


Generalized additive models to analyze biomedical non-linear longitudinal data in R:

Beyond repeated measures ANOVA and Linear Mixed Models

SUPPLEMENTARY MATERIALS: APPENDIX

Ariel I. Mundo 

Department of Biomedical Engineering, University of Arkansas, Fayetteville, AR, USA

John R. Tipton 

Department of Mathematical Sciences, University of Arkansas, Fayetteville, AR, USA

Timothy J. Muldoon*

Department of Biomedical Engineering, University of Arkansas, Fayetteville, AR, USA

tmuldoon@uark.edu

1 Abstract

In biomedical research, the outcome of longitudinal studies has been traditionally analyzed using the *repeated measures analysis of variance* (rm-ANOVA) or more recently, *linear mixed models* (LMEMs). Although LMEMs are less restrictive than rm-ANOVA in terms of correlation and missing observations, both methodologies share an assumption of linearity in the measured response, which results in biased estimates and unreliable inference when they are used to analyze data where the trends are non-linear, which is a common occurrence in biomedical research.

In contrast, generalized additive models (GAMs) relax the linearity assumption, and allow the data to determine the fit of the model while permitting missing observations and different correlation structures. Therefore, GAMs present an excellent choice to analyze non-linear longitudinal data in the context of biomedical research. This paper summarizes the limitations of rm-ANOVA and LMEMs and uses simulated data to visually show how both methods produce biased estimates when used on non-linear data. We present the basic theory of GAMs, and using reported trends of oxygen saturation in tumors we simulate example longitudinal data (2 treatment groups, 10 subjects per group, 5 repeated measures for each group) to demonstrate their implementation in R. We also show that GAMs are able to produce estimates that are consistent with the trends of non-linear data even in the case when missing observations exist (with 40% of the simulated observations missing). To make this work reproducible, the code and data used in this paper are available at: <https://github.com/aimundo/GAMs-biomedical-research>.

Keywords

cancer biology; tumor response; generalized additive models; simulation; R

2 Background

Longitudinal studies are designed to repeatedly measure a variable of interest in a group (or groups) of subjects, with the intention of observing the evolution of effect across time rather than analyzing a single time point (e.g., a cross-sectional study). Biomedical research frequently uses longitudinal studies to analyze the evolution of a “treatment” effect across multiple time points; and in such studies the subjects of analysis range from animals (mice, rats, rabbits), to human patients, cells, or blood samples, among many others. Tumor response [1–4], antibody expression [5,6], and cell metabolism [7,8] are examples of the different situations where researchers have used longitudinal designs to study some physiological response. Because the frequency of the measurements in a longitudinal study is dependent on the biological phenomena of interest and the experimental design of the study, the frequency of such measurements can range from minute intervals to study a short-term response such as anesthesia effects in animals [9], to weekly measurements to analyze a mid-term response like the evolution of dermatitis symptoms in breast cancer patients [10], to monthly measurements to study a long-term response such as mouth opening following radiotherapy (RT) in neck cancer patients [11].

Traditionally, a “frequentist” or “classical” statistical paradigm is used in biomedical research to derive inferences from a longitudinal study. The frequentist paradigm regards probability as the limit of the expected outcome when an experiment is repeated a large number of times [12], and such view is applied to the analysis of longitudinal data by assuming a null hypothesis under a statistical model that is often an *analysis of variance over repeated measures* (repeated measures ANOVA or rm-ANOVA). The rm-ANOVA model makes three assumptions regarding longitudinal data: 1) linearity of the response across time, 2) constant correlation across same-subject measurements, and 3) observations from each subject are obtained at all time points through the study (a condition also known as *complete observations*) [13,14].

The expected linear behavior of the response through time is a key requisite in rm-ANOVA [15]. This “linearity assumption” in rm-ANOVA implies that the model is misspecified when the data does not follow a linear trend, which results in unreliable inference. In biomedical research, non-linear trends are the norm rather than the exception in longitudinal studies. A particular example of this non-linear behavior in longitudinal

data arises in measurements of tumor response to chemo and/or radiotherapy in preclinical and clinical settings [1,8,16]. These studies have shown that the collected signal does not follow a linear trend over time, and presents extreme variability at different time points, making the fit of rm-ANOVA model inconsistent with the observed variation. Therefore, when rm-ANOVA is used to draw inference of such data the estimates are inevitably biased, because the model is only able to accommodate linear trends that fail to adequately represent the biological phenomenon of interest.

A *post hoc* analysis is often used in conjunction with rm-ANOVA to perform repeated comparisons to estimate a *p-value*, which in turn is used as a measure of significance. Although it is possible that a *post hoc* analysis of rm-ANOVA is able to find “significant” *p-values* ($p < 0.05$) from data that shows non-linear trends, the validity of such metric is dependent on how adequate the model fits the data. In other words, *p-values* are valid only if the model and the data have good agreement; if that is not the case, a “Type III” error (known as “model misspecification”) occurs [17]. For example, model misspecification will occur when a model that is only able to explain linear responses (such as rm-ANOVA) is fitted to data that follows a quadratic trend, thereby causing the resulting *p-values* and parameter estimates to be invalid [18].

Additionally, the *p-value* itself is highly variable, and multiple comparisons can inflate the false positivity rate (Type I error or α) [19,20], consequently biasing the conclusions of the study. Corrections exist to address the Type I error issue of multiple comparisons (such as Bonferroni [21]), but they in turn reduce statistical power $(1-\beta)$ [22], and lead to increased Type II error (failing to reject the null hypothesis when it is false) [23,24]. Therefore, the tradeoff of *post hoc* comparisons in rm-ANOVA between Type I, II and III errors might be difficult to resolve in a biomedical longitudinal study where a delicate balance exists between statistical power and sample size.

On the other hand, the assumption of constant correlation in rm-ANOVA (often known as the *compound symmetry assumption*) is typically unreasonable because correlation between the measured responses often diminishes as the time interval between the observation increases [25]. Corrections can be made in rm-ANOVA in the absence of compound symmetry [26,27], but the effectiveness of the correction is limited by the size of the sample, the number of measurements [28], and group sizes [29]. In the case of biomedical research, where living subjects are frequently used, sample sizes are often not “large” due to ethical and budgetary reasons [30] which might cause the corrections for lack of compound symmetry to be ineffective.

Due to a variety of causes, the number of observations during a study can vary between all subjects. For example, in a clinical trial patients may voluntarily withdraw, whereas attrition due to injury or weight loss in preclinical animal studies is possible. It is even plausible that unexpected complications with equipment or supplies arise that prevent the researcher from collecting measurements at certain time points. In each of these missing data scenarios, the *complete observations* assumption of classical rm-ANOVA is violated. When incomplete observations occur, a rm-ANOVA model is fit by excluding all subjects with missing observations from the analysis [13]. This elimination of partially missing data from the analysis can result in increased costs if the desired statistical power is not met with the remaining observations, because it would be necessary to enroll more subjects. At the same time, if the excluded observations contain insightful information that is not used, their elimination from the analysis may limit the demonstration of significant differences between groups.

During the last decade, the biomedical community has started to recognize the limitations of rm-ANOVA in the analysis of longitudinal data. The recognition on the shortcomings of rm-ANOVA is exemplified by the use of linear mixed effects models (LMEMs) by certain groups to analyze longitudinal tumor response data [8,16]. Briefly, LMEMs incorporate *fixed effects*, which correspond to the levels of experimental factors in the study (e.g., the different drug regimens in a clinical trial), and *random effects*, which account for random variation within the population (e.g., the individual-level differences not due to treatment such as weight or age). When compared to the traditional rm-ANOVA, LMEMs are more flexible as they can accommodate missing observations for multiple subjects and allow different modeling strategies for the variability within each measure in every subject [15]. However, LMEMs impose restrictions in the distribution of the random effects, which need to be independent [13,31]. And even more importantly, LMEMs also assume by default a linear relationship between the response and time [15] (polynomial effects can be used with LMEMs, but this approach has its own shortcomings as we discuss in Section 4.2.1) .

As the rm-ANOVA and the more flexible LMEM approaches make overly restrictive assumptions regarding the trend of the response, there is a need for biomedical researchers to explore the use of additional statistical tools that allow the data (and not an assumption in trend) to determine the trend of the fitted model, to enable appropriate inference. In this regard, generalized additive models (GAMs) present an alternative approach to analyze longitudinal data. Although not frequently used by the biomedical community, these semi-parametric models are customarily used in other fields to analyze longitudinal data. Examples of the use of GAMs include the analysis of temporal variations in geochemical and palaeoecological data [32–34], health-environment interactions [35] and the dynamics of government in political science [36]. There are several advantages of GAMs over LMEMs and rm-ANOVA models: 1) GAMs can fit a more flexible class of smooth responses that enable the data to dictate the trend in the fit of the model, 2) they can model non-constant correlation between repeated measurements [37], and 3) can easily accommodate missing observations. Therefore, GAMs can provide a more flexible statistical approach to analyze non-linear biomedical longitudinal data than LMEMs and rm-ANOVA.

The current advances in programming languages designed for statistical analysis (specifically R), have eased the computational implementation of traditional models such as rm-ANOVA and more complex approaches such as LMEMs and GAMs. In particular, R [38] has an extensive collection of documentation and functions to fit GAMs in the package *mgcv* [37,39] that not only speed up the initial stages of the analysis but also enable the use of advanced modeling structures (e.g. hierarchical models, confidence interval comparisons) without requiring advanced programming skills from the user. At the same time, R has many tools that simplify data simulation, an emerging strategy used to test statistical models [28]. Data simulation methods allow the researcher to create and explore different alternatives for analysis without collecting information in the field, reducing the time window between experiment design and its implementation, and simulation can be also used for power calculations and study design questions.

This work provides biomedical researchers with a clear understanding of the theory and the practice of using GAMs to analyze longitudinal data using by focusing on four areas. First, the limitations of LMEMs and rm-ANOVA regarding an expected trend of the response, constant correlation structures, and missing observations are explained in detail. Second, the key theoretical elements of GAMs are presented using clear and simple mathematical notation while explaining the context and interpretation of the equations. Third, we illustrate the type of non-linear longitudinal data that often occurs in biomedical research using simulated data that reproduces patterns in previously reported studies [16]. The simulated data experiments highlight the differences in inference between rm-ANOVA, LMEMs and GAMs on data similar to what is commonly observed in biomedical studies. Finally, reproducibility is emphasized by providing the code to generate the simulated data and the implementation of different models in R, in conjunction with a step-by-step guide demonstrating how to fit models of increasing complexity.

In summary, this work will allow biomedical researchers to identify when the use of GAMs instead of rm-ANOVA or LMEMs is appropriate to analyze longitudinal data, and provide guidance on the implementation of these models to improve the standards for reproducibility in biomedical research.

3 Challenges presented by longitudinal studies

3.1 The repeated measures ANOVA and Linear Mixed Model

The *repeated measures analysis of variance* (rm-ANOVA) and the *linear mixed model* (LMEM) are the most commonly used statistical analysis for longitudinal data in biomedical research. These statistical methodologies require certain assumptions for the model to be valid. From a practical view, the assumptions can be divided in three areas: 1) an assumed relationship between covariates and response, 2) a constant correlation between measurements, and, 3) complete observations for all subjects. Each one of these assumptions is discussed below.

3.2 Assumed relationship

3.2.1 The repeated measures ANOVA case

In a longitudinal biomedical study, two or more groups of subjects (e.g., human subject, mice, samples) are subject to different treatments (e.g., a “treatment” group receives a novel drug or intervention vs. a “control” group that receives a placebo), and measurements from each subject within each group are collected at specific time points. The collected response is modeled with *fixed* components. The *fixed* component can be understood as a constant value in the response which the researcher is interested in measuring, i.e., the average effect of the novel drug/intervention in the “treatment” group.

Mathematically speaking, a rm-ANOVA model with an interaction can be written as

$$y_{ijt} = \beta_0 + \beta_1 \times treatment_j + \beta_2 \times time_t + \beta_3 \times time_t \times treatment_j + \varepsilon_{ijt}, \quad (1)$$

In this model y_{ijt} is the response for subject i , in treatment group j at time t , which can be decomposed in a mean value β_0 , *fixed effects* of treatment ($treatment_j$), time ($time_t$), and their interaction $time_t * treatment_j$ which have linear slopes given by β_1, β_2 and β_3 , respectively. Independent errors ε_{ijt} represent random variation from the sampling process assumed to be $\overset{i.i.d.}{\sim} N(0, \sigma^2)$ (independently and identically normally distributed with mean zero and variance σ^2). In a biomedical research context, suppose two treatments groups are used in a study (e.g., “placebo” vs. “novel drug,” or “saline” vs. “chemotherapy”). Then, the group terms in Equation (1) can be written as below with $treatment_j = 0$ representing the first treatment group (Group A) and $treatment_j = 1$ representing the second treatment group (Group B). With this notation, the linear model then can be expressed as

$$y_{ijt} = \begin{cases} \beta_0 + \beta_2 \times time_t + \varepsilon_{ijt} & \text{if Group A,} \\ \beta_0 + \beta_1 + \beta_2 \times time_t + \beta_3 \times time_t + \varepsilon_{ijt} & \text{if Group B.} \end{cases} \quad (2)$$

To further simplify the expression, substitute $\widetilde{\beta}_0 = \beta_0 + \beta_1$ and $\widetilde{\beta}_1 = \beta_2 + \beta_3$ in the equation for Group B. This substitution allows for a different intercept and slope for Groups A and B. The model is then written as

$$y_{ijt} = \begin{cases} \beta_0 + \beta_2 \times time_t + \varepsilon_{ijt} & \text{if Group A,} \\ \widetilde{\beta}_0 + \widetilde{\beta}_1 \times time_t + \varepsilon_{ijt} & \text{if Group B.} \end{cases} \quad (3)$$

Presenting the model in this manner makes clear that when treating different groups, an rm-ANOVA model is able to accommodate non-parallel lines in each case (different intercepts and slopes per group). In other words, the rm-ANOVA model “expects” a linear relationship between the covariates and the response. This means that either presented as Equations (1), (2) or (3), an rm-ANOVA model is only able to accommodate linear patterns in the data. If the data show non-linear trends, the rm-ANOVA model will approximate this behavior with non-parallel lines.

3.2.2 The Linear Mixed Model (LMEM) Case

A LMEM is a class of statistical models that incorporates *fixed effects* to model the relationship between the covariates and the response, and *random effects* to model subject variability that is not the primary focus of the study but that might be important to account for [15,40]. A LMEM with interaction between time and treatment for a longitudinal study can be written as

$$y_{ijt} = \beta_0 + \beta_1 \times treatment_j + \beta_2 \times time_t + \beta_3 \times time_t \times treatment_j + \alpha_{ij} + \varepsilon_{ijt}. \quad (4)$$

When Equations (1) and (4) are compared, it is noticeable that LMEMs and rm-ANOVA have the same construction regarding the *fixed effects* of time and treatment, but that the LMEM incorporates an additional source of variation (the term α_{ij}). This term α_{ij} corresponds to the *random effect*, accounting for variability

in each subject (subject_{*i*}) within each group (group_{*j*}). The *random* component can also be understood as modeling some “noise” in the response, but that does not arise from the sampling error term ε_{ijt} from Equations (1) through (3).

For example, if the blood concentration of a drug is measured in certain subjects in the early hours of the morning while other subjects are measured in the afternoon, it is possible that the difference in the collection time introduces some “noise” in the data that needs to be accounted for. As the name suggests, this “random” variability needs to be modeled as a variable rather than as a constant value. The random effect α_{ij} in Equation (4) is assumed to be $\mu_{ij} \sim N(0, \sigma_\mu^2)$. In essence, the *random effect* in a LMEM enables fitting models with different intercepts at the subject-level[15]. However, the expected linear relationship of the covariates and the response in Equation (1) and in Equation (4) is essentially the same, representing a major limitation of LMEMs to fit a non-linear response.

Of note, LMEMs are capable of fitting non-linear trends using an “empirical” approach (using polynomial fixed effects instead of linear effects such as in Equation (4)), which is described in detail by Pinheiro and Bates [15]. However, polynomial fits have limited predictive power and cause bias on the boundaries of the covariates [36]; more importantly, polynomial effects lack biological or mechanistic interpretation [15] which limits their use in biomedical studies.

3.3 Covariance in rm-ANOVA and LMEMs

In a longitudinal study there is an expected *covariance* between repeated measurements on the same subject, and because repeated measures occur in the subjects within each group, there is a *covariance* between measurements at each time point within each group. The *covariance matrix* (also known as the variance-covariance matrix) is a matrix that captures the variation between and within subjects in a longitudinal study[41] (For an in-depth analysis of the covariance matrix see West[40] and Weiss[42]).

In the case of an rm-ANOVA analysis, it is typically assumed that the covariance matrix has a specific construction known as *compound symmetry* (also known as “sphericity” or “circularity”). Under this assumption, the between-subject variance and within-subject correlation are constant across time [26,42,43]. However, it has been shown that this condition is frequently not justified because the correlation between measurements tends to change over time [44]; and it is higher between consecutive measurements [13,25]. Although corrections can be made (such as Huynh-Feldt or Greenhouse-Geisser)[26,27] the effectiveness of each correction is limited because it depends on the size of the sample, the number of repeated measurements[28], and they are not robust if the group sizes are unbalanced [29]. Because biomedical longitudinal studies are often limited in sample size and can have an imbalanced design, the corrections required to use an rm-ANOVA model may not be able to provide a reasonable adjustment that makes the model valid.

In the case of LMEMs, one key advantage over rm-ANOVA is that they allow different structures for the variance-covariance matrix including exponential, autoregressive of order 1, rational quadratic and others [15]. Nevertheless, the analysis required to determine an appropriate variance-covariance structure for the data can be a challenging process by itself. Overall, the spherical assumption for rm-ANOVA may not capture the natural variations of the correlation in the data, and can bias the inferences from the analysis.

3.4 Unbalanced data

In a longitudinal study, it is frequently the case that the number of observations is different across subjects. In biomedical research, this imbalance in sample size can be caused by reasons beyond the control of the investigator (such as dropout from patients in clinical studies and attrition or injury of animals in preclinical research) leading to what is known as “missing,” “incomplete,” or (more generally speaking) unbalanced data [45]. The rm-ANOVA model is very restrictive in these situations as it assumes that observations exist for all subjects at every time point; if that is not the case subjects with one or more missing observations are excluded from the analysis. This is inconvenient because the remaining subjects might not accurately represent the population and statistical power is affected by this reduction in sample size [46].

On the other hand, LMEMs and GAMs can work with missing observations, and inferences from the model are valid when the imbalance in the observations are *missing at random* (MAR) or *completely missing at*

random (MCAR) [40,42]. In a MAR scenario, the pattern of the missing information is related to some variable in the data, but it is not related to the variable of interest [47]. If the data are MCAR, this means that the missingness is completely unrelated to the collected information [48]. Missing observations can also be *missing not at random* (MNAR) and in the case the missing observations are dependent on their value. For example, if attrition occurs in all mice that had lower weights at the beginning of a chemotherapy response study, the missing data can be considered MAR because the missingness is unrelated to other variables of interest.

However, it is worth reminding that “all models are wrong” [49] and that the ability of LMEMs and GAMs to work with unbalanced data does not make them immune to problems that can arise due to high rates of missing data, such as sampling bias or a drastic reduction in statistical power. Researchers must ensure that the study design is statistically sound and that measures exist to minimize missing observation rates.

