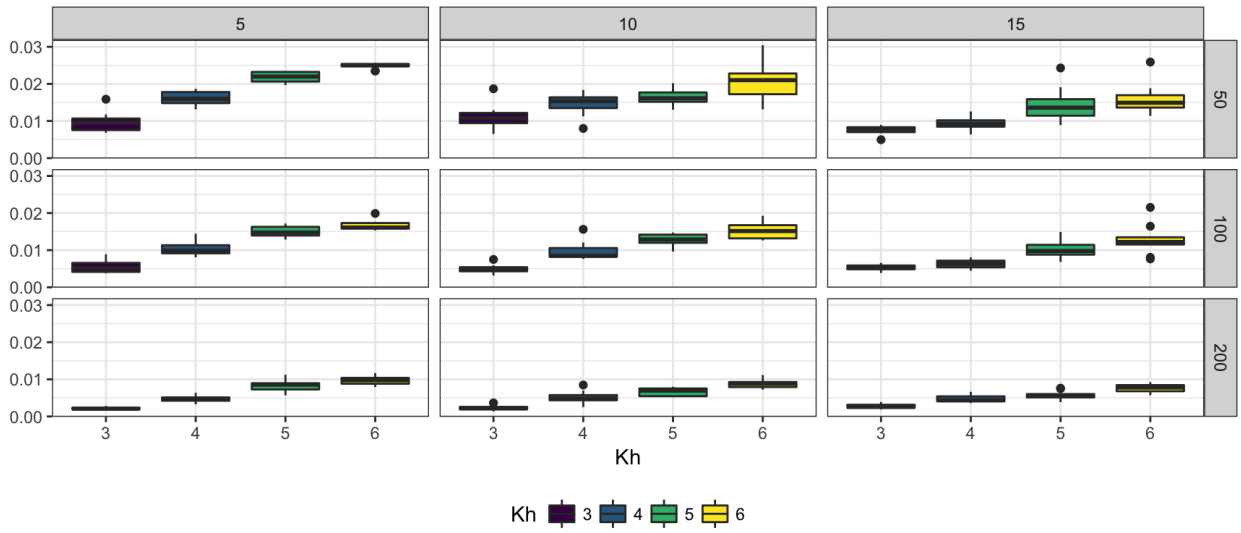

Figure A.2: Parameter sensitivity across values of K_ϕ and K_h for *registr* method. Shown are mean integrated squared error (MISE) summaries across 10 datasets for each parameter scenario. Columns represent distinct values of K_ϕ and rows distinct grid lengths D.

Both MISE and computation time increase linearly with K_h. Mean integrated squared errors decrease slightly with increasing K_ϕ, and computation time slightly increases with increasing K_ϕ.