3.5 What do an rm-ANOVA and LMEM fit look like? A visual representation using simulated data

To visually demonstrate the limitations of rm-ANOVA and LMEMs for longitudinal data with non-linear trends, this section presents a simulation experiment of a normally distributed response of two groups of 10 subjects each. An rm-ANOVA model (Equation (1)), and a LMEM (Equation (4)) are fitted to each group, using R [38] and the package *nlme* [50].

Briefly, two cases for the mean response for each group are considered: in the first case, the mean response in each group is a linear function over time with different intercepts and slopes; a negative slope is used for Group 1 and a positive slope is used for Group 2 (Figure 1A). In the second case, a second-degree polynomial (quadratic) function is used for the mean response per group: the quadratic function is concave down for Group 1 and it is concave up for Group 2 (Figure 1C). In both the linear and quadratic simulated data, the groups start with the same mean value at the first time point. This is intentional in order to simulate the expected temporal evolution of some physiological quantity, which is typical in biomedical experiments where a strong non-linear trend is present.

Specifically, the rationale for the chosen linear and quadratic functions is the expectation that a measured response in two treatment groups is similar in the initial phase of the study, but as therapy progresses a divergence in the trend of the response indicates a treatment effect. In other words, Group 1 can be thought as a “Control” group and Group 2 as a “Treatment” group. From the mean response per group (linear or quadratic), the variability or “error” of individual responses within each group is simulated using a covariance matrix with compound symmetry (constant variance across time). Thus, the response per subject in both the linear and quadratic simulation corresponds to the mean response per group plus the error (Figure 1 B,D).

A more comprehensive exploration of the fit of rm-ANOVA and LMEMs for linear and non-linear longitudinal data appears in the Appendix (Figure A.1 and Figure A.2), where a simulation with compound symmetry and independent errors (errors generated from a normal distribution that are not constant over time) is presented. We are aware that the simulated data used in this section present an extreme case that might not occur frequently in biomedical research, but they are used to 1) present the consequences of modeling non-linear trends in data with a linear model such as rm-ANOVA or a LMEM with “default” (linear) effects and, 2) demonstrate that a visual assessment of model fit is an important tool that helps determine the validity of any statistical assumptions. In Section 5 we use simulated data that does follow reported trends in the biomedical literature to implement GAMs.

The simulation shows that the fits produced by the LMEM and the rm-ANOVA model are good for linear data, as the predictions for the mean response are reasonably close to the “truth” of the simulated data (Figure 1A). When the linearity and compound symmetry assumptions are met, the rm-ANOVA model approximates well the global trend by group (Figure 1B). Note that because the LMEM incorporates *random effects*, is able to provide estimates for each subject and a “population” estimate (Figure 1C).

However, consider the case when the data follows a non-linear trend, such as the simulated data in Figure 1D. Here, the mean response per group was simulated using a quadratic function, and errors and individual responses were produced as in Figure 1A. The mean response in the simulated data with quadratic behavior

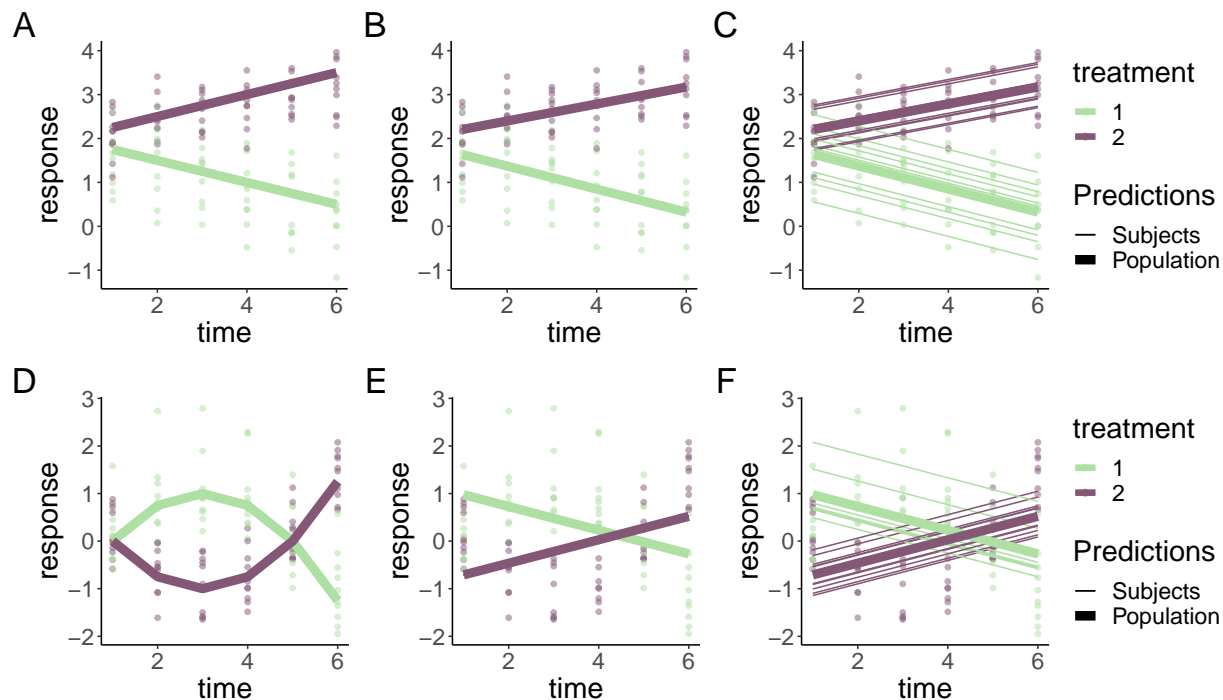


Figure 1: Simulated responses from two groups with correlated errors using a LMEM and a rm-ANOVA model. Top row: linear response, bottom row: quadratic response. **A**: Simulated linear data with known mean response (thick lines) and individual responses (points) showing the dispersion of the data. **D**: Simulated quadratic data with known mean response (thick lines) and individual responses (points) showing the dispersion of the data. **B,E**: Estimates from the rm-ANOVA model for the mean group response (linear of quadratic). Points represent the original raw data. The rm-ANOVA model not only fails to pick the trend of the quadratic data (D) but also assigns a global estimate that does not take between-subject variation. **C, F**: Estimates from the LMEM in the linear and quadratic case (subject: thin lines, population: thick lines). The LMEM incorporates a random effect for each subject, but this model and the rm-ANOVA model are unable to follow the trend of the data and grossly bias the initial estimates for each group in the quadratic case (bottom row).

changes in each group through the timeline, and the mean value is the same as the initial value by the fifth time point for each group. Fitting an rm-ANOVA model (Equation (1)) or a LMEM (Equation (4)) to this data produces the fit that appears in Figure 1E, F.

Comparing the fitted responses of the LMEM and the rm-ANOVA models used in the simulated quadratic data (Figure 1E, F) indicates that the models are not capturing the changes within each group. Specifically, note that the fitted mean response of both models shows that the change (increase for Treatment 1 or decrease for Treatment 2) in the response through time points 2 and 4 is not being captured.

The LMEM is only able to account for between-subject variation by providing estimates for each subject (Figure 1F), but both models are unable to capture the fact that the initial values are the same in each group, and instead fit non-parallel lines that have initial values that are markedly different from the “true” initial values in each case (compare Figure 1D with Figure 1E, F). If such a change has important physiological implications, both rm-ANOVA and LMEMs omit it from the fitted mean response. Thus, even though the model correctly detects a divergence between treatment groups, the exact nature of this difference is not correctly identified, limiting valuable inferences from the data. It could be argued that a LMEM with quadratic effects should have been used to fit the data in Figure 1F. However, because in reality the true function is not known, choosing a polynomial degree causes more questions (e.g., is it quadratic?, cubic?, or a higher degree?). Additionally, polynomial effects have other limitations, which we cover in Section 4.2.1.

This section has used simulation to better convey and visualize the limitations of linearity and correlation in the response in data with non-linear trends using an rm-ANOVA model and a LMEM, where the main issue is the expected linear trend in the response. Notice that the model misspecification is easily noticeable if the model fit and the response are visualized. In the following section, we provide a brief overview of linear models, general linear models and generalized linear mixed models before presenting the theory of GAMs.

4 Linear Models, and beyond

Linear models (LMs) are those that assume a normal (Gaussian) distribution of the errors, and only incorporate *fixed effects* (such as rm-ANOVA). These are by far the models most commonly used to analyze data within the biomedical research community. On the other hand, Linear Mixed Effect Models (LMEMs) also incorporate *random effects*, as it has been described in Section 3.2.2.

In reality, rm-ANOVA and LMEMs are just *special cases* of a broader class of models (General Linear Models and Generalized Linear Mixed Models, respectively). In order to fully capture the constraints of such models and to understand how GAMs overcome those limitations this section will briefly provide an overview of the different classes of models, indicate how rm-ANOVA and LMEMs fit within this framework and introduce the theory of GAMs.

4.1 Generalized Linear Models (GLMs)

A major limitation of LMs is their assumption of normality in the errors. If the residuals are non-normal, a transformation is necessary in order to properly fit the model. However, transformation can lead to poor model performance [51], and can cause problems with the biological interpretation of the model estimates. McCullagh and Nelder [52] introduced General Linear Models (GLMs) as an extension of LMs, where the errors do not need to be normally distributed. To achieve this, consider the following model

$$y_{ijt} \sim \mathcal{D}(\mu_{ijt}, \phi), \quad (5)$$

where y_{ijt} is the observation i in group j at time t , that is assumed to come from some distribution of the exponential family \mathcal{D} , with some mean μ_{ijt} and potentially, a dispersion parameter ϕ (which in the Gaussian case is the variance σ^2). The mean (μ_{ijt}) is also known as the *expected value* (or *expectation*), $E(y_{ijt})$ of the observed response y_{ijt} .

Then, the *linear predictor* η , which defines the relationship between the mean and the covariates can be defined as

$$\eta_{ijt} = \beta_0 + \beta_1 \times treatment_j + \beta_2 \times time_t + \beta_3 \times time_t \times treatment_j, \quad (6)$$

where η_{ijt} is the linear predictor for each observation in each group at each timepoint, and β_0 (the intercept), β_1, β_2 , and β_3 are the model parameters for each group, which can be referred globally to as β_j . Finally, $time_{ijt}$ represents the covariates from each subject in each group at each time point.

Finally,

$$E(y_{ijt}) = \mu_{ijt} = g^{-1}(\eta_{ijt}), \quad (7)$$

where $E(y_{ijt})$ is the expectation, and g^{-1} is the inverse of a *link function* (g). The link function transforms the values from the response scale to the scale of the linear predictor η (Equation (6)). Therefore, it can be seen that LMs (such as rm-ANOVA) are a special case of GLMs where the response is normally distributed.

4.2 Generalized linear mixed models (GLMMs)

Although GLMs relax the normality assumption, they only accommodate fixed effects. Generalized Linear Mixed Models (GLMMs) are an extension of GLMs that incorporate *random effects*, which have an associated probability distribution [53]. Therefore, in GLMMs the linear predictor takes the form

$$\eta_{ijt} = \beta_0 + \beta_1 \times treatment_j + \beta_2 \times time_t + \beta_3 \times time_t \times treatment_j + \alpha_{ij}, \quad (8)$$

where α_{ij} corresponds to the random effects that can be estimated within each subject in each group, and all the other symbols correspond to the notation of Equation (6). Therefore, LMEMs are special case of GLMMs where the distribution of the response is normally distributed [52], and GLMs are a special case of GLMMs where there are no random effects. In-depth and excellent discussions about LMs, GLMs and GLMMs can be found in Dobson [54] and Stroup [55].

4.2.1 GAMs as a special case of Generalized Linear Models

4.2.1.1 GAMs and Basis Functions Notice that in the previous sections, the difference between GLMs and GLMMs resides on their linear predictors (Equations (6), (8)). Generalized additive models (GAMs) are an extension of the GLM family that allow the estimation of smoothly varying trends where the relationship between the covariates and the response is modeled using *smooth functions* [34,37,56]. In a GAM, the linear predictor has the form

$$\eta_{ijt} = \beta_0 + \beta_1 \times treatment_j + f(time_t | \beta_j). \quad (9)$$

Where β_0 is the intercept, and β_1 is the coefficient for each treatment group. Notice that the construction of the predictor is similar to that of Equation (6), but in this case the parametric terms involving the effect of time have been replaced by $f(time_t | \beta_j)$, which represents the smooth function of time with inputs as the covariates $time_t$ and parameters β_j . A GAM version of a linear model can be written as:

$$y_{ijt} = \beta_0 + f(time_t | \beta_j) + \varepsilon_{ijt} \quad (10)$$

Where y_{ijt} is the response at time t of subject i in group j , β_0 is the expected value at time 0, the change of y_{ijt} over time is represented by the smooth function $f(time_t | \beta_j)$, and ε_{ijt} represents the deviation of each observation from the mean.

In contrast to the linear functions used to model the relationship between the covariates and the response in rm-ANOVA or LMEM, the use of smooth functions in GAMs is more advantageous as it allows more flexibility because it does not restrict the model to a linear relationship, although a GAM can estimate a linear relationship if the data is consistent with a linear response. One possible set of functions for $f(time_t | \beta_j)$ that

allow for non-linear responses are polynomials (which can also be used in LMEMs), but a major limitation is that polynomials create a “global” fit as they assume that the same relationship exists everywhere, which can cause problems with inference [36]. In particular, polynomial fits are known to show boundary effects because as t goes to $\pm\infty$, $f(\text{time}_t | \beta_j)$ goes to $\pm\infty$ which is almost always unrealistic and causes bias at the endpoints of the time period.

The smooth functional relationship between the covariates and the response in GAMs is specified using a semi-parametric relationship that can be fit within the GLM framework, by using a *basis function* expansion of the covariates and by estimating random coefficients associated with these basis functions. A *basis* is a set of functions that spans the mathematical space within which the true but unknown $f(\text{time}_t)$ is thought to exist [34]. For the linear model in Equation (1), the basis coefficients are β_1 , β_2 and β_3 and the basis vectors are treatment_j , time_t , and $\text{time}_t \times \text{treatment}_j$. The basis function then, is the linear combination of basis coefficients and basis vectors that map the possible relationship between the covariates and the response [57], which in the case of Equation (1) is restricted to a linear family of functions. In the case of Equation (10), the basis functions are contained in the expression $f(\text{time}_t | \beta_j)$, which means that the model allows for non-linear relationships among the covariates.

Splines (which derive their name from the physical devices used by draughtsmen to draw smooth curves) are commonly used as basis functions as they have a long history in solving semi-parametric statistical problems and are often a default choice to fit GAMs as they are a simple, flexible, and powerful option to obtain smoothness [58]. Although different types of splines exist, cubic, thin plate splines, and thin plate regression splines will be briefly discussed next to give a general idea of these type of basis functions, and their use within the GAM framework.

Cubic splines (CS) are smooth curves constructed from cubic polynomials joined together in a manner that enforces smoothness. The use of CS as smoothers in GAMs was discussed within the original GAM framework [56], but they are limited by the fact that their implementation requires the selection of some points along the covariates (known as ‘knots,’ the points where the basis functions meet) to obtain the finite basis, which could affect the model fit [59]. A solution to the “knot” placement of CS is provided by thin plate splines (TPS), which provide optimal smooth estimation without knot placement, but that are computationally costly to calculate [37,59].

In contrast, thin plate regression splines (TPRS) provide a reasonable “low rank” (truncated) approximation to the optimal TPS estimation, which can be implemented in an efficient computational manner [59]. Like TPS, TPRS only require the number of basis functions to be used to create the smoother (for mathematical details on both TPS and TPRS see Wood[37,59]).

To further clarify the concept of basis functions and smooth functions, consider the simulated response for Group 1 in Figure 1C. The simplest GAM model that can be used to estimate such response is that of a single smooth term for the time effect; i.e., a model that fits a smooth to the trend of the group through time. A computational requisite in *mgcv* is that the number of basis functions to be used to create the smooth cannot be larger than the number of unique values from the independent variable. Because the data has six unique time points, we can specify a maximum of six basis functions (including the intercept) to create the smooth. It is important to note that is not necessary to specify a number of basis equal to the number of unique values in the independent variable; fewer basis functions can be specified to create the smooth as well, as long as they reasonably capture the trend of the data.

Here, the main idea is that the resulting smooth matches the data and approximates the true function without becoming too “wiggly” due to the noise present. A detailed exploration of wiggleness and smooth functions is beyond the scope of this manuscript, but in essence controlling the wiggleness (or “roughness”) of the fit is achieved by using a *smoothness parameter* (λ), which is used to penalize the likelihood by multiplying it with the integrated square of the second derivative of the spline. The second derivative of the spline is a measure of curvature, or the rate of change of the slope [34,37], and increasing the penalty by increasing λ results in models with less curvature. As λ increases, the parameter estimates are penalized (shrunk towards 0) where the penalty reduces the wiggleness of the smooth fit to prevent overfitting. In other words, a low penalty estimate will result in wiggly functions whereas a high penalty estimate provides evidence that a linear response is appropriate.

With this in mind, if five basis functions are used to fit a GAM for the data that appears in Figure 1C, the resulting fitting process is shown in Figure 2A. The four basis functions (and the intercept) are shown. Each of the basis functions is evaluated at six different points (because there are six points on the timeline). The coefficients for each of the basis functions of Figure 2A are estimated using a penalized regression with smoothness parameter λ , that is estimated when fitting the model. The penalized estimates fitted for our example are shown in Figure 2B.

To get the weighted basis functions, each basis (from Figure Figure 2A) is multiplied by the corresponding coefficients in Figure 2B, thereby increasing or decreasing the original basis functions. Figure 2C shows the resulting weighted basis functions. Note that the magnitude of the weighting for the first basis function has resulted in a decrease of its overall contribution to the smoother term (because the coefficient for that basis function is negative and less than 1). On the other hand, the third basis function has roughly doubled its contribution to the smooth term. Finally, the weighted basis functions are added at each timepoint to produce the smooth term. The resulting smooth term for the effect of *time* is shown in Figure 2D (orange line), along the simulated values per group, which appear as points.

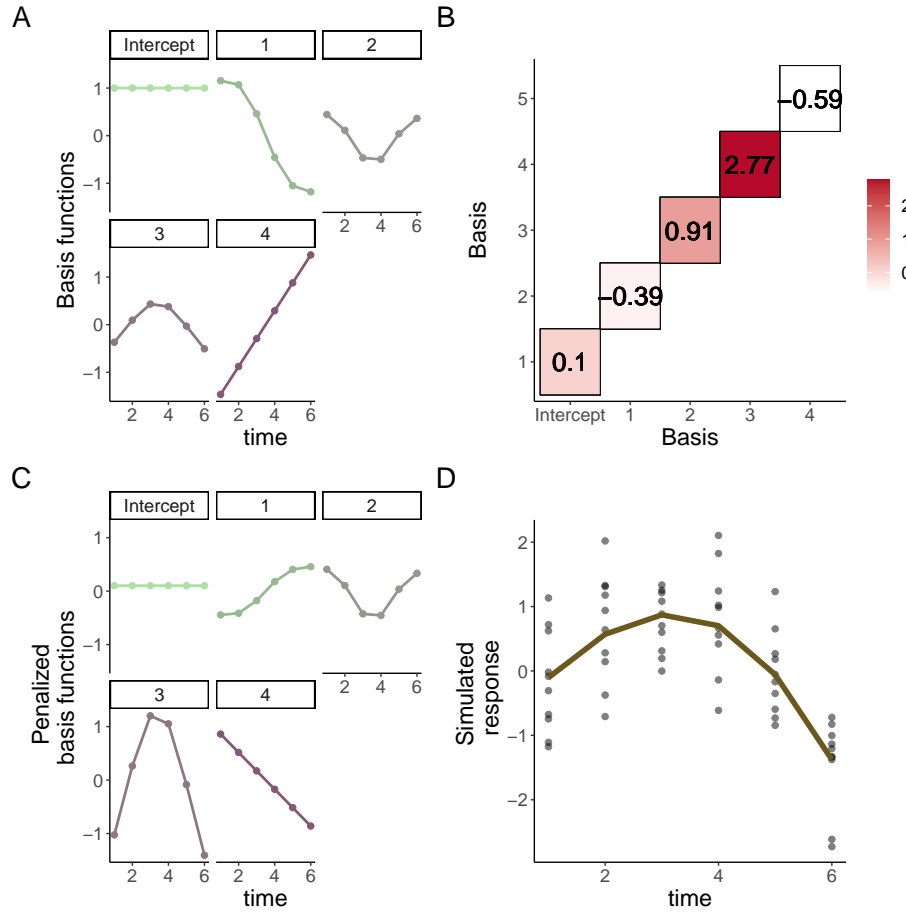


Figure 2: Basis functions for a single smoother for time. **A:** Basis functions for a single smoother for time for the simulated data of Group 1 from Figure 2. **B:** Matrix of basis function weights. Each basis function is multiplied by a coefficient which can be positive or negative. The coefficient determines the overall effect of each basis in the final smoother. **C:** Weighted basis functions. Each of the four basis functions of panel A has been weighted by the corresponding coefficient shown in Panel B. Note the corresponding increase (or decrease) in magnitude of each weighted basis function. **D:** Smoother for time and original data points. The smoother (line) is the result of the sum of each weighted basis function at each time point, with simulated values for the group shown as points.

4.2.2 A Bayesian interpretation of GAMs

Bayes’ theorem states that the probability of an event can be calculated using prior knowledge [60]. In the case of data that shows non-linear trends, the prior that the *true* trend of the data is likely to be smooth rather than “wiggly” introduces the concept of a prior distribution for wiggleness (and therefore a Bayesian view) of GAMs [37]. Moreover, GAMs are considered “empirical” Bayesian models when fitted using the package *mgcv* because the smoothing parameters are estimated from the data (and not from a posterior distribution as in the “fully Bayesian” case, which can be fitted using JAGS, Stan, or other probabilistic programming language) [61]. Therefore, the confidence intervals (CIs) calculated by default for the smooth terms using *mgcv* are considered empirical Bayesian credible intervals [33], which have good *across the function* (“frequentist”) coverage [37].

To understand this last part, it is worth reminding that a CI provides an estimate of the region where the “true” or “mean” value of a function exists, taking into account the randomness introduced by the sampling process. Because random samples from the population are used to calculate the “true” value of the function, there is inherent variability in the estimation process and the CI provides a region with a nominal value (usually, 95%) where the function is expected to lie. In an *across the function* CI (like those estimated by default for GAMs using *mgcv*) if we average the coverage of the interval over the entire function we get approximately the nominal coverage (95%). In other words, we expect that about 95% of the points that compose the true function will be covered by the across the function CI and for this to occur, some areas of the function must have more than nominal coverage and some areas less than the nominal coverage.

However, besides the across the function CI, “simultaneous” or “whole function” CIs can also be computed, which contain the *whole function* with a specified probability [37]. Suppose we chose a nominal value (say, 95%) and compute a simultaneous CI; if we obtain 100 repeated samples and compute a simultaneous CI in each case, we would expect that the true function lies completely within the computed simultaneous CI in 95 of those repeated samples. Briefly, to obtain a simultaneous CI we use simulation to obtain the maximum absolute standardized deviation of the model estimates, which is used to correct the coverage of the across the function CI [62]. Overall, a simultaneous CI provides more robust estimates that can be used to make comparisons between different groups, as we discuss in the following section.

In-depth theory of the Bayesian interpretation of GAMs and details on the computation of simultaneous and across the function CIs are beyond the scope of this paper, but can be found in Miller [61], Wood [37], Simpson [34], Marra [63], and Ruppert [62]. With this brief introduction to the Bayesian interpretation of GAMs, in the next section we consider the use of GAMs to analyze longitudinal biomedical data with non-linear trends.

5 The analysis of longitudinal biomedical data using GAMs

The previous sections provided the basic framework to understand the GAM framework and how these models are more advantageous to analyze non-linear longitudinal data when compared to rm-ANOVA or LMEMs. This section will use simulation to present the practical implementation of GAMs for longitudinal biomedical data using R and the package *mgcv*. The code for the simulated data and figures, and a brief guide for model selection and diagnostics appear in the Appendix.

5.1 Simulated data

The simulated data is based on the reported longitudinal changes in oxygen saturation (StO_2) in subcutaneous tumors (Figure 3C in Vishwanath et. al.[16]). In the paper, diffuse reflectance spectroscopy was used to quantify StO_2 changes in both groups at the same time points (days 0, 2, 5, 7 and 10). In the “Treatment” group (chemotherapy) an increase in StO_2 is observed through time, while a decrease is seen in the “Control” (saline) group. Following the reported trend, we simulated 10 normally distributed observations at each time point with a standard deviation (SD) of 10% (matching the SD in the original paper). The simulated and real data appear in Figure 3A.

5.2 An interaction GAM for longitudinal data

An interaction effect is typically the main interest in longitudinal biomedical data, as it takes into account treatment, time, and their combination. In a practical sense, when a GAM is implemented for longitudinal data, a smooth can be added to the model for the *time* effect to account for the repeated measures over time. Although specific methods of how GAMs model correlation structures is a topic beyond the scope of this paper, it suffices to say that GAMs are flexible and can handle correlation structures beyond compound symmetry. A detailed description on basis functions and correlations can be found in Hefley [57].

For the data in Figure 3A the main effect of interest is how StO₂ changes over time for each treatment. To estimate this, the model incorporates separate smooths for *Group* and *Day*, respectively. The main thing to consider is that model syntax accounts for the fact that one of the variables is numeric (*Day*) and the other is a factor (*Group*). Because the smooths are centered at 0, the factor variable needs to be specified as a parametric term in order to identify any differences between the group means. Using R and the package *mgcv* the model syntax is:

```
gam_02 <- gam(StO2_sim ~ Group + s(Day, by=Group, k=5), method='REML',  
  data = dat_sim)
```

This syntax specifies that `gam_02` (named this way so it matches the model workflow from the Appendix) contains the fitted model, and that the change in the simulated oxygen saturation (`StO2_sim`) is modeled using independent smooths over `Day` for each `Group` (the parenthesis preceded by `s`) using four basis functions (plus the intercept). The smooth is constructed by default using TPRS, but other splines can be used if desired, including Gaussian process smooths [34] (a description of all the available smooths can be found by typing `?mgcv::smooth.terms` in the Console).

The parametric term `Group` is added to quantify overall mean differences in the effect of treatment between groups. Although the default `method` used to estimate the smoothing parameters in *mgcv* is generalized cross validation (GCV), Wood[37] showed the restricted maximum likelihood (REML) to be more resistant to overfitting while also easing the quantification of uncertainty in the smooth parameters; therefore in this manuscript REML is always used for smooth parameter estimation. An additional argument (`family`) allows to specify the expected distribution of the response, but it is not used in this model because we expect a normally-distributed response (which is the default `family` in *mgcv*).

When the smooths are plotted over the raw data, it is clear that the model has been able to capture the trend of the change of StO₂ for each group across time (Figure 3B). Model diagnostics can be obtained using the `gam.check` function, and the function `appraise` from the package *gratia* [64]. A guide for model selection and diagnostics is in the Appendix, and an in-depth analysis can be found in Wood [37] and Harezlak [65].

One question that might arise at this point is “what is the fit that an rm-ANOVA model produces for the simulated data?” The rm-ANOVA model, which corresponds to Equation (1) is presented in Figure 3C. This is a typical case of model misspecification: The slopes of each group are different, which would lead to a *p-value* indicating significance for the treatment and time effects, but the model is not capturing the changes that occur at days 2 and between days 5 and 7, whereas the GAM model is able to reliably estimate the trend over all timepoints (Figure 3B).

Because GAMs do not require equally-spaced or complete observations for all subjects (as rm-ANOVA does), they are advantageous to analyze longitudinal data where missingness exists. The rationale behind this is that GAMs are able to pick the trend in the data even when some observations are missing. However, this usually causes the resulting smooths to have wider confidence intervals and less ability to pick certain trends. Consider the simulated StO₂ values from Figure 3B. If 40% of the observations are randomly deleted and the same interaction GAM (`gam_02`) is fitted, the resulting smooths appear in (Figure 3D). Note that the model is still able to show a different trend for each group, but with a somewhat more linear profile in some areas.

Additionally, note that in Figures 3B,D we show two CIs for each of the fitted smooths (shaded regions). The across the function CIs are represented by the narrow regions and because the simultaneous CIs contain the whole function on a nominal value, we expect them to be wider than the across the function CI, resulting in the wide shaded regions. For the dataset with missing observations, the smooths CIs overlap during the first

3 days because the estimates are less robust with fewer data points, and the trend is less pronounced than in the full dataset. However, the overall trend of the data is picked by the model in both cases, with as few as 4 observations per group at certain time points.

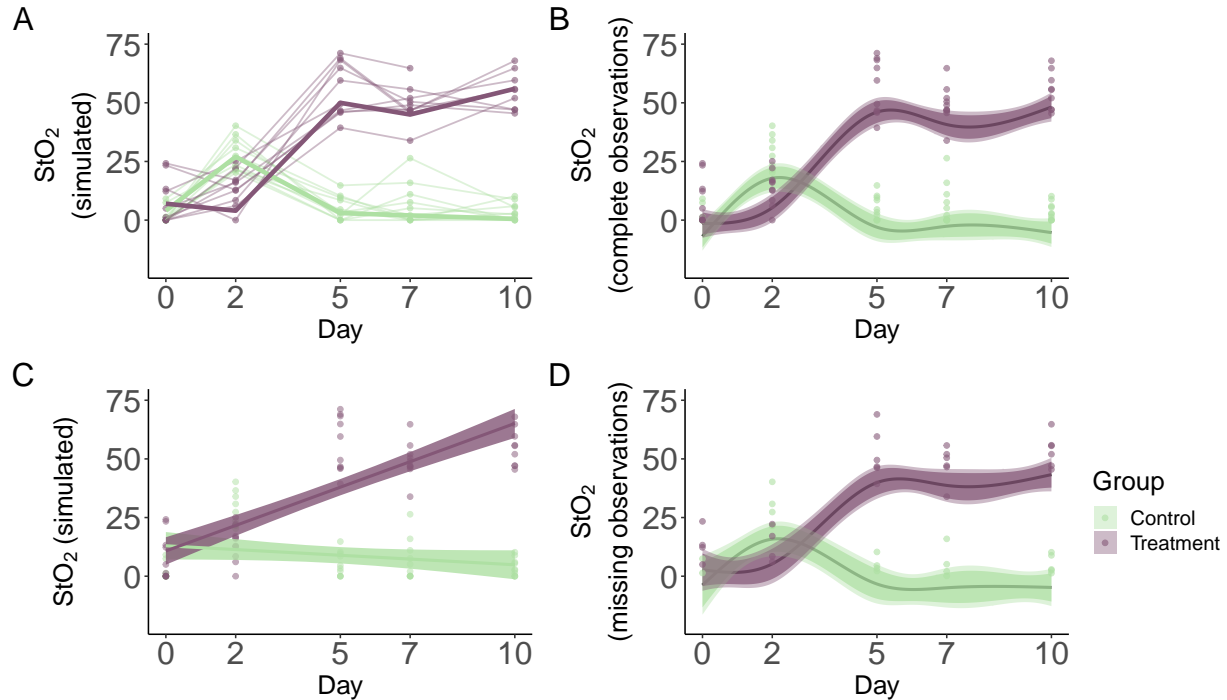


Figure 3: Simulated data and smooths for oxygen saturation in tumors. **A:** Simulated data (thin lines) that follows previously reported trends (thick lines) in tumors under chemotherapy (Treatment) or saline (Control) treatment. Simulated data is from a normal distribution with standard deviation of 10% with 10 observations per time point. **B:** Smooths from the GAM model for the full simulated data with interaction of Group and Treatment. Lines represent trends for each group, shaded regions are 95% across the function (narrow region) and simultaneous (wide region) confidence intervals. **C:** The rm-ANOVA model for the simulated data, which does not capture the changes in each group over time. **D:** Smooths for the GAM model for the simulated data with missing observations (40%). Lines represent trends for each group, shaded regions are 95% across the function (narrow region) and simultaneous (wide region) confidence intervals.

5.3 Determination of significance in GAMs for longitudinal data

At the core of a biomedical longitudinal study lies the question of a significant difference between the effect of two or more treatments in different groups. Whereas in rm-ANOVA a *post-hoc* analysis is required to answer such question by calculating some *p-values* after multiple comparisons, GAMs can use a different approach to estimate significance. In essence, the idea behind the estimation of significance in GAMs across different treatment groups in a model where separate smoothers exists per group (such as in `gam_02`) is that the difference between them can be computed, followed by the estimation of a confidence interval around this difference.

The absence of a *p-value* in this context might seem odd, but its interpretation can be conceptualized in the following manner: Different time trends in each group are an indication of an effect by the treatment. This is what happens for the simulated data in Figure 3A, where the chemotherapy causes StO₂ to increase over time.

With this expectation of different trends in each group, computing the difference between the trends will identify if the observed difference is significant. A difference between groups with similar trends is unlikely to be distinguishable from zero, which would indicate that the treatment is not causing a change in the response

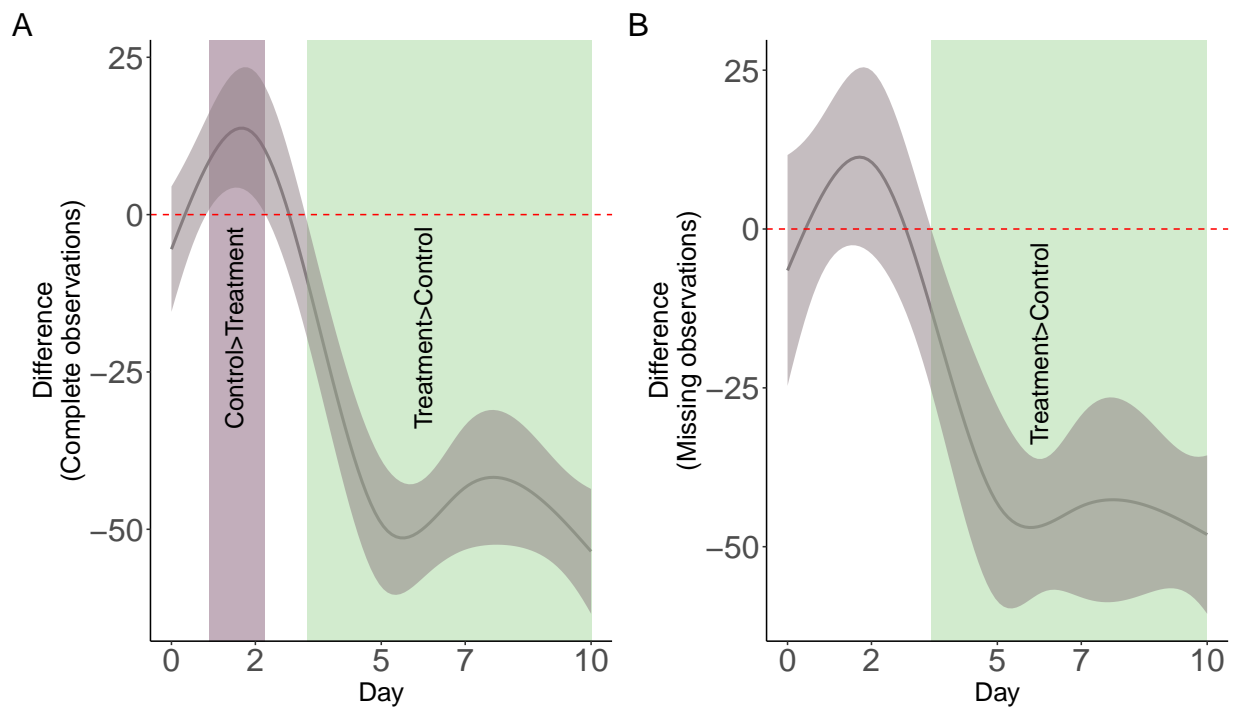


Figure 4: Pairwise comparisons for smooth terms. **A**: Pairwise comparisons for the full dataset. **B**: Pairwise comparisons for the dataset with missing observations. Significant differences exist where the 95% empirical Bayesian credible interval does not cover 0. In both cases the effect of treatment is significant after day 3. In both cases, a simultaneous CI has been computed around the smooth difference.

in one of the groups (assuming the other group is a Control or Reference group).

Consider the calculation of pairwise differences for the smooths in Figure 3B and Figure 3D. Figure 4 shows the comparison between each treatment group for the full and missing datasets with a simultaneous CI computed around the difference. Here, the “Control” group is used as the reference to which “Treatment” group is being compared. Notice that because we have included the means of each group, there is correspondence between the scale of the original data the scale of the pairwise comparisons. This can be seen as in Figure 3B at day 5 there is essentially a 50% difference between StO_2 in both groups, which corresponds to the -50 difference in 4A. However, if there are multiple parametric terms in the model (more factors that need to be specified in the model) such inclusion of the means can become problematic. However, we believe that the model we have presented here suffices in a wide range of situations where adding the group means is a relatively easy implementation that can help better visualize the estimation from the model from a biological perspective.

In Figure 4A the shaded regions over the confidence interval (where it does not cover 0) indicate the time interval where each group has a higher effect than the other. Notice that the shaded region between days 0 and ≈ 2 for the full dataset indicates that through that time, the “Control” group has higher mean StO_2 , but as therapy progresses the effect is reversed and by ≈ 3 day it is the “Treatment” group the one that on average, has greater StO_2 . This would suggest that the effect of chemotherapy in the “Treatment” group becomes significant after day 3 for the given model. Moreover, notice that although there is no actual measurement at day 3, the model is capable of providing an estimate of when the shift in mean StO_2 occurs.

On the data with missing observations (Figure 3D), the smooth pairwise comparison (Figure 4B) shows that because the confidence intervals overlap for the first two days there is essentially no difference between the groups. However, because the model was still able to pick the overall trend in StO_2 , the pairwise comparison is able to estimate the change on day 3 where the Treatment Group becomes significant as in the full dataset smooth pairwise comparison.

In a sense, the pairwise smooth comparison is more informative than a *post-hoc p-value*. For biomedical studies, the smooth comparison is able to provide an estimate of *when* and by *how much* a biological process becomes significant. This is advantageous because it can help researchers gain insight on metabolic changes and other biological processes that can be worth examining, and can help refine the experimental design of future studies in order to obtain measurements at time points where a significant change might be expected.

6 Discussion

Biomedical longitudinal data is particularly challenging to analyze due to the likelihood of missing observations and different correlation structures in the data, which limit the use of rm-ANOVA. Although LMEMs can handle missing observations and different correlation structures, both LMEMs and rm-ANOVA yield biased estimates when they are used to fit data with non-linear trends as we have visually demonstrated in Section 3.5, where it is clear that these models do not capture the non-linear trend of the data, thereby causing a “model misspecification error.”