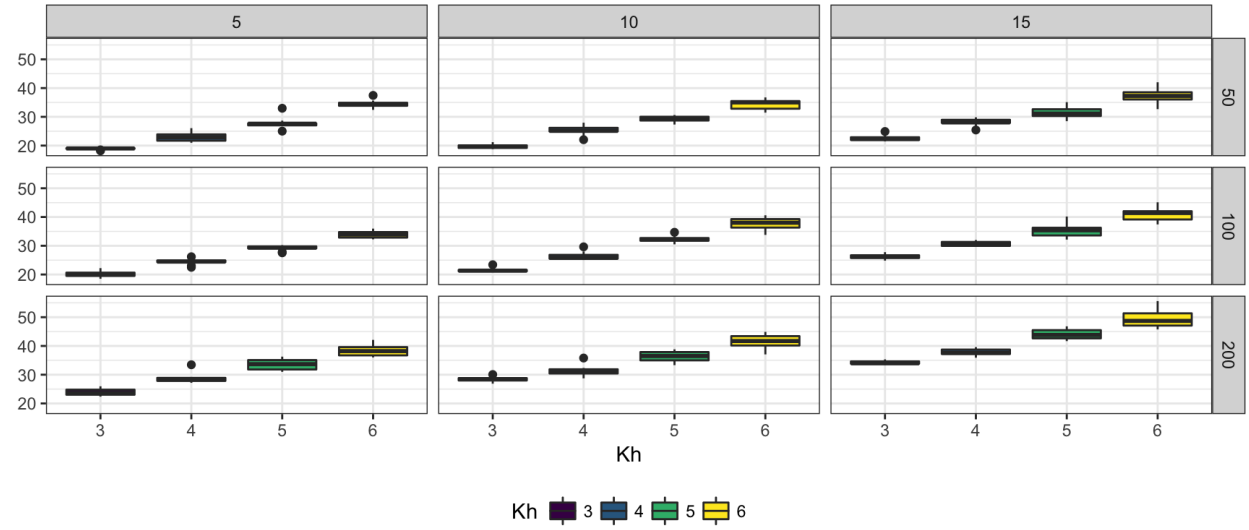

Figure A.3: Parameter sensitivity across values of K_ϕ and K_h for *registr* method. Shown are boxplots of computation time (in seconds) across 10 datasets for each parameter scenario. Columns represent distinct values of K_ϕ and rows distinct grid lengths D.

A.2.3 Sensitivity of BFPCA

Below we compare our binary functional principal components algorithm with that from Hall *et al.* [2008]. Mean integrated squared errors are based on deviations from the population level mean.

A.2.4 Analysis of weekdays for BLSA

In our primary analysis we averaged across visits for subjects. Here we separate visits by day of the week and look at day-specific effects. Figure (A.5) shows unregistered and registered binary and smooth curves for each day of the week. Our algorithm consistently identifies similar patterns across days of the week. Alignment may be slightly better on week days than weekends, which suggests an area for future exploration.

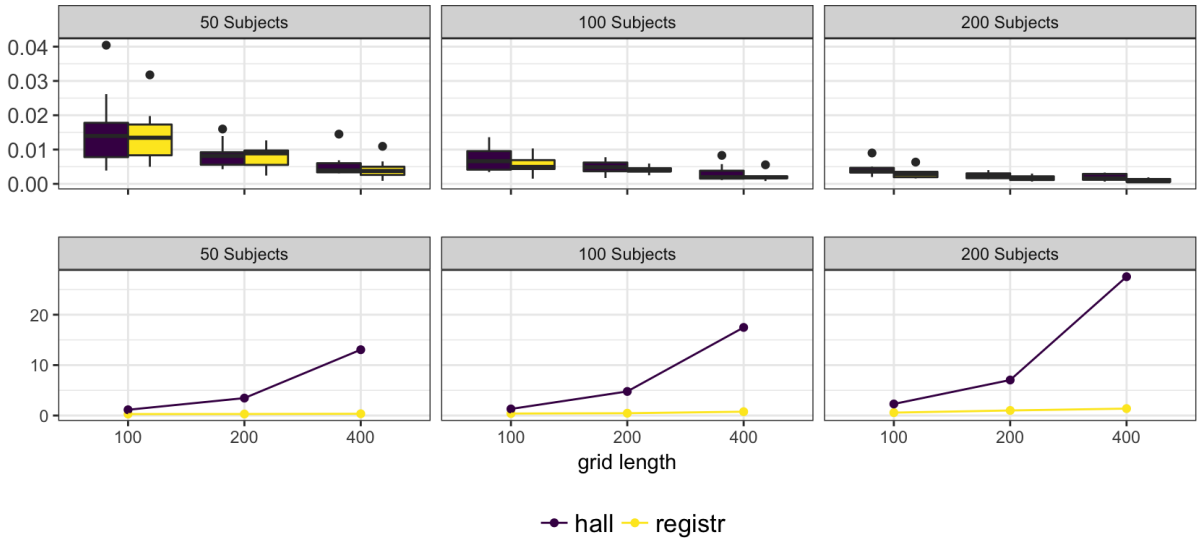

Figure A.4: This figure shows mean integrated squared errors (top row) and median computation times (bottom row) for *hall* (in red) and *registr* (in green) methods across varying sample sizes and grid lengths. The columns, from left to right, show sample sizes 50, 100, and 200, respectively. Within each panel we compare grid lengths of 100, 200, and 400. Mean integrated squared errors are based on deviations from the population level mean.