This “model misspecification” error, also is known as a “Type III” error [17] is particularly important because although the *p-value* is the common measure of statistical significance, the validity of its interpretation is determined by the agreement of the data and the model. Polynomial effects can be used in LMEMs, and it could be argued that in Section 3.5 a LMEM with quadratic effects could have been fitted, but in reality the true function in the data is not known; using polynomial effects presents more questions than answers in this case (i.e., what is the appropriate polynomial effect to use? quadratic?, cubic?, what is the biological interpretation of a polynomial effect?). Moreover, polynomial effects have other associated issues that we have described in Section 4.2.1.

Guidelines for statistical reporting in biomedical journals exist (the SAMPL guidelines) [66] but they have not been widely adopted and in the case of longitudinal data, we consider that researchers would benefit from reporting a visual assessment of the correspondence between the model fit and the data, rather than merely relying on a R^2 value, which is not meaningful in the case of a Type III error.

In this paper we have presented GAMs as a suitable method to analyze longitudinal data with non-linear

trends.

It is interesting to note that although GAMs are a well established method to analyze temporal data in different fields (among which are palaeoecology, geochemistry, and ecology) [33,57] they are not routinely used in biomedical research despite an early publication from Hastie and Tibshirani that demonstrated their use in medical research [67]. This is possibly due to the fact that the theory behind GAMs can seem very different from that of rm-ANOVA and LMEMs, but the purpose of Section 4.2.1 is to demonstrate that at its core the principle is quite simple: Instead of using a linear relationship to model the response (as rm-ANOVA and LMEMs do), GAMs use basis functions to build smooths that are capable of learning non-linear trends in the data, which is a major advantage over models where the user has to know the non-linear relationship *a priori* in order to provide it to the model (i.e., polynomial effects in LMEMs).

However, from a practical standpoint is equally important to demonstrate how GAMs are computationally implemented. We have provided an example on how GAMs can be fitted using simulated data that follows trends reported in biomedical literature [16] using R and the package *mgcv*[37] in Section 5, while a basic workflow for model selection is in the Appendix. One of the features of GAMs is that their Bayesian interpretation allows to indicate differences between groups without the need of a *p-value*, and in turn provide a time-based estimate of shifts in the response that can be directly tied to biological values as the pairwise smooth comparisons in Figure 4 indicate. The model is therefore able to identify changes between the groups at time points where data was not directly measured even with missing data exists (\approx day 3 in Figure 4 A, B), which can be used by researchers as feedback on experiment design and to further evaluate important biological changes in future studies.

We have used R as the software of choice for this paper because not only provides a fully developed environment to fit GAMs, but also eases simulation (which is becoming increasingly used for exploratory statistical analysis and power calculations) and provides powerful and convenient methods of visualization, which are key aspects that biomedical researchers might need to consider to make their work reproducible. In this regard, reproducibility is still an issue in biomedical research [68,69], but it is becoming apparent that what other disciplines have experienced in this aspect is likely to impact sooner rather than later this field. Researchers need to plan on how they will make their data, code, and any other materials open and accessible as more journals and funding agencies recognize the importance and benefits of open science in biomedical research. We have made all the data and code used in this paper accessible, and we hope that this will encourage other researchers to do the same with future projects.

7 Conclusion

We have presented GAMs as a method to analyze longitudinal biomedical data. Future directions of this work will include simulation-based estimations of statistical power using GAMs, as well as demonstrating the prediction capabilities of these models using large datasets. By making the data and code used in this paper accessible, we hope to address the need of creating and sharing reproducible work in biomedical research.

8 Acknowledgements

This work was supported by the National Science Foundation Career Award (CBET 1751554, TJM) and the Arkansas Biosciences Institute.

9 Declaration of Conflicting Interests

The Authors declare that there is no conflict of interest.

Supplementary Materials

An Appendix which contains all the code used to create this manuscript, along with a basic workflow to implement GAMs in R is available as Supplementary Material in PDF. A GitHub repository containing all

the code used for this paper along with detailed instructions for its use is available at <https://github.com/aimundo/GAMs-biomedical-research>.

10 References

- [1] D. Roblyer, S. Ueda, A. Cerussi, W. Tanamai, A. Durkin, R. Mehta, D. Hsiang, J.A. Butler, C. McLaren, W.-P. Chen, B. Tromberg, Optical imaging of breast cancer oxyhemoglobin flare correlates with neoadjuvant chemotherapy response one day after starting treatment, *Proceedings of the National Academy of Sciences*. 108 (2011) 14626–14631. <https://doi.org/10.1073/pnas.1013103108>.
- [2] A. Tank, H.M. Peterson, V. Pera, S. Tabassum, A. Leproux, T. O’Sullivan, E. Jones, H. Cabral, N. Ko, R.S. Mehta, B.J. Tromberg, D. Roblyer, Diffuse optical spectroscopic imaging reveals distinct early breast tumor hemodynamic responses to metronomic and maximum tolerated dose regimens, *Breast Cancer Research*. 22 (2020). <https://doi.org/10.1186/s13058-020-01262-1>.
- [3] M.V. Pavlov, T.I. Kalganova, Y.S. Lyubimtseva, Multimodal approach in assessment of the response of breast cancer to neoadjuvant chemotherapy, *Journal of Biomedical Optics*. 23 (2018) 1. <https://doi.org/10.1117/1.jbo.23.9.091410>.
- [4] V. Demidov, A. Maeda, M. Sugita, V. Madge, S. Sadanand, C. Flueraaru, I.A. Vitkin, Preclinical longitudinal imaging of tumor microvascular radiobiological response with functional optical coherence tomography, *Scientific Reports*. 8 (2018). <https://doi.org/10.1038/s41598-017-18635-w>.
- [5] G. Ritter, L. Cohen, C. Williams, E. Richards, L. Old, S. Welt, Serological analysis of human anti-human antibody responses in colon cancer patients treated with repeated doses of humanized monoclonal antibody A33, *Cancer Research*. 61 (2001) 6851–6859.
- [6] E.M. Roth, A.C. Goldberg, A.L. Catapano, A. Torri, G.D. Yancopoulos, N. Stahl, A. Brunet, G. Lecorps, H.M. Colhoun, Antidrug antibodies in patients treated with alirocumab, *New England Journal of Medicine*. 376 (2017) 1589–1590. <https://doi.org/10.1056/nejmc1616623>.
- [7] J.D. Jones, H.E. Ramser, A.E. Woessner, K.P. Quinn, In vivo multiphoton microscopy detects longitudinal metabolic changes associated with delayed skin wound healing, *Communications Biology*. 1 (2018). <https://doi.org/10.1038/s42003-018-0206-4>.
- [8] M.C. Skala, A. Fontanella, L. Lan, J.A. Izatt, M.W. Dewhirst, Longitudinal optical imaging of tumor metabolism and hemodynamics, *Journal of Biomedical Optics*. 15 (2010) 011112. <https://doi.org/10.1117/1.3285584>.
- [9] G.J. Greening, K.P. Miller, C.R. Spainhour, M.D. Cato, T.J. Muldoon, Effects of isoflurane anesthesia on physiological parameters in murine subcutaneous tumor allografts measured via diffuse reflectance spectroscopy, *Biomedical Optics Express*. 9 (2018) 2871. <https://doi.org/10.1364/boe.9.002871>.
- [10] T.T. Sio, P.J. Atherton, B.J. Birkhead, D.J. Schwartz, J.A. Sloan, D.K. Seisler, J.A. Martenson, C.L. Loprinzi, P.C. Griffin, R.F. Morton, J.C. Anders, T.J. Stoffel, R.E. Haselow, R.B. Mowat, M.A.N. Wittich, J.D. Bearden, R.C. Miller, Repeated measures analyses of dermatitis symptom evolution in breast cancer patients receiving radiotherapy in a phase 3 randomized trial of mometasone furoate vs placebo (N06C4 [alliance]), *Supportive Care in Cancer*. 24 (2016) 3847–3855. <https://doi.org/10.1007/s00520-016-3213-3>.
- [11] J.I. Kamstra, P.U. Dijkstra, M. van Leeuwen, J.L.N. Roodenburg, J.A. Langendijk, Mouth opening in patients irradiated for head and neck cancer: A prospective repeated measures study, *Oral Oncology*. 51 (2015) 548–555. <https://doi.org/10.1016/j.oraloncology.2015.01.016>.
- [12] E.-J. Wagenmakers, M. Lee, T. Lodewyckx, G.J. Iverson, Bayesian versus frequentist inference, in: *Bayesian Evaluation of Informative Hypotheses*, Springer New York, 2008: pp. 181–207. https://doi.org/10.1007/978-0-387-09612-4_9.
- [13] R. Gueorguieva, J.H. Krystal, Move over ANOVA, *Archives of General Psychiatry*. 61 (2004) 310. <https://doi.org/10.1001/archpsyc.61.3.310>.
- [14] P. Schober, T.R. Vetter, Repeated measures designs and analysis of longitudinal data, *Anesthesia & Analgesia*. 127 (2018) 569–575. <https://doi.org/10.1213/ane.0000000000003511>.
- [15] J. Pinheiro, D. Bates, *Mixed-effects models in S and S-PLUS*, Springer Science; Business Media, 2006. <https://doi.org/https://doi.org/10.1007/b98882>.

- [16] K. Vishwanath, H. Yuan, W.T. Barry, M.W. Dewhirst, N. Ramanujam, Using optical spectroscopy to longitudinally monitor physiological changes within solid tumors, *Neoplasia*. 11 (2009) 889–900. <https://doi.org/10.1593/neo.09580>.
- [17] B. Dennis, J.M. Ponciano, M.L. Taper, S.R. Lele, Errors in statistical inference under model misspecification: Evidence, hypothesis testing, and AIC, *Frontiers in Ecology and Evolution*. 7 (2019). <https://doi.org/10.3389/fevo.2019.00372>.
- [18] B. Wang, Z. Zhou, H. Wang, X.M. Tu, C. Feng, The p-value and model specification in statistics, *General Psychiatry*. 32 (2019) e100081. <https://doi.org/10.1136/gpsych-2019-100081>.
- [19] C. Liu, T.P. Cripe, M.-O. Kim, Statistical issues in longitudinal data analysis for treatment efficacy studies in the biomedical sciences, *Molecular Therapy*. 18 (2010) 1724–1730. <https://doi.org/10.1038/mt.2010.127>.
- [20] L.G. Halsey, D. Curran-Everett, S.L. Vowler, G.B. Drummond, The fickle p value generates irreproducible results, *Nature Methods*. 12 (2015) 179–185. <https://doi.org/10.1038/nmeth.3288>.
- [21] H. Abdi, Holm’s sequential Bonferroni procedure, *Encyclopedia of Research Design*. 1 (2010) 1–8. <https://doi.org/10.4135/9781412961288.n178>.
- [22] S. Nakagawa, A farewell to bonferroni: The problems of low statistical power and publication bias, *Behavioral Ecology*. 15 (2004) 1044–1045. <https://doi.org/10.1093/beheco/arh107>.
- [23] A. Gelman, J. Hill, M. Yajima, Why we (usually) don't have to worry about multiple comparisons, *Journal of Research on Educational Effectiveness*. 5 (2012) 189–211. <https://doi.org/10.1080/19345747.2011.618213>.
- [24] C. Albers, The problem with unadjusted multiple and sequential statistical testing, *Nature Communications*. 10 (2019). <https://doi.org/10.1038/s41467-019-09941-0>.
- [25] C. Ugrinowitsch, G.W. Fellingham, M.D. Ricard, Limitations of ordinary least squares models in analyzing repeated measures data, *Medicine & Science in Sports & Exercise*. (2004) 2144–2148. <https://doi.org/10.1249/01.mss.0000147580.40591.75>.
- [26] H. Huynh, L.S. Feldt, Estimation of the box correction for degrees of freedom from sample data in randomized block and split-plot designs, *Journal of Educational Statistics*. 1 (1976) 69–82. <https://doi.org/10.3102/10769986001001069>.
- [27] S.W. Greenhouse, S. Geisser, On methods in the analysis of profile data, *Psychometrika*. 24 (1959) 95–112. <https://doi.org/10.1007/bf02289823>.
- [28] N. Haverkamp, A. Beauducel, Violation of the sphericity assumption and its effect on type-i error rates in repeated measures ANOVA and multi-level linear models (MLM), *Frontiers in Psychology*. 8 (2017). <https://doi.org/10.3389/fpsyg.2017.01841>.
- [29] H.J. Keselman, J. Algina, R.K. Kowalchuk, The analysis of repeated measures designs: A review, *British Journal of Mathematical and Statistical Psychology*. 54 (2001) 1–20. <https://doi.org/10.1348/000711001159357>.
- [30] J. Charan, N. Kantharia, How to calculate sample size in animal studies?, *Journal of Pharmacology and Pharmacotherapeutics*. 4 (2013) 303. <https://doi.org/10.4103/0976-500x.119726>.
- [31] D.J. Barr, R. Levy, C. Scheepers, H.J. Tily, Random effects structure for confirmatory hypothesis testing: Keep it maximal, *Journal of Memory and Language*. 68 (2013) 255–278. <https://doi.org/10.1016/j.jml.2012.11.001>.
- [32] N.L. Rose, H. Yang, S.D. Turner, G.L. Simpson, An assessment of the mechanisms for the transfer of lead and mercury from atmospherically contaminated organic soils to lake sediments with particular reference to scotland, UK, *Geochimica Et Cosmochimica Acta*. 82 (2012) 113–135. <https://doi.org/10.1016/j.gca.2010.12.026>.
- [33] E.J. Pedersen, D.L. Miller, G.L. Simpson, N. Ross, Hierarchical generalized additive models in ecology: An introduction with mgcv, *PeerJ*. 7 (2019) e6876. <https://doi.org/10.7717/peerj.6876>.
- [34] G.L. Simpson, Modelling palaeoecological time series using generalised additive models, *Frontiers in Ecology and Evolution*. 6 (2018). <https://doi.org/10.3389/fevo.2018.00149>.

- [35] L. Yang, G. Qin, N. Zhao, C. Wang, G. Song, Using a generalized additive model with autoregressive terms to study the effects of daily temperature on mortality, *BMC Medical Research Methodology*. 12 (2012). <https://doi.org/10.1186/1471-2288-12-165>.
- [36] N. Beck, S. Jackman, Beyond linearity by default: Generalized additive models, *American Journal of Political Science*. 42 (1998) 596. <https://doi.org/10.2307/2991772>.
- [37] S.N. Wood, *Generalized additive models*, Chapman; Hall/CRC, 2017. <https://doi.org/10.1201/9781315370279>.
- [38] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020. <https://www.R-project.org/>.
- [39] S.N. Wood, N. Pya, B. Säfken, Smoothing parameter and model selection for general smooth models, *Journal of the American Statistical Association*. 111 (2016) 1548–1563. <https://doi.org/10.1080/01621459.2016.1180986>.
- [40] B.T. West, K.B. Welch, A.T. Galecki, *Linear mixed models: A practical guide using statistical software*, second edition, Taylor & Francis, 2014. <https://books.google.com/books?id=hjT6AwAAQBAJ>.
- [41] R.D. Wolfinger, Heterogeneous variance: Covariance structures for repeated measures, *Journal of Agricultural, Biological, and Environmental Statistics*. 1 (1996) 205. <https://doi.org/10.2307/1400366>.
- [42] R.E. Weiss, *Modeling longitudinal data*, Springer New York, 2005. <https://doi.org/10.1007/0-387-28314-5>.
- [43] S. Geisser, S.W. Greenhouse, An extension of box's results on the use of the F distribution in multivariate analysis, *The Annals of Mathematical Statistics*. 29 (1958) 885–891. <https://doi.org/10.1214/aoms/1177706545>.
- [44] S.E. Maxwell, H.D. Delaney, K. Kelley, *Designing experiments and analyzing data*, Routledge, 2017. <https://doi.org/10.4324/9781315642956>.
- [45] G. Molenberghs, Analyzing incomplete longitudinal clinical trial data, *Biostatistics*. 5 (2004) 445–464. <https://doi.org/10.1093/biostatistics/kxh001>.
- [46] Y. Ma, M. Mazumdar, S.G. Memtsoudis, Beyond repeated-measures analysis of variance, *Regional Anesthesia and Pain Medicine*. 37 (2012) 99–105. <https://doi.org/10.1097/aap.0b013e31823ebc74>.
- [47] J. Scheffer, Dealing with missing data, *Research Letters in the Information and Mathematical Sciences*. 3 (2002) 153–160.
- [48] R.F. Potthoff, G.E. Tudor, K.S. Pieper, V. Hasselblad, Can one assess whether missing data are missing at random in medical studies?, *Statistical Methods in Medical Research*. 15 (2006) 213–234. <https://doi.org/10.1191/0962280206sm448oa>.
- [49] G.E.P. Box, *Science and statistics*, 71 (1976) 791–799. <https://doi.org/10.1080/01621459.1976.10480949>.
- [50] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, R Core Team, *nlme: Linear and nonlinear mixed effects models*, 2020. <https://CRAN.R-project.org/package=nlme>.
- [51] R. OHara, J. Kotze, Do not log-transform count data, (2010). <https://doi.org/10.1038/npre.2010.4136.1>.
- [52] J.A. Nelder, R.W.M. Wedderburn, Generalized linear models, *Journal of the Royal Statistical Society. Series A (General)*. 135 (1972) 370. <https://doi.org/10.2307/2344614>.
- [53] C. McCulloch, *Generalized, linear, and mixed models*, John Wiley & Sons, New York, 2001.
- [54] A. Dobson, *An introduction to generalized linear models*, CRC Press, Boca Raton, 2008.
- [55] W. Stroup, *Generalized linear mixed models : Modern concepts, methods and applications*, CRC Press, Taylor & Francis Group, Boca Raton, 2013.
- [56] T. Hastie, R. Tibshirani, Generalized additive models: Some applications, *Journal of the American Statistical Association*. 82 (1987) 371–386. <https://doi.org/10.1080/01621459.1987.10478440>.

- [57] T.J. Hefley, K.M. Broms, B.M. Brost, F.E. Buderman, S.L. Kay, H.R. Scharf, J.R. Tipton, P.J. Williams, M.B. Hooten, The basis function approach for modeling autocorrelation in ecological data, *Ecology*. 98 (2017) 632–646. <https://doi.org/10.1002/ecy.1674>.
- [58] E.J. Wegman, I.W. Wright, Splines in statistics, *Journal of the American Statistical Association*. 78 (1983) 351–365. <https://doi.org/10.1080/01621459.1983.10477977>.
- [59] S.N. Wood, Thin plate regression splines, 65 (2003) 95–114. <https://doi.org/10.1111/1467-9868.00374>.
- [60] R. McElreath, *Statistical rethinking*, Chapman; Hall/CRC, 2018. <https://doi.org/10.1201/9781315372495>.
- [61] D.L. Miller, Bayesian views of generalized additive modelling, *arXiv Preprint arXiv:1902.01330*. (2019).
- [62] D. Ruppert, *Semiparametric regression*, Cambridge University Press, Cambridge New York, 2003.
- [63] G. Marra, S.N. Wood, Coverage properties of confidence intervals for generalized additive model components, *Scandinavian Journal of Statistics*. 39 (2012) 53–74. <https://doi.org/10.1111/j.1467-9469.2011.00760.x>.
- [64] G.L. Simpson, Gratia: Graceful 'ggplot'-based graphics and other functions for GAMs fitted using 'mgcv', 2020. <https://CRAN.R-project.org/package=gratia>.
- [65] J. Harezlak, D. Ruppert, M.P. Wand, *Semiparametric regression with R*, Springer New York, 2018. <https://doi.org/10.1007/978-1-4939-8853-2>.
- [66] T.A. Lang, D.G. Altman, Basic statistical reporting for articles published in biomedical journals: The “statistical analyses and methods in the published literature” or the SAMPL guidelines, *International Journal of Nursing Studies*. 52 (2015) 5–9. <https://doi.org/10.1016/j.ijnurstu.2014.09.006>.
- [67] T. Hastie, R. Tibshirani, Generalized additive models for medical research, *Statistical Methods in Medical Research*. 4 (1995) 187–196. <https://doi.org/10.1177/096228029500400302>.
- [68] C.G. Begley, J.P.A. Ioannidis, Reproducibility in science, *Circulation Research*. 116 (2015) 116–126. <https://doi.org/10.1161/circresaha.114.303819>.
- [69] T.L. Weissgerber, O. Garcia-Valencia, V.D. Garovic, N.M. Milic, S.J. Winham, Why we need to report more than 'data were analyzed by t-tests or ANOVA', *eLife*. 7 (2018). <https://doi.org/10.7554/elife.36163>.

A APPENDIX

This section presents the code used to generate figures, models and simulated data from the main manuscript.

A.1 Compound symmetry and independent errors in linear and quadratic responses

This section simulates linear and quadratic data in the same manner as in Section 3.5 in the main manuscript. The linear simulations using Figure A.1 show in panels A and D the simulated mean responses and individual data points. Panels C and G show a visual interpretation of “correlation” in the responses: In panel C, subjects that have a value of the random error ε either above or below the mean group response are more likely to have other observations that follow the same trajectory, thereby demonstrating correlation in the response. In panel G, because the errors are independent, there is no expectation that responses are likely to follow a similar pattern. Panels D and H show the predictions from the rm-ANOVA model.

The following code chunks produce a more comprehensive exploration of Figure 1 in the main manuscript.

First, a function is created to simulate data across six timepoints using a linear or quadratic mean response, with correlated or uncorrelated errors. Each group has a different slope/concavity. The main function is the same for both groups, but a change in the sign allows to invert the trend.

```
#####Section for calculations#####

## Example with linear response

#This function simulates data using a linear or quadratic mean response
  and each with correlated
#or uncorrelated errors. Each group has a different slope/concavity.
example <- function(n_time = 6, #number of time points
                    fun_type = "linear", #type of response
                    error_type = "correlated") {

  if (!(fun_type %in% c("linear", "quadratic")))
    stop('fun_type must be either "linear", or "quadratic"')
  if (!(error_type %in% c("correlated", "independent")))
    stop('fun_type must be either "correlated", or "independent"')

  x <- seq(1,6, length.out = n_time)

  #Create mean response matrix: linear or quadratic
  mu <- matrix(0, length(x), 2)
  # linear response
  if (fun_type == "linear") {
    mu[, 1] <- - (0.25*x)+2
    mu[, 2] <- 0.25*x+2
  } else {
    # quadratic response (non-linear)

    mu[, 1] <- -(0.25 * x^2) +1.5*x-1.25
    mu[, 2] <- (0.25 * x^2) -1.5*x+1.25
  }

  #create an array where individual observations per each time point for
    each group are to be stored. Currently using 10 observations per
```



```

    timepoint
y <- array(0, dim = c(length(x), 2, 10))

#Create array to store the "errors" for each group at each timepoint.
  The "errors" are the
#between-group variability in the response.
errors <- array(0, dim = c(length(x), 2, 10))
#create an array where 10 observations per each time point for each
  group are to be stored

#The following loops create independent or correlated responses. To each
  value of mu (mean response per group) a randomly generated error (
  correlated or uncorrelated) is added and thus the individual response
  is created.
if (error_type == "independent") {
  ## independent errors
  for (i in 1:2) {
    for (j in 1:10) {
      errors[, i, j] <- rnorm(6, 0, 0.25)
      y[, i, j] <- mu[, i] + errors[, i, j]
    }
  }
} else {
  for (i in 1:2) {      # number of treatments
    for (j in 1:10) {    # number of subjects
      # compound symmetry errors: variance covariance matrix
      errors[, i, j] <- rmvn(1, rep(0, length(x)), 0.1 * diag(6) + 0.25
        * matrix(1, 6, 6))
      y[, i, j] <- mu[, i] + errors[, i, j]
    }
  }
}

## subject random effects

## visualizing the difference between independent errors and compound
  symmetry
## why do we need to account for this -- overly confident inference

#labeling y and errors
dimnames(y) <- list(time = x,
                    treatment = 1:2,
                    subject = 1:10)

dimnames(errors) <- list(time = x,
                        treatment = 1:2,
                        subject = 1:10)

#labeling the mean response
dimnames(mu) <- list(time = x,
                    treatment = 1:2)

#convert y, mu and errors to dataframes with time, treatment and

```

```

      subject columns
dat <- as.data.frame.table(y,
                           responseName = "y")
dat_errors <- as.data.frame.table(errors,
                                   responseName = "errors")
dat_mu <- as.data.frame.table(mu,
                              responseName = "mu")

#join the dataframes to show mean response and errors per subject
dat <- left_join(dat, dat_errors,
                by = c("time", "treatment", "subject"))
dat <- left_join(dat, dat_mu,
                by = c("time", "treatment"))

#add time
dat$time <- as.numeric(as.character(dat$time))
#label subjects per group
dat <- dat %>%
  mutate(subject = factor(paste(subject,
                                treatment,
                                sep = "-")))

## repeated measures ANOVA

fit_anova <- lm(y ~ time + treatment + time * treatment, data = dat)

#LME: time and treatment interaction model, compound symmetry
fit_lme <- lme(y ~ treatment + time + treatment:time,
              data = dat,
              random = ~ 1 | subject,
              correlation = corCompSymm(form = ~ 1 | subject)
            )

#create a prediction frame where the model can be used for plotting
purposes
pred_dat <- expand.grid(
  treatment = factor(1:2),
  time = unique(dat$time)
)

#add model predictions to the dataframe that has the simulated data
dat$pred_anova <- predict(fit_anova)
dat$pred_lme <- predict(fit_lme)

#return everything in a list
return(list(
  dat = dat,
  pred_dat = pred_dat,
  fit_anova=fit_anova,
  fit_lme = fit_lme
))
}

#####Section for plotting#####

```

```

#This function will create the plots for either a "linear" or "quadratic"
  response

plot_example <- function(sim_dat) {
  ## Plot the simulated data (scatterplot)
  txt<-20
  p1 <- sim_dat$dat %>%
    ggplot(aes(x = time,
                y = y,
                group = treatment,
                color = treatment)
            ) +
    geom_point(show.legend=FALSE) +
    labs(y='response')+
    geom_line(aes(x = time,
                  y = mu,
                  color = treatment),
              show.legend=FALSE) +
    theme_classic() +
    theme(plot.title = element_text(size = txt,
                                     face = "bold"),
          text=element_text(size=txt))+
    thm1

#plot the simulated data with trajectories per each subject
p2 <- sim_dat$dat %>%
  ggplot(aes(x = time,
              y = y,
              group = subject,
              color = treatment)
          ) +
  geom_line(aes(size = "Subjects"),
            show.legend = FALSE) +
  # facet_wrap(~ treatment) +
  geom_line(aes(x = time,
                y = mu,
                color = treatment,
                size = "Simulated Truth"),
            lty = 1, show.legend = FALSE) +
  labs(y='response')+
  scale_size_manual(name = "Type", values=c("Subjects" = 0.5, "Simulated
    Truth" = 3)) +
  theme_classic()+
  theme(plot.title = element_text(size = txt,
                                   face = "bold"),
        text=element_text(size=txt))+
  thm1

#plot the errors
p3 <- sim_dat$dat %>%
  ggplot(aes(x = time,
              y = errors,
              group = subject,

```

```

        color = treatment)) +
geom_line(show.legend=FALSE) +
  labs(y='errors')+
  theme_classic()+
  theme(plot.title = element_text(size = txt,
                                   face = "bold"),
        text=element_text(size=txt))+
thm1

#plot the model predictions for rm-ANOVA
p4 <- ggplot(sim_dat$dat,
            aes(x = time,
                y = y,
                color = treatment)) +
  geom_point(show.legend=FALSE)+
  labs(y='response')+
  geom_line(aes(y = predict(sim_dat$fit_anova),
                      group = subject, size = "Subjects"),show.legend = FALSE)
  +
  geom_line(data = sim_dat$pred_dat,
            aes(y = predict(sim_dat$fit_anova,
                            level = 0,
                            newdata = sim_dat$pred_dat),
                size = "Population"),
            show.legend=FALSE) +
  guides(color = guide_legend(override.aes = list(size = 2)))+
  scale_size_manual(name = "Predictions",
                    values=c("Subjects" = 0.5, "Population" = 3)) +
  theme_classic() +
  theme(plot.title = element_text(size = txt,
                                   face = "bold"),
        text=element_text(size=txt))+
thm1

#plot the LMEM predictions
p5 <- ggplot(sim_dat$dat,
            aes(x = time,
                y = y,
                color = treatment)) +
  geom_point()+
  labs(y='response')+
  geom_line(aes(y = predict(sim_dat$fit_lme),
                      group = subject, size = "Subjects")) +
  geom_line(data = sim_dat$pred_dat,
            aes(y = predict(sim_dat$fit_lme,
                            level = 0,
                            newdata = sim_dat$pred_dat),
                size = "Population")) +
  guides(color = guide_legend(override.aes = list(size = 2)))+
  scale_size_manual(name = "Predictions",
                    values=c("Subjects" = 0.5, "Population" = 3)) +
  theme_classic() +

```

```

      theme(plot.title = element_text(size = txt,
                                      face = "bold"),
            text=element_text(size=txt))+
      thm1

    return((p1+p3+p2+p4+p5)+plot_layout(nrow=1)+plot_annotation(tag_levels =
      'A'))

}

#Store each plot in a separate object
A1<-plot_example(example(fun_type = "linear", error_type = "correlated"))

B1<-plot_example(example(fun_type = "linear", error_type = "independent"))

C1<-plot_example(example(fun_type = "quadratic", error_type = "correlated"
  ))

D1<-plot_example(example(fun_type = "quadratic", error_type = "independent"
  ))

```

For the quadratic response case, Figure A.2 shows the simulated responses using compound symmetry and independent errors.

A.2 Basis functions and GAMs

This code produces Figure 2 from the main manuscript. Briefly, a non-linear (quadratic) response is simulated, a gam model is fitted and the basis are extracted in order to explain how the smooth is constructed. The code for data simulation is used again here for the sake of keeping the same structure, although the data can be simulated in a more simple fashion.

```

#generate the response: the same initial procedure from the previous
  section to simulate
#the response
set.seed(1)
n_time = 6
x <- seq(1,6, length.out = n_time)
mu <- matrix(0, length(x), 2)
mu[, 1] <- -(0.25 * x^2) +1.5*x-1.25 #mean response
mu[, 2] <- (0.25 * x^2) -1.5*x+1.25 #mean response
y <- array(0, dim = c(length(x), 2, 10))
errors <- array(0, dim = c(length(x), 2, 10))
for (i in 1:2) { # number of treatments
  for (j in 1:10) { # number of subjects
    # compound symmetry errors
    errors[, i, j] <- rmvn(1, rep(0, length(x)), 0.1 * diag(6) + 0.25
      * matrix(1, 6, 6))
    y[, i, j] <- mu[, i] + errors[, i, j]
  }
}

#label each table

```

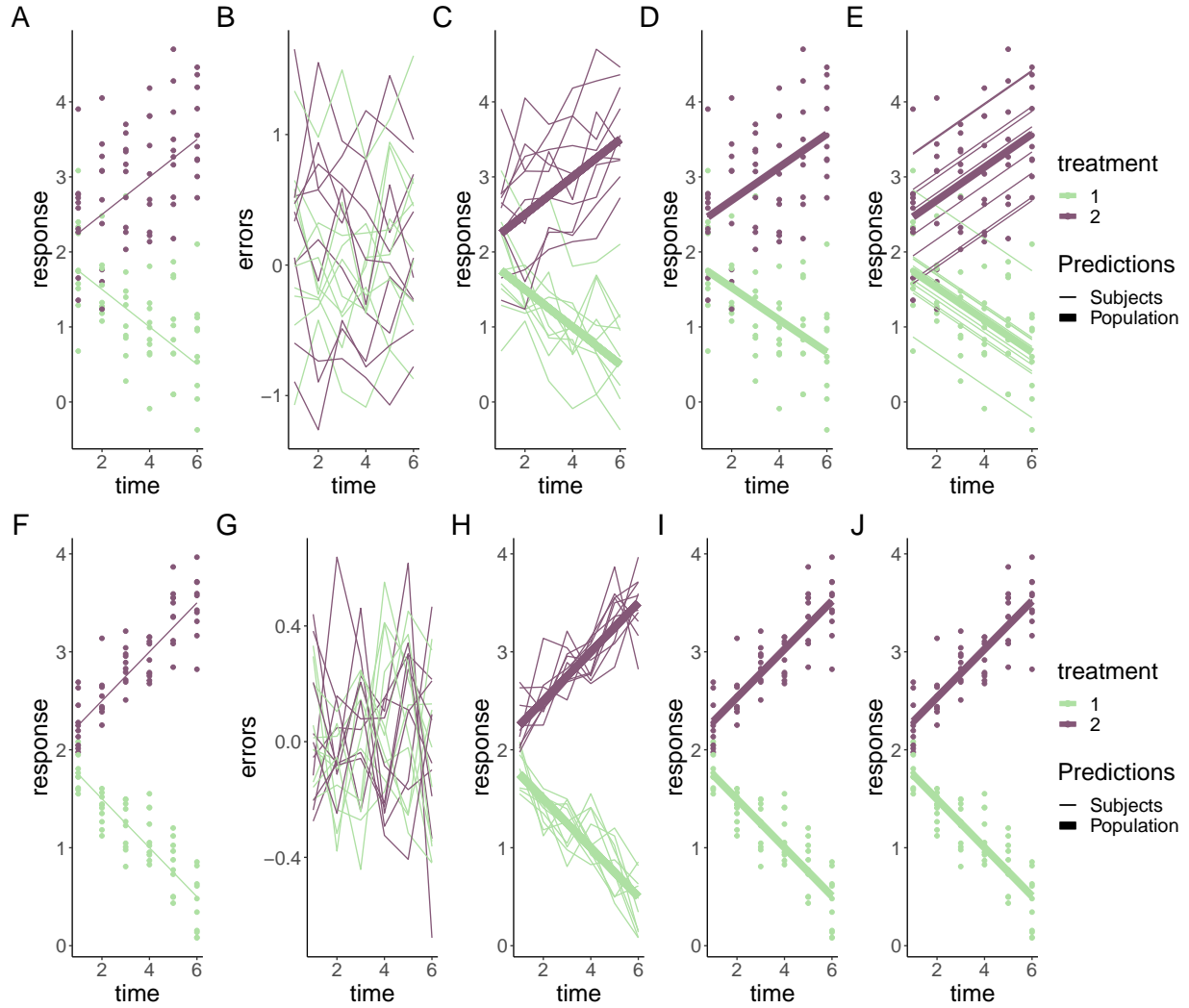


Figure A.1: Simulated linear responses from two groups with correlated (top row) or independent (bottom row) errors using a rm-ANOVA model and a LMEM. A, F: Simulated data with known mean response and individual responses (points) showing the dispersion of the data. B, G: Generated errors showing the difference in the behavior of correlated and independent errors. C, H: Simulated data with thin lines representing individual trajectories. D, I: Estimations from the rm-ANOVA model for the mean group response. E, J: Estimations from the LMEM for the mean group response and individual responses (thin lines). In all panels, thick lines are the predicted mean response per group, thin lines are the random effects for each subject and points represent the original raw data. Both rm-ANOVA and the LMEM are able to capture the trend of the data.