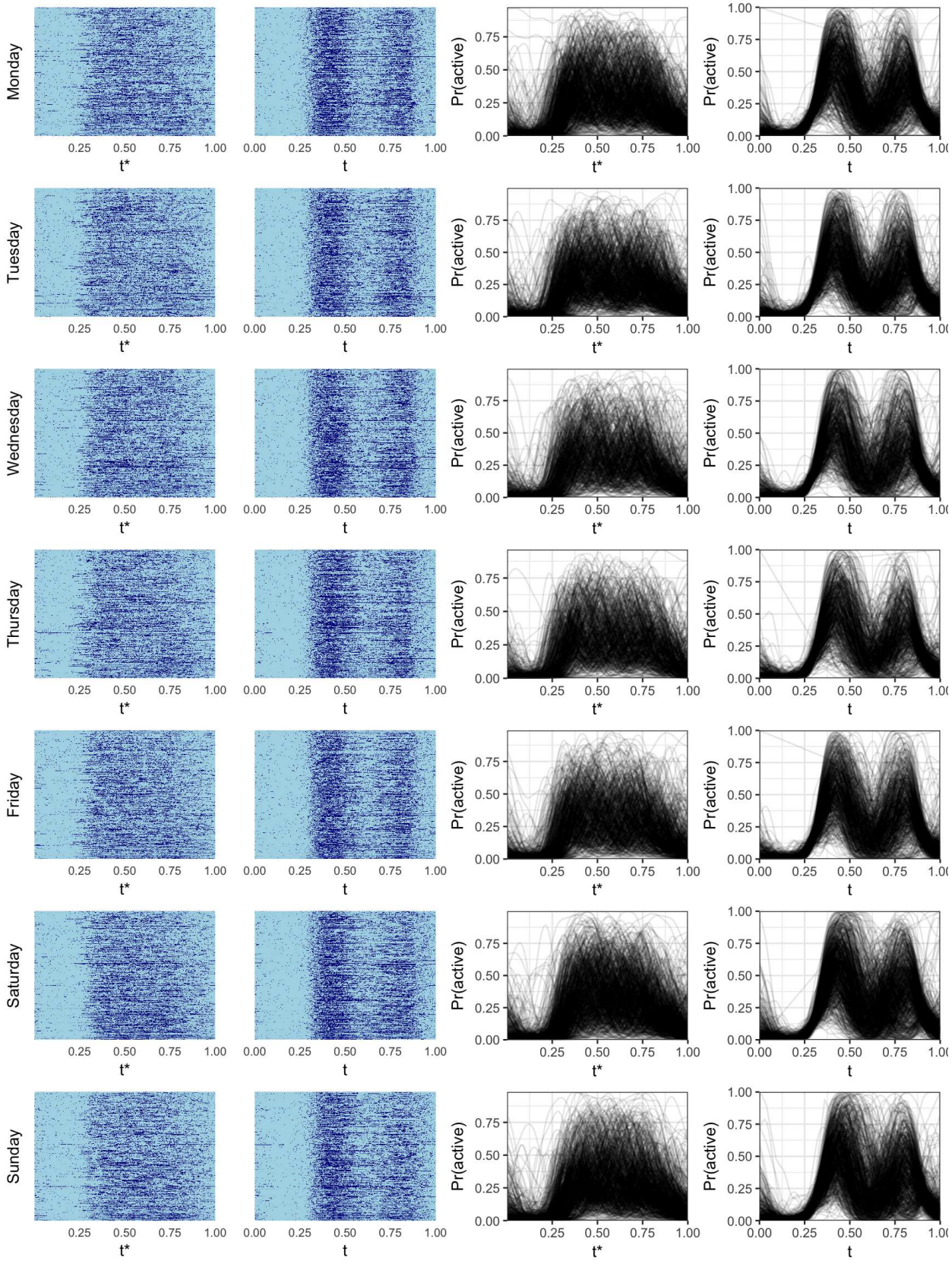

Figure A.5: Analysis results for each day of the week

Appendix B

Appendix to A dynamical systems model for the relationship between the motor cortex and skilled movement

B.1 Data collection

Here we provide additional information about the data collection process. A schematic image of the data collection is shown in Figure B.1. The mouse is positioned at a platform with its head fixed in place to reduce mobility, an auditory cue is triggered, and this cue is timed with the release of a food pellet the mouse then reaches for. Cameras positioned at orthogonal angles capture the paw position over time, and electrodes in the motor cortex capture neural activity. This describes a single trial of the experiment, which was repeated several times.