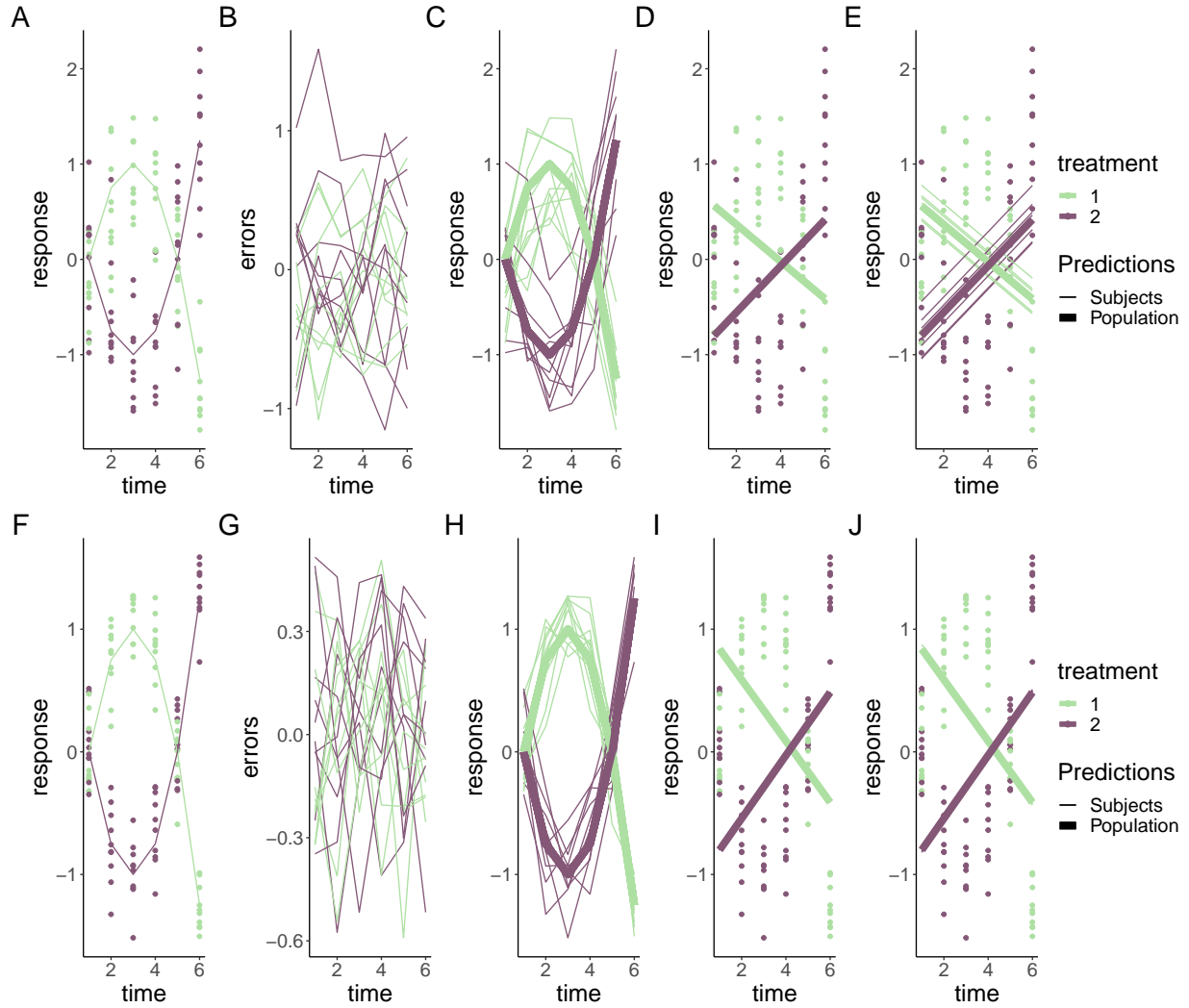


Figure A.2: Simulated quadratic responses from two groups with correlated (top row) or independent (bottom row) errors using a rm-ANOVA model and a LMEM. A, F: Simulated data with known mean response and individual responses (points) showing the dispersion of the data. B, G: Generated errors showing the difference in the behavior of correlated and independent errors. C, H: Simulated data with thin lines representing individual trajectories. D, I: Estimations from the rm-ANOVA model for the mean group response. E, J: Estimations from the LMEM for the mean group response and individual responses (thin lines). In all panels, thick lines are the predicted mean response per group, thin lines are the random effects for each subject and points represent the original raw data. Both rm-ANOVA and the LMEM are not able to capture the changes in each group over time.

```

  dimnames(y) <- list(time = x, treatment = 1:2, subject = 1:10)
  dimnames(errors) <- list(time = x, treatment = 1:2, subject = 1:10)
  dimnames(mu) <- list(time = x, treatment = 1:2)

#Convert to dataframes with subject, time and group columns
dat <- as.data.frame.table(y, responseName = "y")
dat_errors <- as.data.frame.table(errors, responseName = "errors")
dat_mu <- as.data.frame.table(mu, responseName = "mu")
dat <- left_join(dat, dat_errors, by = c("time", "treatment", "subject"))
dat <- left_join(dat, dat_mu, by = c("time", "treatment"))
dat$time <- as.numeric(as.character(dat$time))

#label subject per group
dat <- dat %>%
  mutate(subject = factor(paste(subject, treatment, sep = "-")))

#extract "Group 1" to fit the GAM
dat<-subset(dat,treatment==1)
#keep just the response and timepoint columns
dat<-dat[,c('y','time')]

#GAM model of time, 5 basis
gm<-gam(y~s(time,k=5),data=dat)

#model_matrix (also known as) 'design matrix'
#will contain the smooths used to create model 'gm'
model_matrix<-as.data.frame(predict(gm,type='lpmatrix'))

time<-c(1:6)

basis<-model_matrix[1:6,] #extracting basis (because the values are
  repeated after every 6 rows)
#basis<-model_matrix[1:6,-1] #extracting basis
colnames(basis)[colnames(basis)=="(Intercept)"]<- "s(time).0"
basis<-basis %>% #pivoting to long format
  pivot_longer(
    cols=starts_with("s")
  )%>%
  arrange(name) #ordering

#length of dataframe to be created: number of basis by number of
  timepoints (minus 1 for the intercept that we won't plot)
ln<-6*(length(coef(gm)))

basis_plot<-data.frame(Basis=integer(ln),
  value_orig=double(ln),
  time=integer(ln),
  cof=double(ln)
)

basis_plot$time<-rep(time) #pasting timepoints
basis_plot$Basis<-factor(rep(c(1:5),each=6)) #pasting basis number values
basis_plot$value_orig<-basis$value #pasting basis values

```



```

basis_plot$cof<-rep(coef(gm)[1:5],each=6) #pasting coefficients
basis_plot<-basis_plot%>%
  mutate(mod_val=value_orig*cof) #the create the predicted values the
    bases need to be
#multiplied by the coefficients

#creating labeller to change the labels in the basis plots

basis_names<-c(
  '1'="Intercept",
  '2'="1",
  '3'="2",
  '4'="3",
  '5'="4"
)

#calculating the final smooth by aggregating the basis functions

smooth<-basis_plot%>%
  group_by(time)%>%
  summarize(smooth=sum(mod_val))

#original basis
sz<-1
p11<-ggplot(basis_plot,
  aes(x=time,
      y=value_orig,
      colour=as.factor(Basis)
    )
  )+
  geom_line(size=sz,
    show.legend=FALSE)+
  geom_point(size=sz+1,
    show.legend = FALSE)+
  labs(y='Basis functions')+
  facet_wrap(~Basis,
    labeller = as_labeller(basis_names)
  )+
  theme_classic()+
  thm

#penalized basis
p12<-ggplot(basis_plot,
  aes(x=time,
      y=mod_val,
      colour=as.factor(Basis)
    )
  )+
  geom_line(show.legend = FALSE,
    size=sz)+
  geom_point(show.legend = FALSE,
    size=sz+1)+

```

```

labs(y='Penalized \n basis functions')+
scale_y_continuous(breaks=seq(-1,1,1))+
facet_wrap(~Basis,
            labeller=as_labeller(basis_names)
            )+
theme_classic()+
thm

#heatmap of the coefficients
x_labels<-c("Intercept","1","2","3","4")
p13<-ggplot(basis_plot,
            aes(x=Basis,
                y=Basis))+
geom_tile(aes(fill = cof),
          colour = "black") +
  scale_fill_gradient(low = "white",
                      high = "#B50A2AFF")+ #color picked from KikiMedium

labs(x='Basis',
     y='Basis')+
scale_x_discrete(labels=x_labels)+
geom_text(aes(label=round(cof,2)),
          size=7,
          show.legend = FALSE)+
theme_classic()+
theme(legend.title = element_blank())

#plotting simulated datapoints and smooth term
p14<-ggplot(data=dat,
            aes(x=time,y=y))+
geom_point(size=sz+1)+
labs(y='Simulated \n response')+
geom_line(data=smooth,
          aes(x=time,
              y=smooth),
          color="#6C581DFF",
          size=sz+1)+
theme_classic()

#Combining all
b_plot<-p11+p13+p12+p14+plot_annotation(tag_levels='A')&
  theme(
    text=element_text(size=18)
  )

```

A.3 Longitudinal biomedical data simulation and GAMs

This section describes how to fit GAMs to longitudinal data using simulated data. First, data is simulated according to Section 5, where reported data of oxygen saturation (StO₂) in tumors under either chemotherapy or saline control is used as a starting point to generate individual responses in each group.

```

dat<-tibble(StO2=c(4,27,3,2,0.5,7,4,50,45,56),
            Day=rep(c(0,2,5,7,10),times=2),
            Group=as.factor(rep(c("Control","Treatment"),each=5))

```

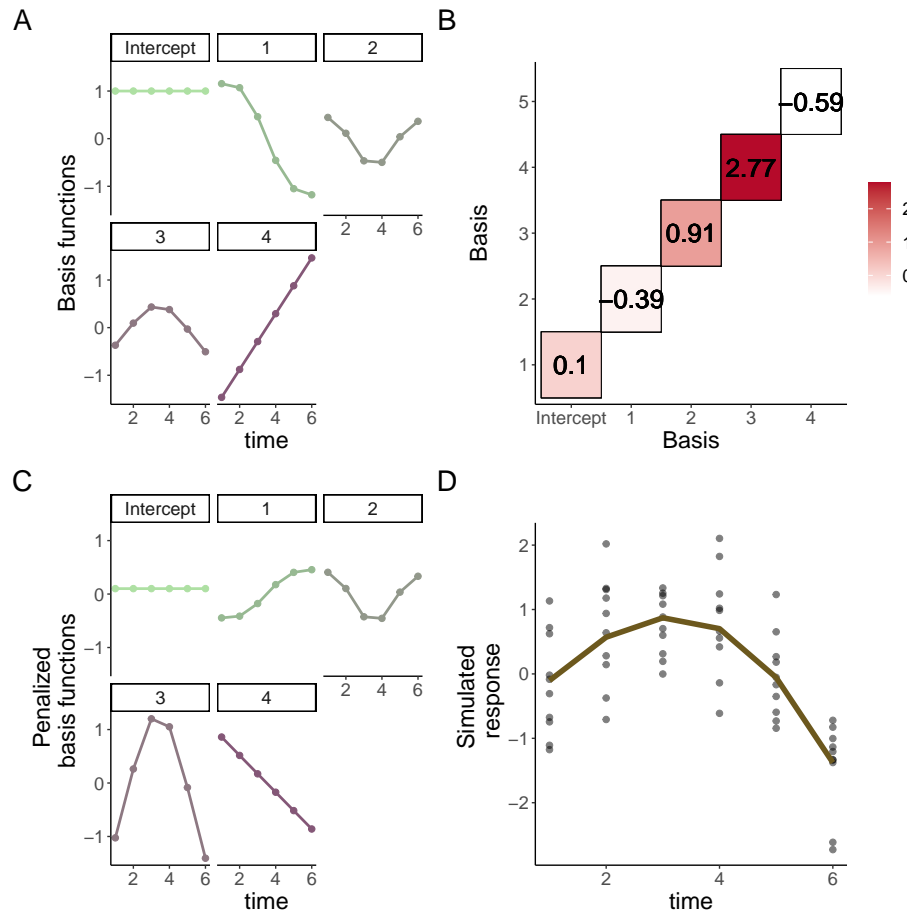


Figure A.3: Basis functions for a single smoother for time. **A:** Basis functions for a single smoother for time for the simulated data of Group 1 from Figure 2. **B:** Matrix of basis function weights. Each basis function is multiplied by a coefficient which can be positive or negative. The coefficient determines the overall effect of each basis in the final smoother. **C:** Weighted basis functions. Each of the four basis functions of panel A has been weighted by the corresponding coefficient shown in Panel B. Note the corresponding increase (or decrease) in magnitude of each weighted basis function. **D:** Smoother for time and original data points. The smoother (line) is the result of the sum of each weighted basis function at each time point, with simulated values for the group shown as points.

```

    )
#alpha for ribbon
al<-0.8

#This function simulates data for the tumor data using default parameters
  of 10 observations per time point, and Standard deviation (sd) of 5%.
#Because physiologically St02 cannot go below 0%, data is generated with
  a cutoff value of 0.0001 (the "St02_sim")

simulate_data <- function(dat, n = 10, sd = 5) {
  dat_sim <- dat %>%
    slice(rep(1:n(), each = n)) %>%
    group_by(Group, Day) %>%
    mutate(
      St02_sim = pmax(rnorm(n, St02, sd), 0.0001),
      subject=rep(1:10),
      subject=factor(paste(subject, Group, sep = "-"))
    ) %>%
    ungroup()

  return(dat_sim)
}

#subject = factor(paste(subject, treatment, sep = "-"))

n <- 10 #number of observations
sd <- 10 #approximate sd from paper
df <- 6
dat_sim <- simulate_data(dat, n, sd)

#plotting simulated data
f2<-ggplot(dat_sim, aes(x = Day,
  y = St02_sim,
  color = Group,
  group=subject)) +
  geom_point(show.legend=FALSE,
    size=1.5,
    alpha=0.6)+
  geom_line(size=0.6, alpha=0.4, show.legend=FALSE)+
  geom_line(aes(x = Day,
    y = St02,
    color=Group),
    size=1.5,
    data=dat,
    inherit.aes=FALSE,
    show.legend = FALSE)+
  labs(y=expression(atop(St0[2], '(simulated)')))+
  theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+
  scale_x_continuous(breaks=c(0,2,5,7,10))+

```

A.4 A basic Workflow for GAMs

This section shows a basic workflow to fit a series of increasingly complex GAMs to the simulated data from the previous section. Graphical and parameter diagnostics for goodness of fit are discussed, as well as model comparison via AIC (Aikake Information Criterion). For simplicity, the confidence intervals (CIs) shown in this section for the models are the across the function CIs created by `mgcv` by default. However, for the pairwise comparisons of the third model we use *simultaneous intervals* as in the main manuscript, and the code for creating simultaneous CIs for the smooths can be found in Section A.5.1.

A.4.1 First model

The first model fitted to the data is one that only accounts for different smooths by day. The model syntax specifies that `gam_00` is the object that will contain all the model information, and that the model attempts to explain changes in `StO2_sim` (simulated StO₂) using a smooth per Day. The model will use 5 basis functions (`k=5`) for the smooth. The smooth is constructed by default using thin plate regression splines. The smoothing parameter estimation method used is the restricted maximum likelihood (REML).

```
gam_00<-gam(StO2_sim ~ s(Day, k = 5),
            method='REML',
            data = dat_sim)
```

To obtain model diagnostics, two methodologies are to be used: 1) graphical diagnostics, and 2) a model check. In the first case, the functions `appraise` and `draw` from the package *gratia* can be used to obtain a single output with all the graphical diagnostics. For model check, the functions `gam.check` and `summary` from *mgcv* provide detailed information about the model fit and its parameters. Keep in mind that `gam.check` is a function that also provides the graphical diagnostics obtained using *gratia*, if such graphical output is not desired the source code can be accessed typing `gam.check` in the Console, and the code without the graphical output can be used in a custom function, which is the approach we follow later).

A.4.1.1 Graphical diagnostics From the output of the function `appraise` in Figure A.4, the major indicators of concern about the model are the QQ plot of residuals and the histogram of residuals. The QQ plot shows that the errors are not reasonably located along the 45° line (which indicates normality), as there are multiple points that deviate from the trend, specially in the tails. The histogram also shows that the variation (residuals) is not following the assumption of a normal distribution.

The `draw` function permits to plot the smooths as `ggplot2` objects, which eases subsequent manipulation, if desired. Because model `gam_00` specifies only one smooth for the time covariate (Day), the plot only contains only one smooth. Note that the smooth shows an almost linear profile.

A.4.1.2 Model check Special attention must be paid to the parameter ‘k-index’ from `gam.check` (which calls `k.check` to perform the calculation). This parameter indicates if the basis dimension of the smooth is adequate, i.e., it checks that the basis used to create the smooth are adequate to capture the trends in the data. If the model is not adequately capturing the trends in the data, this is indicated by a low k-index value (<1). Because we plot the model diagnostics using `appraise` later, the graphical output from `gam.check` will be suppressed by creating a custom function to obtain just the model estimates, thus avoiding repetition of the diagnostic plots. This will be achieved by calling the source code of `gam.check` and using the appropriate code in a new function that will be called `gam.diagnostics`. We are not including in the Appendix the code for the function `gam.diagnostics` as it is rather long, but if desired it can be accessed by going to the `Appendix.Rmd` file in the GitHub repository and scrolling to this exact place (the code is not included in the final output, but is evaluated to create the function).

We now call `gam.diagnostics` to provide the desired diagnostic output, as well as a summary of the fitted model, which is obtained by calling `summary`.

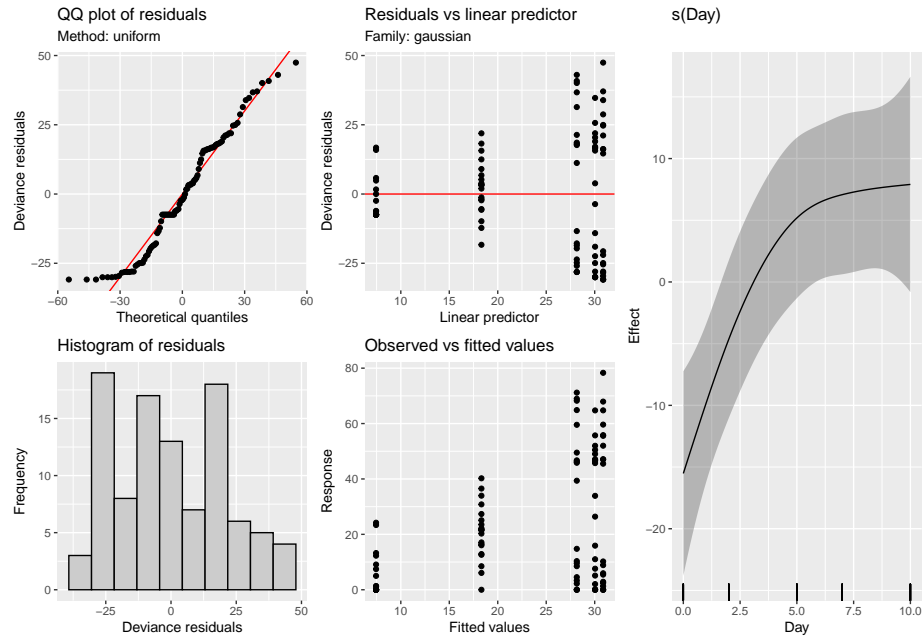


Figure A.4: Graphical diagnostics for the first GAM model. Left: Graphical diagnostics provided by the function `appraise` from the package *gratia*. Right: Fitted smooth for the model, provided by the function `draw`.

```
gam.diagnostics(gam_00)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-0.0003727881,-6.621452e-07]
## (score 444.0118 & scale 450.6638).
## Hessian positive definite, eigenvalue range [0.3881695,49.00676].
## Model rank =  5 / 5
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(Day) 4.00 2.11    0.36  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the output, it can be seen that the *k-index* is 0.36, which indicates that the model is not capturing the variability in the data. The *edf* (effective degrees of freedom) is an indicator of the complexity of the smooth. Here the complexity of the smooth is comparable to that of a 4th degree polynomial.

```
summary(gam_00)
```

```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## St02_sim ~ s(Day, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.967      2.123   10.82  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F  p-value
## s(Day)  2.114   2.565  7.633 0.000517 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.153   Deviance explained = 17.2%
## -REML = 444.01   Scale est. = 450.66      n = 100
```

From the `summary` function, information about the assumed distribution of the errors (Gaussian in this case) and the link function can be obtained. The link function is ‘identity’ as the model does not make any transformation on the predictors. The ‘significance of smooth terms’ *p-value* indicates if each smooth is adding significance to the model. Here, the *p-value* is low but we have seen that there are issues with the model from the previous outputs. Finally, the ‘deviance explained’ indicates how much of the data the model is able to capture, which in this case corresponds to $\approx 17\%$.

A.4.2 Second model

The major flaw of `gam_00` is that this model is not taking into account the fact that the data is nested in groups. The next iteration is a model where a different smooth of time (Day) is assigned for each group using `by = Group` in the model syntax.

```
gam_01<-gam(St02_sim ~ s(Day, by=Group,k = 5),
            method = 'REML',
            data = dat_sim)

gam.diagnostics(gam_01)

##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-5.51754e-05,2.671715e-06]
## (score 423.3916 & scale 280.8777).
## Hessian positive definite, eigenvalue range [0.3162258,48.5557].
## Model rank =  9 / 9
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##             k'   edf k-index p-value
## s(Day):GroupControl  4.00 3.39    0.43  <2e-16 ***
## s(Day):GroupTreatment 4.00 3.23    0.43  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Diagnostics for this model indicate that the k-index is still below 1 (0.43 from `gam.check`), and that the residuals are still not following a normal distribution (Figure A.5). Moreover, the smooths (plotted via the `draw()` function) appear with a fairly linear profile, which indicates they are still not capturing the trends observed in the data.

```
summary(gam_01)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## St02_sim ~ s(Day, by = Group, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   22.967      1.676    13.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(Day):GroupControl  3.392  3.794  3.817  0.0304 *
## s(Day):GroupTreatment 3.229  3.682 21.174 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.472   Deviance explained = 50.8%
## -REML = 423.39   Scale est. = 280.88    n = 100
```

From `summary()`, the deviance explained by the model is $\approx 51\%$.

A.4.3 Third model

Model `gam_00` was built for didactic purposes to cover the simplest case, but it does not account for the nesting of the data by `Group`, which is apparent from the type of smooth fitted (a single smooth), the model diagnostics, and, the low variance explained by the model. On the other hand, `gam_01` takes into account the nesting within each group and provides better variance explanation, but as indicated in Section 5, in order to differentiate between each group a parametric term needs to be added to the model for the interaction of *Day* and *Group*.

This is because in `gam_01` separate smooths were fitted per group and those smooths also tried to account for the different means of the response in the two groups. Adding a parametric term for `Group` enables the smooths to capture the time course-differences of each group. The resulting model is `gam_02`, which is the model fitted in the main manuscript.

```
#GAM for St02

gam_02 <- gam(St02_sim ~ Group+s(Day, by = Group, k = 5),
              method='REML',
              data = dat_sim)

gam.diagnostics(gam_02)
```

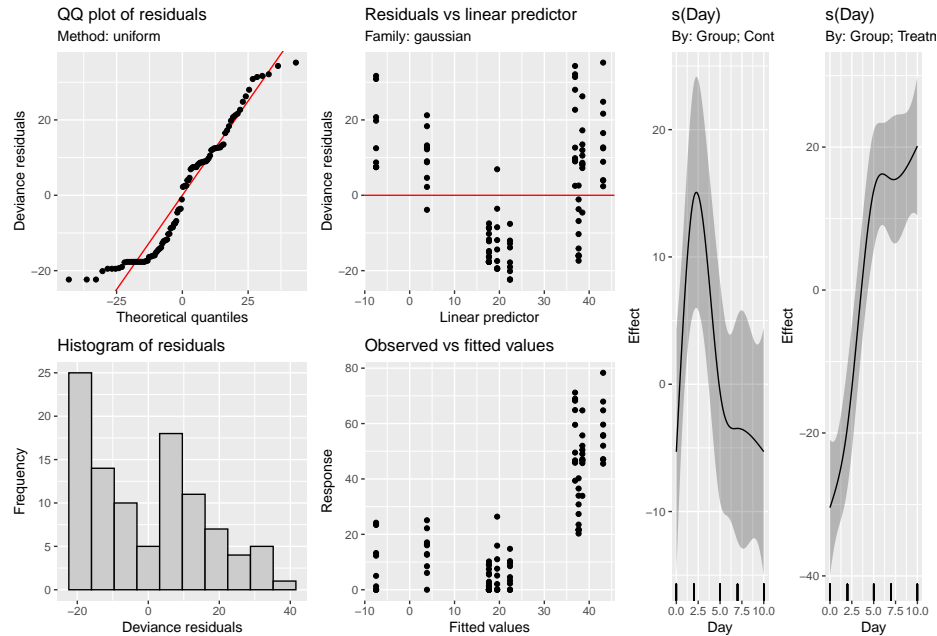



Figure A.5: Graphical diagnostics for the second GAM model. Left: Graphical diagnostics provided by the function `appraise` from the package *gratia*. Right: Fitted smooth for the model, provided by the function `draw`.

```
##
## Method: REML    Optimizer: outer newton
## full convergence after 10 iterations.
## Gradient range [-8.164307e-08,1.500338e-08]
## (score 355.8554 & scale 64.53344).
## Hessian positive definite, eigenvalue range [1.174841,48.08834].
## Model rank = 10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##               k'   edf k-index p-value
## s(Day):GroupControl  4.00 3.87   1.02   0.55
## s(Day):GroupTreatment 4.00 3.88   1.02   0.58
```

By using `appraise()` and `draw` on this model (Figure A.6) we see that the trend on the QQ plot has improved, the histogram of the residuals appears to be reasonably distributed, and the smooths are capturing the trend of the data within each group. From `gam.check`, the k-index is now at an acceptable value (≈ 1.02).

```
summary(gam_02)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## St02_sim ~ Group + s(Day, by = Group, k = 5)
##
```

```
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      9.084      1.136   7.996 4.09e-12 ***
## GroupTreatment    27.766      1.607  17.282 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Day):GroupControl  3.873  3.990 17.57 <2e-16 ***
## s(Day):GroupTreatment 3.879  3.991 89.33 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.879   Deviance explained = 88.9%
## -REML = 355.86   Scale est. = 64.533      n = 100
```

From `summary`, the model is able to capture 89% of the variance in the data, which is a substantial improvement over the variance explained by `gam_00` and `gam_01`.

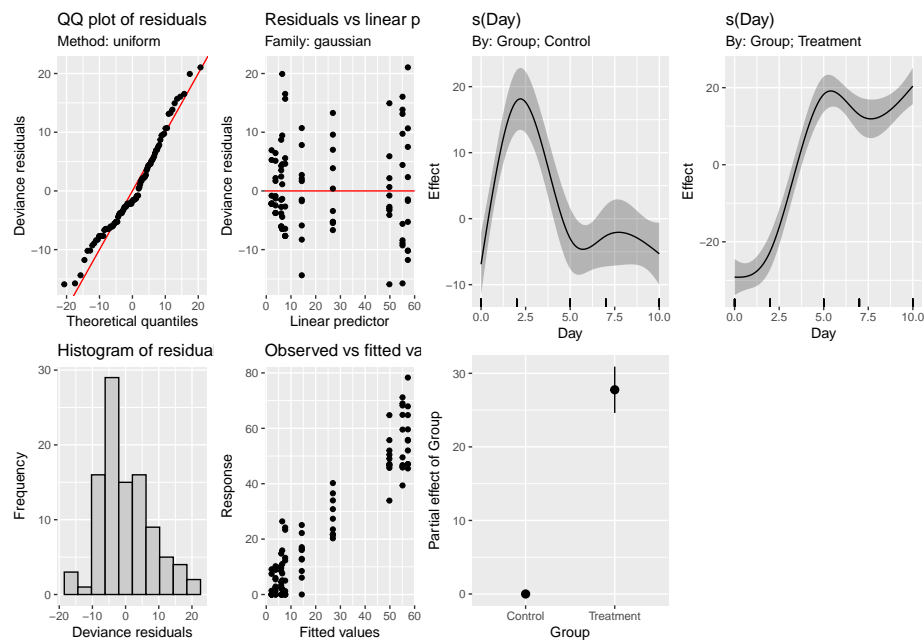


Figure A.6: Graphical diagnostics for the final GAM model. Left: Graphical diagnostics provided by the function `appraise` from the package *gratia*. Right: Fitted smooths for the model, provided by the function `draw`.

A.4.4 Comparing models via AIC

One final comparison that can be made for model selection involves the use of the Aikake Information Criterion (AIC). This metric is used to estimate information loss, which we want to minimize with an appropriate model. Therefore, when 2 or more models are compared, the model with lower AIC is preferred. In R, the comparison is done using the `AIC` function.

```
AIC(gam_00, gam_01, gam_02)
```

##		df	AIC
##	gam_00	4.564893	900.8257
##	gam_01	9.476137	858.6051
##	gam_02	10.980983	712.2067

The output in this case is expected: model `gam_02` has a lower AIC (712.46) whereas the initial two models have higher AICs (900 and 858). The AIC should not be considered as the only estimator of model quality, instead to be used as complimentary information to the graphical diagnostics and model checks described above.

A.4.4.1 Pairwise comparisons of smooth confidence intervals The estimation of significant differences between each treatment group can be achieved via pairwise comparisons of the smooth confidence intervals as described in section 5.3.

In this case, the “design matrix” is used to estimate the pairwise comparisons (see main manuscript for details and associated references). Briefly, the “design matrix” (also known as the “Xp matrix”) from the selected model (`gam_02`) is used to calculate a 95% confidence interval of the difference between the smooth terms for each group. This approach allows to estimate the time intervals where a significant difference exists between the groups (confidence interval above or below 0).

We want to emphasize that for the model used here we have included the group means in order to keep the pairwise comparisons on the scale of the response (a change from the code which appears in the chunk with a comment that says `###IMPORTANT`). This approach works well for models like `gam_02` but implementing this on models with more parametric terms can be challenging. However, we do believe that the model presented in the paper covers a wide range of situations and our approach here for the pairwise comparisons will be useful for most biomedical researchers.

```
##Pairwise comparisons
difference_pointwise <- function(f1, f2, smooth, by_var, smooth_var, data,
  Xp, V, coefs, nrep = 1000) {
  ## make sure f1 and f2 are characters
  f1 <- as.character(f1)
  f2 <- as.character(f2)
  cnames <- colnames(Xp)
  ## columns of Xp associated with pair of smooths
  c1 <- grepl(gratia::mgcv_by_smooth_labels(smooth, by_var, f1), cnames,
    fixed = TRUE)
  c2 <- grepl(gratia::mgcv_by_smooth_labels(smooth, by_var, f2), cnames,
    fixed = TRUE)
  ## rows of Xp associated with pair of smooths
  r1 <- data[[by_var]] == f1
  r2 <- data[[by_var]] == f2

  ## difference rows of Xp for pair of smooths
  X <- Xp[r1, ] - Xp[r2, ]

  ### IMPORTANT: Keep group means###
  ## zero the cols related to other splines
  # X[, !(c1 | c2)] <- 0

  ## zero out the parametric cols
  #X[, !grepl('^s\\(', cnames)] <- 0

  ## compute difference
```

```

sm_diff <- drop(X %*% coefs)
se <- sqrt(rowSums((X %*% V) * X))
nr <- NROW(X)

## Calculate posterior simulation for smooths
coefs_sim <- t(rmvn(nrep, rep(0, nrow(V)), V))
rownames(coefs_sim) <- rownames(V)
simDev <- X %*% coefs_sim
absDev <- abs(sweep(simDev, 1, se, FUN = "/"))
masd <- apply(absDev, 2, max)
crit_s <- quantile(masd, prob = 0.95, type = 8)

out <- list(smooth = rep(smooth, nr), by = rep(by_var, nr),
            level_1 = rep(f1, nr),
            level_2 = rep(f2, nr),
            diff = sm_diff, se = se,
            lower_s = sm_diff - crit_s * se,
            upper_s = sm_diff + crit_s*se)

out <- new_tibble(out, nrow = NROW(X), class = "difference_smooth")
## Only need rows associated with one of the levels
out <- bind_cols(out, data[r1, smooth_var])

out
}

#does both ci and si
difference_smooths <- function(model,
                                smooth,
                                n = 100,
                                ci_level = 0.95,
                                newdata = NULL,
                                partial_match = TRUE,
                                unconditional = FALSE,
                                frequentist = FALSE,
                                nrep = 10000,
                                include_means = TRUE,
                                ...) {
  if (missing(smooth)) {
    stop("Must specify a smooth to difference via 'smooth'.")
  }

  # smooths in model
  S <- gratia::smooths(model) # vector of smooth labels - "s(x)"
  # select smooths
  select <-
    gratia:::check_user_select_smooths(smooths = S, select = smooth,
                                         partial_match = partial_match)#,
  # model_name = expr_label(substitute(object)))
  sm_ids <- which(select)
  smooths <- gratia::get_smooths_by_id(model, sm_ids)
  sm_data <- map(sm_ids, gratia:::smooth_data,
                 model = model, n = n, include_all = TRUE)

```

```

sm_data <- bind_rows(sm_data)
by_var <- by_variable(smooths[[1L]])
smooth_var <- gratia::smooth_variable(smooths[[1L]])
pairs <- as_tibble(as.data.frame(t(combn(levels(sm_data[[by_var]]), 2)),
                                stringsAsFactor = FALSE))

names(pairs) <- paste0("f", 1:2)

Xp <- predict(model, newdata = sm_data, type = "lpmatrix")
V <- gratia::get_vcov(model, unconditional = unconditional,
                     frequentist = frequentist)
coefs <- coef(model)

out <- pmap(pairs, difference_pointwise, smooth = smooth, by_var = by_
  var,
           smooth_var = smooth_var, data = sm_data, Xp = Xp, V = V,
           coefs = coefs, nrep = nrep)
out <- bind_rows(out)
crit <- qnorm((1 - ci_level) / 2, lower.tail = FALSE)

out <- add_column(out,
                 lower = out$diff - (crit * out$sse),
                 upper = out$diff + (crit * out$sse),
                 .after = 6L)

  out
}

#compute difference between smooths and calculate confidence interval:
  complete data
diff_complete <- difference_smooths(gam_02, smooth = "s(Day)", newdata =
  newdat,
                                unconditional = TRUE, frequentist = FALSE,
                                n=100, partial_match = TRUE, nrep=10000)

#function to obtain values for the shading regions
pairwise_limits<-function(dataframe){
  #extract values where the lower limit of the ribbon is greater than
    zero
  #this is the region where the control group effect is greater

  v1<-dataframe%>%
    filter(lower_s>0)%>%
    select(Day)
  #get day initial value
  init1=v1$Day[[1]]
  #get day final value
  final1=v1$Day[[nrow(v1)]]
  #extract values where the value of the upper limit of the ribbon is
    lower than zero
  #this corresponds to the region where the treatment group effect is
    greater
  v2<-dataframe%>%
    filter(upper_s<0)%>%
    select(Day)

```

```

    init2=v2$Day[[1]]
    final2=v2$Day[[nrow(v2)]]
    #store values
    my_list<-list(init1=init1,
                  final1=final1,
                  init2=init2,
                  final2=final2)

    return(my_list)

}

my_list<-pairwise_limits(diff_complete)
rib_col<-'#8D7D82' #color for ribbon for confidence interval
control_rib <- '#875F79' #color for ribbon for control region
treat_rib <- '#A7D89E' #color for ribbon treatment region

c1 <- ggplot() +
  geom_line(data = diff_complete, aes(x = Day, y = diff),size=1, alpha
    =0.5) +
  annotate("rect",
          xmin =my_list$init1, xmax =my_list$final1,ymin=-Inf,ymax=
            Inf,
          fill=control_rib,
          alpha = 0.5,
          )+
  annotate("text",
          x=1.5,
          y=-18,
          label="Control>Treatment",
          size=6,
          angle=90
          )+
  annotate("rect",
          xmin =my_list$init2, xmax =my_list$final2,ymin=-Inf,ymax=Inf,
          fill=treat_rib,
          alpha = 0.5
          ) +
  annotate("text",
          x=6,
          y=-18,
          label="Treatment>Control",
          size=6,
          angle=90
          )+
  geom_ribbon(data = diff_complete, aes(x = Day, ymin = lower_s, ymax =
    upper_s),
            alpha = 0.5, fill = rib_col, inherit.aes = FALSE) +
  geom_hline(yintercept = 0, lty = 2, color = "red")+
  scale_x_continuous(breaks=c(0,2,5,7,10))+
  labs(y="Difference\n(Complete observations)")+
  theme_classic()+

```

```
theme(
  axis.text=element_text(size=22))
```

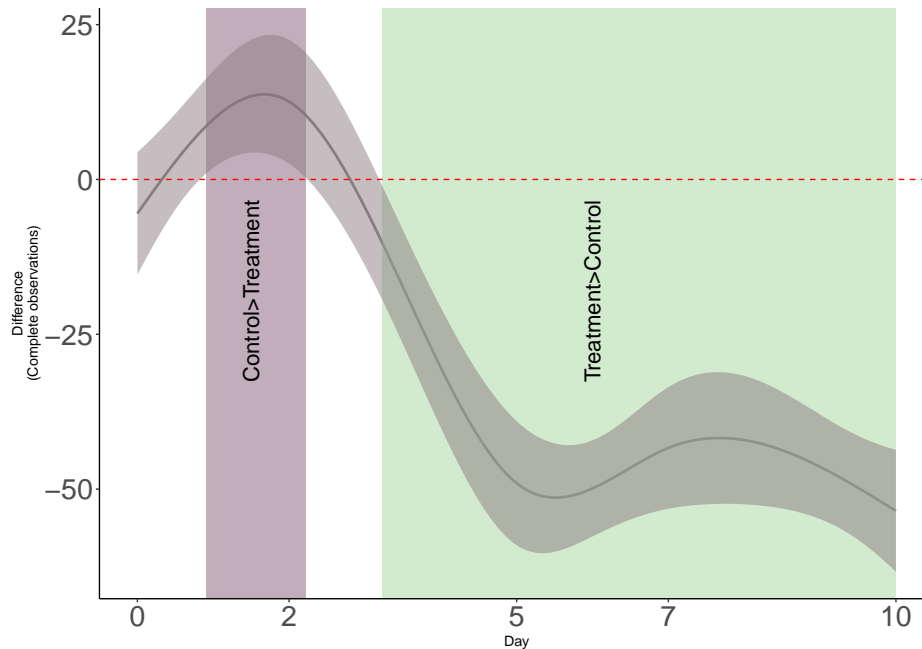


Figure A.7: Smooth pairwise comparisons for model `gam_02` using a 95% simultaneous CI for the difference between smooths. The comparison includes the group means and therefore can be directly correlated with the magnitude of the response. Shaded regions indicate time intervals where each treatment group has a non-zero effect.

Of notice, the package `gratia` contains a function that allows to do pairwise comparisons. In this package, `difference_smooths` is a function that makes the comparisons and produces Figure A.7 when is used on a fitted model. The function syntax and an example can be found at:

<https://cran.r-project.org/web/packages/gratia/gratia.pdf>

Keep in mind that this function removes the group means for the pairwise comparison so the result will not be directly interpretable with the scale of the measured response.

A.5 GAM and Linear model plots and Missing data

This section covers the code used to generate Figure 3, where the simulated data, fit of the “final” GAM (`gam_02`), linear model and GAM on data with missing observations are presented. Note that panel A in Figure 3 and the inset are generated in the code chunk where the data is simulated in Section A.3, and are called later to build the figure.