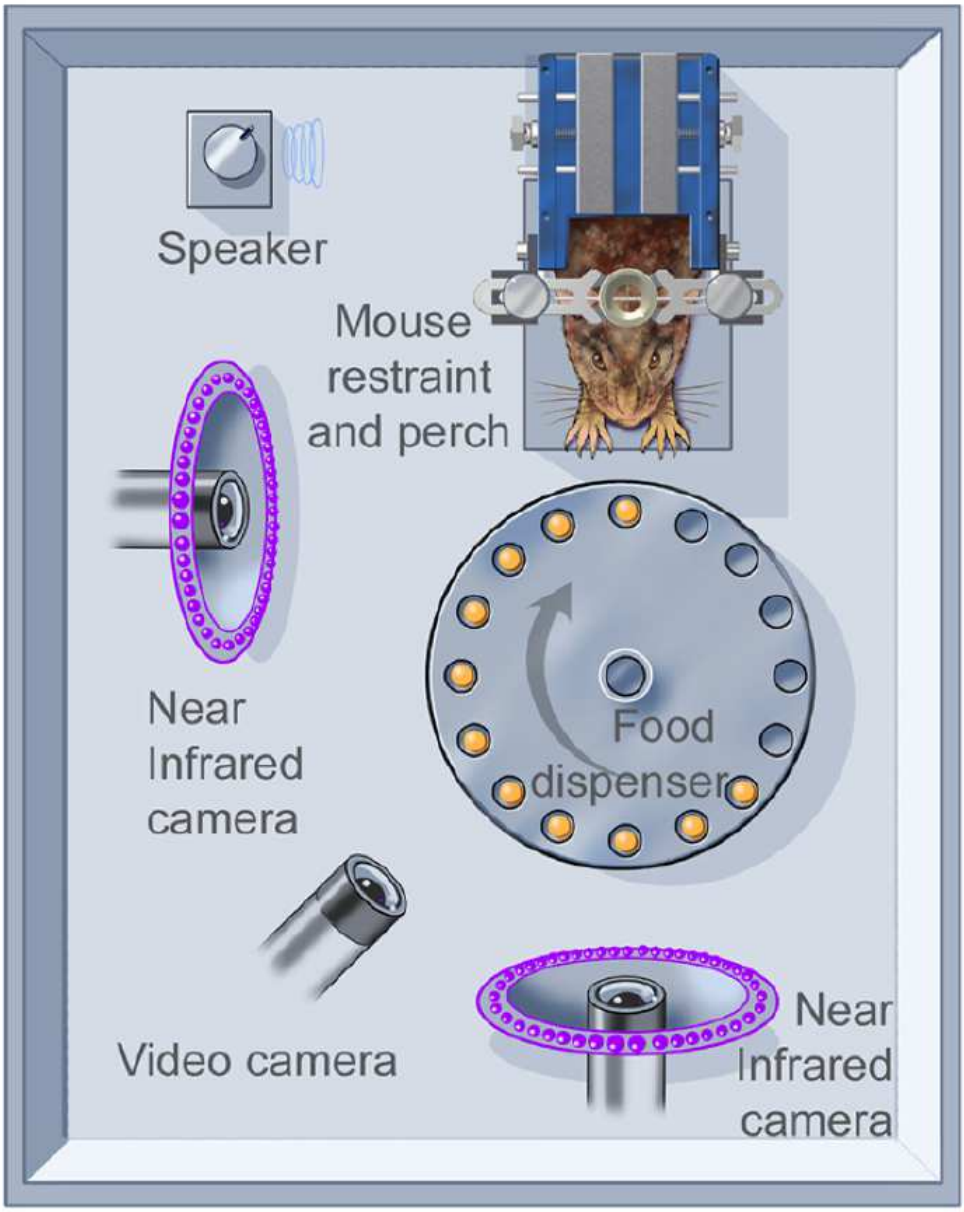

Figure B.1: Experimental setup for paw trajectory data. A mouse is positioned at a platform with its head fixed in place to reduce mobility, an auditory cue is triggered, and this cue is timed with the release of a food pellet the mouse then reaches for. Cameras positioned at orthogonal angles capture the paw position over time, and electrodes in the motor cortex capture neural activity.