A.5.1 GAM and Linear model plots

This code chunk creates panels B and C in Figure 3. Note that this code uses the final GAM from the previous section (`gam_02`), so the simulated data and the model should be generated before running this section.

```
#linear model
lm1<-lm(St02_sim ~ Day + Group + Day * Group, data = dat_sim)

## point-wise interval
```

```

ci <- confint(gam_02, parm = "s(Day)", partial_match = TRUE, type = "
  confidence")
## simultaneous interval
si <- confint(gam_02, parm = "s(Day)", type = "simultaneous", partial_
  match = TRUE)

# mean shift for Treatment group
const <- coef(gam_02)[2]

#pointwise confidence interval
ci <- ci %>%
mutate(est = case_when(Group == "Treatment" ~ est + const,
  TRUE ~ est),
lower = case_when(Group == "Treatment" ~ lower + const,
  TRUE ~ lower),
upper = case_when(Group == "Treatment" ~ upper + const,
  TRUE ~ upper))

#simultaneous interval
si <- si %>%
mutate(est = case_when(Group == "Treatment" ~ est + const,
  TRUE ~ est),
lower = case_when(Group == "Treatment" ~ lower + const,
  TRUE ~ lower),
upper = case_when(Group == "Treatment" ~ upper + const,
  TRUE ~ upper))

#creates a dataframe using the length of the covariates for rm-ANOVA
lm_predict<-expand_grid(Group = factor(c("Control", "Treatment")),
  Day = c(0:10),
  subject=factor(rep(1:10)),
  )
lm_predict$subject<-factor(paste(lm_predict$subject, lm_predict$Group, sep
  = "-"))

#using lm
lm_predict<-lm_predict%>%
  mutate(fit = predict(lm1,lm_predict,se.fit = TRUE,type='response')$fit
    ,
    se.fit = predict(lm1, lm_predict,se.fit = TRUE,type='response')
    $se.fit)

#plot smooths and pointwise and simulatenous confidence intervals for GAM
f3<-ggplot(ci, aes(x = Day, y = est, group = smooth)) +
  geom_line(lwd = 1) +
  geom_ribbon(data = ci, mapping = aes(ymin = lower, ymax = upper, x = Day,
    group = smooth,fill = Group),
    inherit.aes = FALSE, alpha = 0.7,
    show.legend=FALSE) +
  geom_ribbon(data = si,
    mapping = aes(ymin = lower, ymax = upper, x = Day, group = smooth,fill =
    Group),

```



```

inherit.aes = FALSE, alpha = 0.4,
show.legend=FALSE)+
  geom_point(data=dat_sim, aes(x = Day,
                                y = St02_sim,
                                color = Group),
              size=1.5,
              alpha=0.6,
              inherit.aes = FALSE,
              show.legend = FALSE)+
  labs(y=expression(atop(St0[2], '(complete observations)')))+
  scale_x_continuous(breaks=c(0,2,5,7,10))+
  theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+thm1

#plot linear fit for rm-ANOVA
f4<-ggplot(data=dat_sim, aes(x=Day, y=St02_sim, group=Group)) +
  geom_point(aes(color=Group),size=1.5,alpha=0.5,show.legend = FALSE)+
  geom_ribbon(aes( x=Day,ymin=(fit - 2*se.fit),
                  ymax=(fit + 2*se.fit),fill=Group),
             alpha=al,
             data=lm_predict,
             show.legend = FALSE,
             inherit.aes=FALSE) +
  geom_line(aes(y=fit,
                color=Group),
            size=1,data=lm_predict,
            show.legend = FALSE)+
  #facet_wrap(~Group)+
  labs(y=expression(paste('St0'[2], ' (simulated)')))+
  scale_x_continuous(breaks=c(0,2,5,7,10))+
  theme_classic()+
  theme(
    axis.text=element_text(size=22)
  )+thm1

```

A.6 Working with Missing data in GAMs

This code chunk first randomly deletes 40% of the total observations in the original simulated data, and then an interaction GAM is fitted to the remaining data. Model diagnostics are presented, and an object that stores the fitted smooths is saved to be called in the final code chunk to build the figure.

```

#missing data
#create a sequence of 40 random numbers between 1 and 100, these numbers
  will
#correspond to the row numbers to be randomly erased from the original
  dataset
missing <- sample(1:100, 40)

#create a new dataframe from the simulated data with 40 rows randomly
  removed, keep the missing values as NA

```

```

ind <- which(dat_sim$StO2_sim %in% sample(dat_sim$StO2_sim, 40))

#create a new dataframe, remove the StO2 column
dat_missing <- dat_sim[,-1]

#add NAs at the ind positions
dat_missing$StO2_sim[ind]<-NA

#Count the number of remaining observations per day (original dataset had
  10 per group per day)
dat_missing %>%
  group_by(Day,Group) %>%
  filter(!is.na(StO2_sim))%>%
  count(Day)

#the same model used for the full dataset
mod_m1 <- gam(StO2_sim ~ Group+s(Day,by=Group,k=5), data = dat_missing,
  family=scat)
#appraise the model
appraise(mod_m1)

ci <- confint(mod_m1, parm = "s(Day)", partial_match = TRUE, type = "
  confidence")
## simultaneous interval
si <- confint(mod_m1, parm = "s(Day)", type = "simultaneous", partial_
  match = TRUE)

# mean shift for group 2
const <- coef(mod_m1)[2]

ci <- ci %>%
mutate(est = case_when(Group == "Treatment" ~ est + const,
  TRUE ~ est),
  lower = case_when(Group == "Treatment" ~ lower + const,
  TRUE ~ lower),
  upper = case_when(Group == "Treatment" ~ upper + const,
  TRUE ~ upper))

si <- si %>%
mutate(est = case_when(Group == "Treatment" ~ est + const,
  TRUE ~ est),
  lower = case_when(Group == "Treatment" ~ lower + const,
  TRUE ~ lower),
  upper = case_when(Group == "Treatment" ~ upper + const,
  TRUE ~ upper))

#plot model and original data
f6<-ggplot(ci, aes(x = Day, y = est, group = smooth)) +
  geom_line(lwd = 1) +
  geom_ribbon(data = ci, mapping = aes(ymin = lower, ymax = upper, x = Day,
    group = smooth,fill = Group),

```

```

inherit.aes = FALSE, alpha = 0.7,
show.legend=FALSE) +
geom_ribbon(data = si,
mapping = aes(ymin = lower, ymax = upper, x = Day, group = smooth, fill =
  Group),
inherit.aes = FALSE, alpha = 0.4,
show.legend=TRUE)+
  geom_point(data=dat_missing, aes(x = Day,
    y = StO2_sim,
    color = Group),
    size=1.5,
    alpha=0.6,
    inherit.aes = FALSE)+
  labs(y=expression(atop(StO[2], '(missing observations)')))+
  scale_x_continuous(breaks=c(0,2,5,7,10))+
  theme_classic()+
  theme(
    axis.text=element_text(size=22))+
    thm1

```

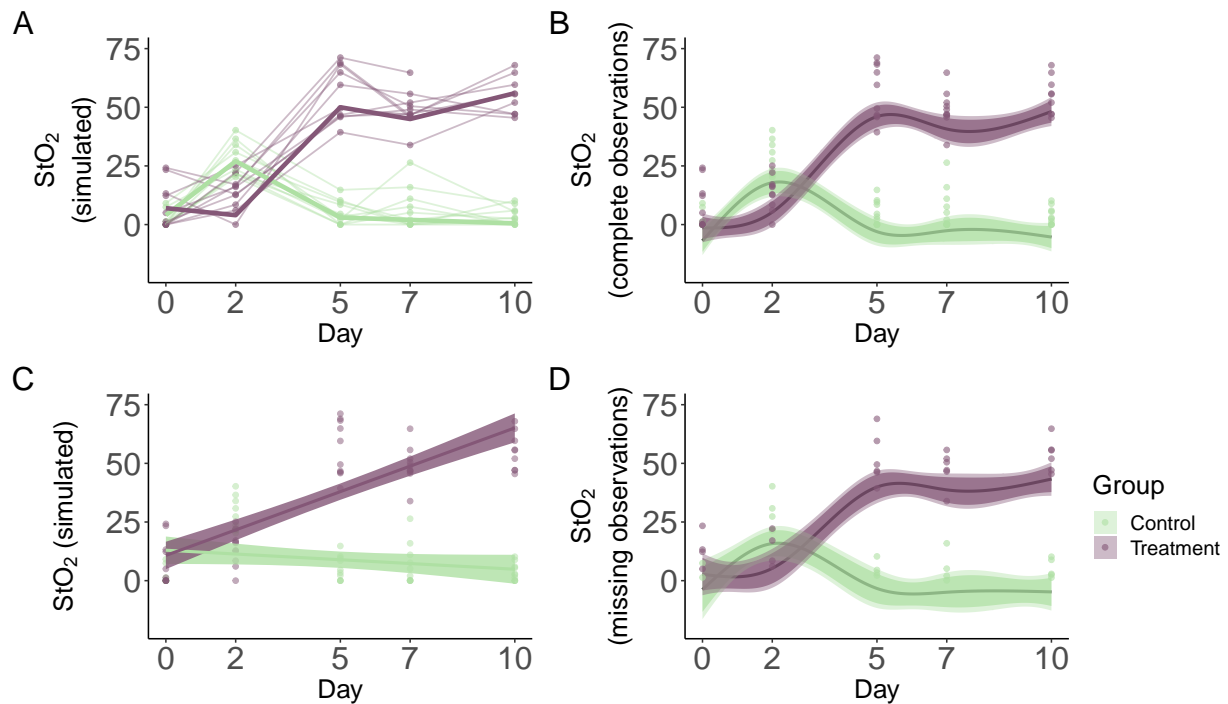


Figure A.8: Simulated data and smooths for oxygen saturation in tumors. **A:** Simulated data (thin lines) that follows previously reported trends (thick lines) in tumors under chemotherapy (Treatment) or saline (Control) treatment. Simulated data is from a normal distribution with standard deviation of 10% with 10 observations per time point. **B:** Smooths from the GAM model for the full simulated data with interaction of Group and Treatment. Lines represent trends for each group, shaded regions are 95% across the function (narrow region) and simultaneous (wide region) confidence intervals. **C:** The rm-ANOVA model for the simulated data, which does not capture the changes in each group over time. **D:** Smooths for the GAM model for the simulated data with missing observations (40%). Lines represent trends for each group, shaded regions are 95% across the function (narrow region) and simultaneous (wide region) confidence intervals.

A.7 Pairwise comparisons in GAMs: full and missing data cases

The next code chunk reproduces Figure 4. Here pairwise comparisons are made for the full and missing datasets by computing the difference between the smooths and a 95% simultaneous CI.

```
##Pairwise comparisons

difference_pointwise <- function(f1, f2, smooth, by_var, smooth_var, data,
  Xp, V, coefs, nrep = 1000) {
  ## make sure f1 and f2 are characters
  f1 <- as.character(f1)
  f2 <- as.character(f2)
  cnames <- colnames(Xp)
  ## columns of Xp associated with pair of smooths
  c1 <- grepl(gratia::mgcv_by_smooth_labels(smooth, by_var, f1), cnames,
    fixed = TRUE)
  c2 <- grepl(gratia::mgcv_by_smooth_labels(smooth, by_var, f2), cnames,
    fixed = TRUE)
  ## rows of Xp associated with pair of smooths
  r1 <- data[[by_var]] == f1
  r2 <- data[[by_var]] == f2

  ## difference rows of Xp for pair of smooths
  X <- Xp[r1, ] - Xp[r2, ]

  ## zero the cols related to other splines
  # X[, ! (c1 | c2)] <- 0

  ## zero out the parametric cols
  #X[, !grepl('^s\\(', cnames)] <- 0

  ## compute difference
  sm_diff <- drop(X %>% coefs)
  se <- sqrt(rowSums((X %>% V) * X))
  nr <- NROW(X)

  ## Calculate posterior simulation for smooths
  coefs_sim <- t(rmvn(nrep, rep(0, nrow(V)), V))
  rownames(coefs_sim) <- rownames(V)
  simDev <- X %>% coefs_sim
  absDev <- abs(sweep(simDev, 1, se, FUN = "/"))
  masd <- apply(absDev, 2, max)
  crit_s <- quantile(masd, prob = 0.95, type = 8)

  out <- list(smooth = rep(smooth, nr), by = rep(by_var, nr),
    level_1 = rep(f1, nr),
    level_2 = rep(f2, nr),
    diff = sm_diff, se = se,
    lower_s = sm_diff - crit_s * se,
    upper_s = sm_diff + crit_s*se)

  out <- new_tibble(out, nrow = NROW(X), class = "difference_smooth")
  ## Only need rows associated with one of the levels
```

```

    out <- bind_cols(out, data[r1, smooth_var])

    out
  }

#does both ci and si
difference_smooths <- function(model,
                                smooth,
                                n = 100,
                                ci_level = 0.95,
                                newdata = NULL,
                                partial_match = TRUE,
                                unconditional = FALSE,
                                frequentist = FALSE,
                                nrep = 10000,
                                include_means = TRUE,
                                ...) {

  if (missing(smooth)) {
    stop("Must specify a smooth to difference via 'smooth'.")
  }

  # smooths in model
  S <- gratia::smooths(model) # vector of smooth labels - "s(x)"
  # select smooths
  select <-
    gratia:::check_user_select_smooths(smooths = S, select = smooth,
                                         partial_match = partial_match)#,
  # model_name = expr_label(substitute(object))
  sm_ids <- which(select)
  smooths <- gratia::get_smooths_by_id(model, sm_ids)
  sm_data <- map(sm_ids, gratia:::smooth_data,
                 model = model, n = n, include_all = TRUE)
  sm_data <- bind_rows(sm_data)
  by_var <- by_variable(smooths[[1L]])
  smooth_var <- gratia:::smooth_variable(smooths[[1L]])
  pairs <- as_tibble(as.data.frame(t(combn(levels(sm_data[[by_var]]), 2)),
                                     stringsAsFactor = FALSE))
  names(pairs) <- paste0("f", 1:2)

  Xp <- predict(model, newdata = sm_data, type = "lpmatrix")
  V <- gratia:::get_vcov(model, unconditional = unconditional,
                        frequentist = frequentist)
  coefs <- coef(model)

  out <- pmap(pairs, difference_pointwise, smooth = smooth, by_var = by_
    var,
              smooth_var = smooth_var, data = sm_data, Xp = Xp, V = V,
              coefs = coefs, nrep = nrep)
  out <- bind_rows(out)
  crit <- qnorm((1 - ci_level) / 2, lower.tail = FALSE)

  out <- add_column(out,
                    lower = out$diff - (crit * out$se),
                    upper = out$diff + (crit * out$se),

```

```

      .after = 6L)

  out
}

#compute difference between smooths and calculate confidence interval:
  complete data
diff_complete <- difference_smooths(gam_02, smooth = "s(Day)", newdata =
  newdat,
                                unconditional = TRUE, frequentist = FALSE,
                                n=100, partial_match = TRUE, nrep=10000)

#function to obtain values for the shading regions
pairwise_limits<-function(dataframe){
  #extract values where the lower limit of the ribbon is greater than
    zero
  #this is the region where the control group effect is greater

  v1<-dataframe%>%
    filter(lower_s>0)%>%
    select(Day)
  #get day initial value
  init1=v1$Day[[1]]
  #get day final value
  final1=v1$Day[[nrow(v1)]]
  #extract values where the value of the upper limit of the ribbon is
    lower than zero
  #this corresponds to the region where the treatment group effect is
    greater
  v2<-dataframe%>%
    filter(upper_s<0)%>%
    select(Day)

  init2=v2$Day[[1]]
  final2=v2$Day[[nrow(v2)]]
  #store values
  my_list<-list(init1=init1,
                final1=final1,
                init2=init2,
                final2=final2)

  return(my_list)

}

my_list<-pairwise_limits(diff_complete)
rib_col<-'#8D7D82' #color for ribbon for confidence interval
control_rib <- '#875F79' #color for ribbon for control region
treat_rib <- '#A7D89E' #color for ribbon treatment region

c1 <- ggplot() +
  geom_line(data = diff_complete, aes(x = Day, y = diff),size=1, alpha
    =0.5) +

```

```

    annotate("rect",
             xmin =my_list$init1, xmax =my_list$final1,ymin=-Inf,ymax=
             Inf,
             fill=control_rib,
             alpha = 0.5,
             )+
    annotate("text",
             x=1.5,
             y=-18,
             label="Control>Treatment",
             size=6,
             angle=90
             )+
    annotate("rect",
             xmin =my_list$init2, xmax =my_list$final2,ymin=-Inf,ymax=Inf,
             fill=treat_rib,
             alpha = 0.5
             ) +
    annotate("text",
             x=6,
             y=-18,
             label="Treatment>Control",
             size=6,
             angle=90
             )+
    geom_ribbon(data = diff_complete, aes(x = Day, ymin = lower_s, ymax =
    upper_s),
               alpha = 0.5, fill = rib_col, inherit.aes = FALSE) +
    geom_hline(yintercept = 0, lty = 2, color = "red")+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
    labs(y="Difference\n(Complete observations)")+
    theme_classic()+
    theme(
    axis.text=element_text(size=22))

###compute difference in smooths for missing data
diff_missing <- difference_smooths(mod_m1, smooth = "s(Day)", newdata =
    newdat,
                                   unconditional = TRUE, frequentist = FALSE,
                                   n=100, partial_match = TRUE, nrep=10000)

v2<- diff_missing %>%
    filter(upper_s<0)%>%
    select(Day)
init2=v2$Day[[1]]
final2=v2$Day[[nrow(v2)]]

c2 <- ggplot() +
    geom_line(data = diff_missing, aes(x = Day, y = diff),size=1, alpha=0.5)
    +
    annotate("rect",
             xmin =init2, xmax = final2,ymin=-Inf,ymax=Inf,

```

```

        fill=treat_rib,
        alpha = 0.5,
    ) +
    annotate("text",
            x=6,
            y=-18,
            label="Treatment>Control",
            size=6,
            angle=90
    )+
    geom_ribbon(data = diff_missing, aes(x = Day, ymin = lower_s, ymax =
        upper_s),
              alpha = 0.5, fill = rib_col, inherit.aes = FALSE) +
    geom_hline(yintercept = 0, lty = 2, color = "red")+
    scale_x_continuous(breaks=c(0,2,5,7,10))+
    labs(y="Difference\n(Missing observations)")+
    theme_classic()+
    theme(
        axis.text=element_text(size=22))

pair_comp<-c1+c2

```

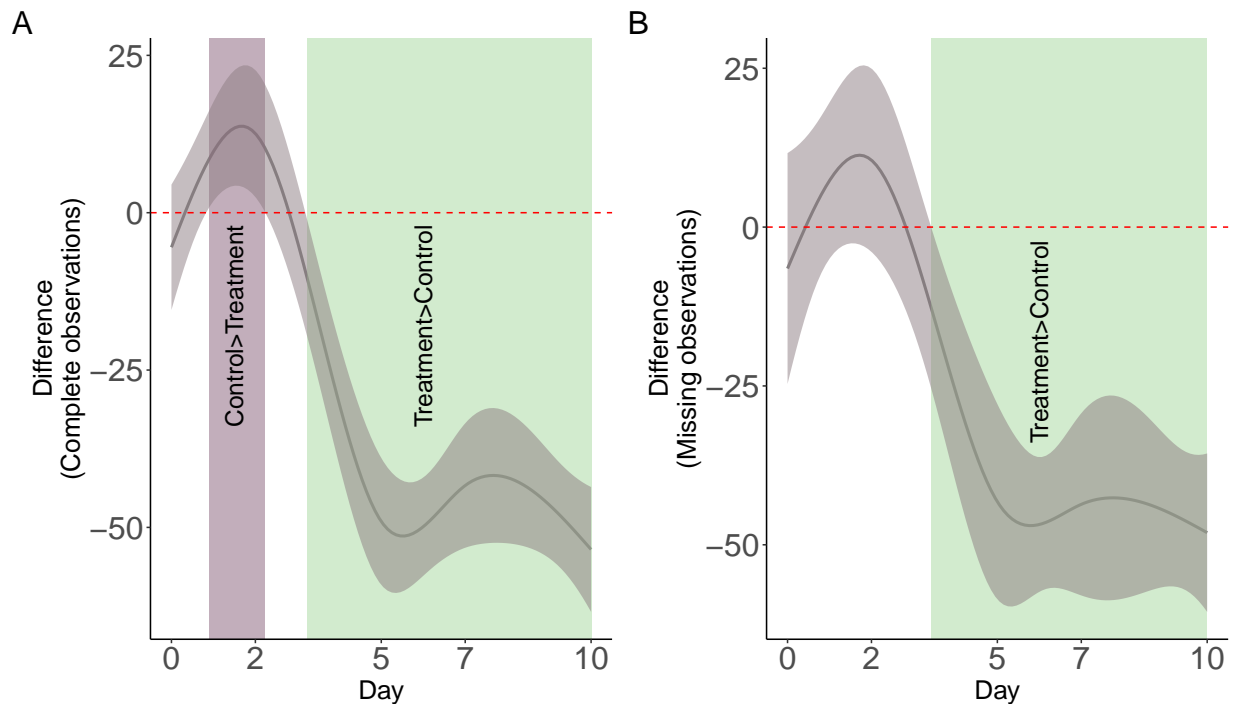


Figure A.9: Pairwise comparisons for smooth terms. **A:** Pairwise comparisons for the full dataset. **B:** Pairwise comparisons for the dataset with missing observations. Significant differences exist where the 95% empirical Bayesian credible interval does not cover 0. In both cases the effect of treatment is significant after day 3. In both cases, a simultaneous CI has been computed around the smooth difference.